UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Aplicación móvil con arquitectura api-rest para mypes

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero de Sistemas

AUTOR:

Torres Núñez, Wilder (ORCID: 0000-0002-0504-4569)

ASESOR:

Dr. Hilario Falcón, Francisco Manuel (ORCID: 0000-0003-3153-9343)

LÍNEA DE INVESTIGACIÓN:

Sistema de Información y Comunicaciones

LIMA – PERÚ

2021

Dedicatoria

En la presente tesis quiero dedicárselo a mis familiares, quienes de manera directa o indirectamente me han ayudado incondicionalmente.

Agradecimiento

Quiero empezar agradeciendo a nuestro Señor por gozar de buena salud y trabajo, a mis padres por enseñarme los valores necesarios para ser una persona correcta e integra, a cada uno de mis hermanos por ayudarme en cada etapa de mi vida, a mi esposa por ayudarme con este informe y a mi asesor por el conocimiento brindado.

Índice de contenidos

I. INTRODUCCIÓN 1
II. MARCO TEÓRICO 7
III. METODOLOGÍA 16
3.1 Tipo y diseño de investigación 17
3.2 Variables y operacionalización 18
3.3 Población, muestra y muestreo 20
3.4 Técnicas e instrumentos de recolección de datos 22
3.5 Procedimientos 23
3.6 Método de análisis de datos 23
3.7 Aspectos éticos 24
IV. RESULTADOS 25
V. DISCUSIÓN 34
VI. CONCLUSIONES 36
VII. RECOMENDACIONES 38
REFERENCIAS 40

Índice de tablas

Tabla 1 Indicador estadístico sobre la reducción del tiempo..... 26
Tabla 2 Prueba de normalidad sobre la reducción del tiempo..... 27
Tabla 3 Prueba de rangos con signos de Wilcoxon sobre el indicador de reducción del tiempo..... 28
Tabla 4 Estadístico de prueba Z sobre el indicador reducción del tiempo..... 29
Tabla 5 Indicadores estadísticos sobre el incremento de la satisfacción..... 29
Tabla 6 Prueba de normalidad sobre el incremento del nivel de satisfacción..... 31
Tabla 7 Prueba de rangos con signos de Wilcoxon sobre el incremento de satisfacción. 32
Tabla 8 Estadístico de prueba Z sobre el incremento de satisfacción. 32
Tabla 9 Resumen de los resultados de aceptación o rechazo de la hipótesis..... 33
Tabla 10: Matriz de consistencia. 49
Tabla 11: Matriz de operacionalización de variables..... 50

Índice de figuras

Figura 1: Ciclo de vida de compra, Laza (2016)..... 2
Figura 2: Framework conceptual Challiol et al.(2017) 4
Figura 3: Esquema de usuario en la base de datos. 61
Figura 4: Store Procedure de validación de usuario. 62
Figura 5: Código fuente del api de validación de usuario. 63
Figura 6: Login de acceso del aplicativo móvil. 64
Figura 7: Código fuente modelo usuario del aplicativo móvil. 67
Figura 8: Código fuente configuración base de datos. 69
Figura 9: Código fuente de configuración inicial de base de datos. 70
Figura 10: Código fuente de la clase login. 73
Figura 11: Código fuente del procedimiento almacenado para la búsqueda de productos. 75
Figura 12: función que retorno el stock general. 76
Figura 13: Código fuente de la función que devuelve el stock según producto y marca. 77
Figura 14: Código fuente de la función que devuelve stock según el productos, marca y color. 78
Figura 15: Código fuente de la función que devuelve el stock según producto, marca, color y talle. 79
Figura 16: Código fuente que controla las peticiones de las imágenes. 80
Figura 17: Código fuente con la función de listar las imágenes. 80
Figura 18: Código fuente del api de gestiona las peticiones para los productos. 82
Figura 19: Código fuente del api que contiene las funciones para los productos. 88
Figura 20: Menú listar productos (Buscar). 89
Figura 21: Pantalla principal de búsqueda de productos. 89
Figura 22: Pantalla emergente que muestra las imágenes de los productos. 90
Figura 23: Pantalla emergente detallando la información del producto. 90
Figura 24: Código fuente de la ventana principal de búsqueda de productos. 93
Figura 25: Código fuente de la pantalla principal que devuelve los productos. 94
Figura 26: Código fuente de la pantalla emergente de muestra las imágenes de los productos. 95
Figura 27: Esquema de los productos nuevos y sus imágenes. 96
Figura 28: Código fuente de verificación del token en el api de productos nuevos. 97
Figura 29: Código fuente del api que recibe peticiones de producto nuevo. 98
Figura 30: Código fuente, extraído del api procesos de productos. 99
Figura 31: Código fuente del receptor de peticiones de subir imágenes. 100
Figura 32: Código fuente de las funciones en general del api de subir imágenes. 102
Figura 33: Código fuente del api peticiones de Productos nuevos. 102
Figura 34: Código fuente api actualizar productos nuevos. 104
Figura 35: Código fuente del api eliminar producto nuevo. 105
Figura 36: Pantalla que lista los productos nuevos ingresados 105
Figura 37: Código fuente de listar productos nuevos. 108
Figura 38: Código fuente del modelo de producto nuevo. 109
Figura 39: Pantalla de producto nuevo. 110
Figura 40: Código fuente de ingresar producto nuevo. 114
Figura 41: Pantalla de modificar o eliminar producto nuevo. 115
Figura 42: Código fuente de la pantalla de modificación o eliminar producto nuevo. . 118

Figura 43: Código fuente del proceso api que permite pasar los productos nuevos... 119
Figura 44: Código fuente de procedimiento almacenado de aprobar nuevos producto. 121
Figura 45: Esquema de orden de compra. 121
Figura 46: Código fuente del api de órdenes de compra, extracción de la función de validación. 122
Figura 47: Código fuente del api que recibe peticiones para las órdenes de compra. 123
Figura 48: Código fuente del api con sus procesos para el listado de órdenes de compra. 124
Figura 49: Código fuente del api de los procesos de órdenes de compra. 125
Figura 50: Código fuente del api con sus procesos de anular para las órdenes de compra. 127
Figura 51: Código fuente de confirmación de órdenes de compra. 128
Figura 52: Código fuente del modelo de orden de compra. 130
Figura 53: Código fuente del modelo de producto para las órdenes de compra. 131
Figura 54: Pantalla del listado de órdenes de compra. 132
Figura 55:Código fuente del listado de órdenes de compra en el aplicativo móvil. 134
Figura 56: Pantalla del listado de órdenes de compra. 135
Figura 57: Código fuente del confirmar, anular y modificar de órdenes de compra. ... 138
Figura 58: Código fuente del trigger para aprobar la orden de compra y se ingresa al stock. 139

Índice de anexos

Anexo 1: Declaratoria de autenticidad del autor 47
Anexo 2: Declaratoria de autenticidad del asesor 48
Anexo 3: Matriz de consistencia 49
Anexo 4: Matriz de operacionalización de variables 50
Anexo 5: Instrumento de recolección de datos 51
Anexo 6: Desarrollo de la metodología 54

Índice de abreviaturas

Frond-end: Se le denomina a la interacción de los usuarios y el back-end mediante un software.

Back-end: Se le denomina a las consultas o peticiones que envía el usuario y procesa la base de datos.

API: Interfaz de programación de aplicaciones.

APK: Paquete de aplicaciones del S.O. Android.

SO: Sistema operativo.

HTTP: Protocolo de transferencia de texto enriquecido.

JSON: Notación de objeto de javascrip.

OC: Orden de Compra

Resumen

La investigación tuvo como problemática la creciente necesidad de pequeños comercios en mejorar sus procesos de compra, que mediante sus sistemas actuales generan demoras en el registro e ingreso de sus productos. El objetivo de la investigación fue reducir el tiempo que generan los procesos de registro de compras, implementando así una aplicación móvil.

Para el desarrollo se pensó en las metodologías ágiles y más concreto en SCRUM donde Deemer et al. (2009) manifestó que es un marco de trabajo que tiene iteraciones y a la vez incremental permitiendo el desarrollo de aplicaciones. Y que existen 3 roles principales; el dueño del producto (PO, por sus siglas en inglés Product Owner), el equipo y el Scrum Master(SM). Para el desarrollo del back-end se prefirió api-rest como una de las tecnologías a usar porque Salas (2021) manifestó que “Con la utilización de las API las compañías tienen la oportunidad de poder abstraer parte de su infraestructura en donde podrán publicar un conjunto de procedimientos y funciones para sus clientes y/o usuarios finales” y para el front-end se prefirió los dispositivos móviles por su disponibilidad y practicidad. Las herramientas a utilizar son PHP para el api-rest y Android Studio para los dispositivos móviles.

Los resultados obtenidos por la implementación de esta aplicación móvil arrojaron que los encuestados redujeron su tiempo en el proceso de orden de compra, logrando reducir el tiempo en 23.60% con respecto a la implementación del mismo.

Donde se puede concluir sobre el uso de la aplicación móvil que tuvo un gran impacto sobre los pequeños comercios en relación a la reducción de tiempos para su proceso de compras. Se recomendó la ampliación del uso de Api-rest para otros procesos de su sistema.

Palabras clave: Aplicación móvil, rest, servicio api-rest, metodologías ágiles, api-rest para un proceso de compra.

Abstract

The research had as a problem the growing need of small businesses to improve their purchasing processes, which through their current systems generate delays in the registration and entry of their products. The objective of the research was to reduce the time generated by the purchase registration processes, thus implementing a mobile application.

For the development we thought of agile methodologies and more specifically SCRUM where Deemer et al. (2009) stated that it is a framework that has iterations and at the same time incremental allowing the development of applications. And that there are 3 main roles; the Product Owner (PO), the team and the Scrum Master (SM). For the back-end development, api-rest was preferred as one of the technologies to be used because Salas (2021) stated that "With the use of APIs, companies have the opportunity to abstract part of their infrastructure where they can publish a set of procedures and functions for their customers and/or end users" and for the front-end, mobile devices were preferred for their availability and practicality. The tools to be used are PHP for the api-rest and Android Studio for the mobile devices.

The results obtained by the implementation of this mobile application showed that the respondents reduced their time in the purchase order management process, being in time a reduction of 23.60% with respect to the implementation of the same.

We can conclude that the use of the mobile application had a great impact on small businesses in terms of time reduction in the purchasing process. It was recommended to expand the use of Api-rest for other processes of their system.

Keywords: Mobile application, rest, api-rest service, agile methodologies, api-rest for a purchasing process.

I. INTRODUCCIÓN

En esta investigación se planteó el impacto que tendría una aplicación móvil en el contexto del registro e ingreso de productos en pequeños comercios, debido a la existencia de herramientas informáticas que, al estar basados en empresas más complejas, dividen claramente los procesos de gestión de compras, generando que las realidades de estos comercios no se adapten de manera natural a estos sistemas, dando así la necesidad de crear una aplicación que permita la separación del comerciante con su sistema con el objetivo de brindarle accesibilidad y movilidad al usuario permitiendo reducir tiempo en dicho proceso.

Una parte importante de la investigación es definir el proceso de compra, donde Laza (2016) define que el proceso de gestión de compra sin lugar a duda requiere de un análisis exhaustivo y de estudio de ofertas y estableció un ciclo de compra:

Primero el análisis de la solicitud de compra, segundo la selección de proveedores, tercero la negociación con los proveedores, cuarto el acompañamiento de la OC, quinto el control de la recepción del material comprado y por último el pago esto lo detallamos en la siguiente imagen:

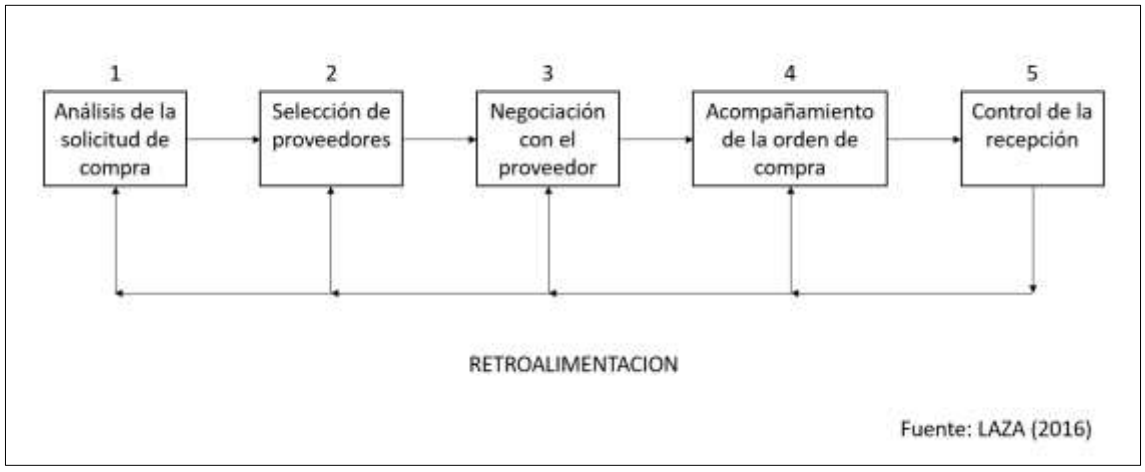


Figura 1: Ciclo de vida de compra, Laza (2016)

En donde nos enfocamos en el punto 4, seguimiento de la orden de compra (OC), necesitando definir ¿qué es una OC? para eso Palacio (2002) nos define que es un documento en donde se acuerdan términos de una negociación, dichos términos deben definir una serie de condiciones con el fin de evitar futuros reclamos por malas interpretaciones.

Entonces podemos decir que la orden de compra es un proceso que un pequeño comerciante no logra utilizar y aprovechar, y queremos mejorar con la tecnología donde Ruiz (2019) menciona que la evolución de las tecnologías ha cambiado el concepto o experiencia en la compra al por menor y que cada vez el consumidor requiera menos esfuerzo y tiempo en sus compras, esto quiere decir que los comerciantes deben evolucionar con los requerimientos de sus clientes mejorando sus procesos para brindar un mejor servicio.

Vera (2018) destacó que el mundo actualmente se encuentra inmerso en la tecnología, a tal grado de permitir la delegación de tareas indeseadas, pero que aún son necesarias para las aplicaciones, donde la tecnología móvil ha facilitado las transacciones diarias comunicaciones e interacciones. Estas tecnologías móviles les facilitan a los pequeños comercios administrar sus procesos de manera eficaz y efectiva.

La justificación es fundamentada con investigaciones sobre las aplicaciones móviles donde Pérez et al. (2018) concluyó que las tecnologías móviles son las más utilizadas debido a su bajo costo y simple uso (Pérez et al., 2018, p. 15).

También podemos ver que no existe las limitaciones de arquitecturas ni sistemas operativos en las aplicaciones donde Espinoza-Galicia et al. (2017) demostró que las aplicaciones web, móvil y de escritorio pueden convivir de una manera sólida y separada, implementando API-RESTfull del lado del servidor y puedan otros sistemas con cualquier lenguaje acceder al mismo. Por lo que se optó por utilizar api-rest como arquitectura para unificar el sistema que ya disponen los usuarios para el manejo o administración de sus negocios, dando así la libertad de elegir el lenguaje de programación para el front-end.

Vera (2018) quien concluye que las tecnologías móviles facilitan las tareas diarias de las personas, permitiendo desarrollar tareas tediosas, generando experiencias positivas en sus vidas. (Vera, 2018, p. 16), demostrando que la opción de desarrollar una aplicación móvil permitirá la reducción de costo en la implementación de la aplicación en los usuarios, por el hecho de que cualquier persona tiene acceso a un dispositivo móvil.

Challiol et al. (2017) diseñó un framework conceptual que permite diseñar aplicaciones móviles basada en el posicionamiento, donde nos permite crear un espacio de discusión sobre la creación de este tipo de aplicaciones, el cual mostramos en la siguiente figura.

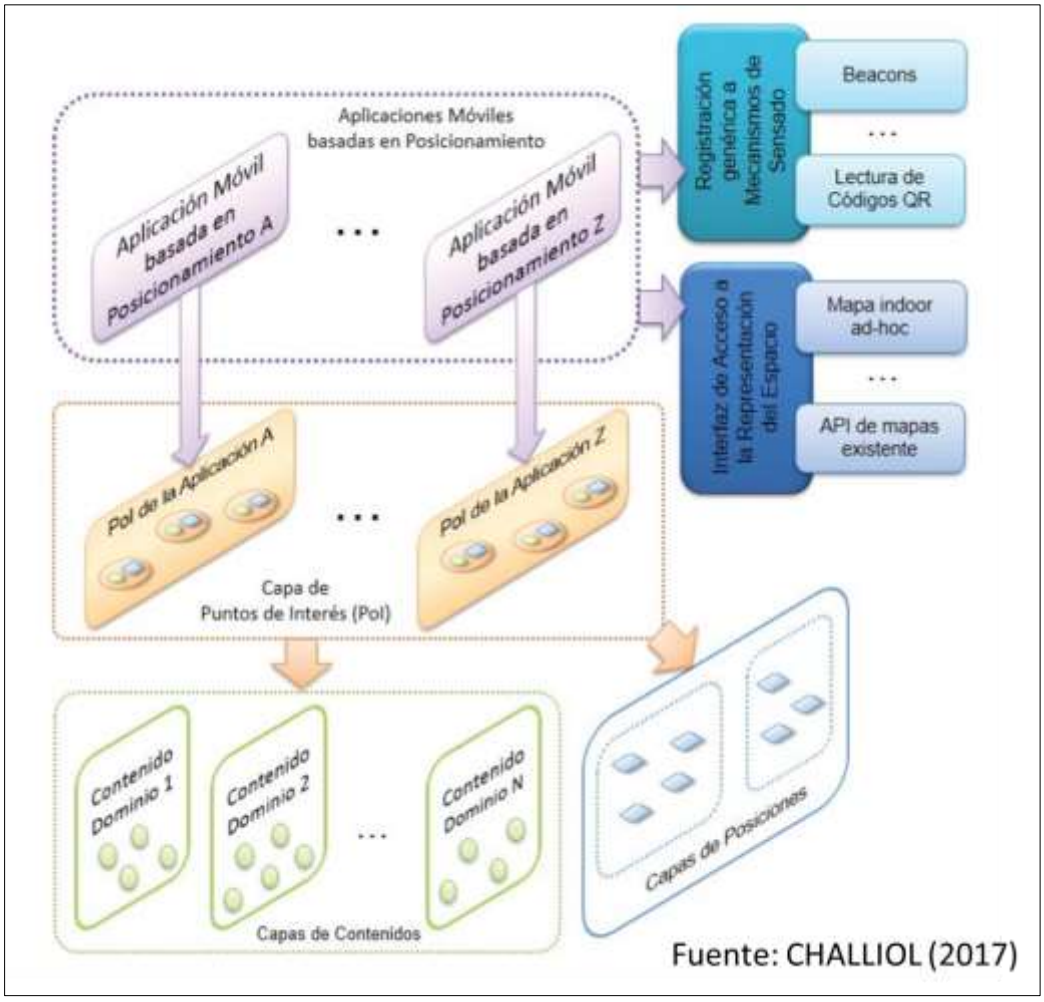


Figura 2: Framework conceptual Challiol et al.(2017)

Sangama (2020) hizo un estudio sobre las características de las “metodologías ágiles” comparando siete metodologías (HME, Scrum, Mobile-D, XP, SLeSS, MASAN y Scrumban) concluyendo que solo scrum, extremen programming (XP) y scrumban las que mejor resultado obtuvieron centrados en el objetivo de obtener software funcional y de calidad en el menor tiempo posible, con la capacidad necesaria para atender las exigencias especiales del desarrollo de las aplicaciones móviles. (Sangama, 2020, p. 17), por lo que se pensó en SCRUM como la metodología que permitirá desarrollar el sistema api-rest como también el sistema para la aplicación móvil.

Tomando conocimiento de la problemática se estableció como problema general el mejorar una parte del proceso de compra de las pymes, adaptando a su realidad actual. Donde se planteó tres problemas específicos a resolver:

- **PE1:** ¿Cómo podemos mejorar el proceso de órdenes de compra?
- **PE2:** ¿Se podrá causar un efecto de satisfacción en el uso de la aplicación móvil?

El objetivo general fue mejorar el proceso de compra, reduciendo el tiempo que toma realizarlo, es así que podemos brindar accesibilidad y disponibilidad de los datos. Donde se planteó tres objetivos específicos:

- **OE1:** Determinar si es posible mejorar el proceso de orden de compra de las mypes, reduciendo el tiempo del mismo, implementando la aplicación móvil.
- **OE2:** Determinar si es posible aumentar el nivel de satisfacción con respecto al manejo del aplicativo móvil sobre la gestión de orden de compra de las mypes.

La hipótesis general de la investigación fue: “Se mejoró el proceso de compra de las mypes, reduciendo el tiempo del mismo, causando satisfacción en el uso de la aplicación móvil” donde Calva et al. (2020) logró demostrar que la aceptación de sus usuarios en la aceptación de dispositivos móviles, logrando obtener que el 91.59% de sus usuarios disponen de dispositivos móviles.

Con respecto a las hipótesis específicas, detallamos a continuación:

- **HE1**: Implementado la aplicación móvil mejoró el proceso de compra, reduciendo el tiempo del mismo, sobre la gestión de orden de compra en las mypes.

  Fernández Alonso et al. (2020) menciona que las api-rest son ya un estándar que todas las empresas están utilizando para crear un servicio, por ejemplo, google, facebook, mercado libre, etc., en gran parte por la eficiencia y facilidad de implementación.

- **HE2**: Se promovió la satisfacción con respecto al manejo del aplicativo móvil sobre el proceso de orden de compra en las mypes.

  Rivera (2020) concluye en su investigación, que la reducción del tiempo en el proceso permite generar satisfacción en el cliente. También menciona que, mediante el uso de aplicativos móviles, este nos puede ayudar en la reducción de tiempo del proceso de gestión.

II. MARCO TEÓRICO

En este capítulo mencionamos los detalles que conlleva desarrollar la aplicación móvil usando arquitecturas de api-rest para mypes para mejorar el tiempo de las órdenes de compra, divididos en tres secciones; primero sobre trabajos previos, segundo sobre teorías relacionadas y tercero sobre marcos conceptuales. Para eso se realizó una búsqueda de información confiable. Con respecto a la primera sección (trabajos previos), se detallan los trabajos encontrados hasta la fecha de las distintas asignaturas y áreas. En la segunda parte del capítulo mencionamos todas las teorías que se encuentran relacionadas con la investigación y profundizan las teorías relacionadas al proceso de orden de compra y como se puede subdividir este proceso para implementarlo en los servicios api-rest. Y para la tercera sección (marcos conceptuales) se describe las distintas fuentes que permitieron construir el marco conceptual de la investigación basado en la variables y dimensiones.

La primera sección del capítulo correspondiente a “trabajos previos”, mencionamos estudios referentes a la implementación de aplicaciones para dispositivos móviles basándose en arquitecturas api-rest para el proceso de órdenes de compra de las pymes. Los estudios de estas investigaciones permitieron comparar metodologías y herramientas de desarrollo como la seguridad de la implementación de api_rest.

Tovar et al. (2021) desarrolló una aplicación móvil que permite realizar pedidos de comida a domicilio, mediante encuestas midió la satisfacción del administrador y sus clientes en el manejo de la aplicación, dando como resultado el renombre del restaurante como también su difusión en la comunidad y la satisfacción de los usuarios al no salir de casa. Asimismo, Tovar et al. (2021) recomendó el uso de otros framework para evaluar el rendimiento y también la difusión de servicios como firebase para tercerizar la gestión de manejo de servidores.

Escobar (2021) en su investigación, propuso un diseño de aplicación que permita desarrollar un asistente de compra en un centro comercial usando las tecnologías móviles. Escobar (2021) usando una metodología Lean Startup la cual permite ahorrar tiempo y dinero al enfocarse en los datos de la primera fase o etapa del proyecto, tomando como muestra a 77 personas mayores de edad y

con diferentes estudios, concluyendo que la transformación digital nos da la necesidad de adaptarnos a los continuos cambios dando como resultado una aplicación escalable que permita un crecimiento exponencial frente al gasto que crece de manera lineal. Escobar (2021) recomendó que, mediante la propuesta del diseño del prototipo funcional, este permitirá continuar con el desarrollo de una aplicación.

Párraga et al. (2021) implementó una aplicación móvil con el propósito de mejorar las ventas y distribución, usando como metodología de desarrollo a XP. Párraga et al. (2021) concluyó que las metodologías ágiles permiten controlar y gestionar un proyecto, siendo esta de manera iterativa e incremental. Párraga et al. (2021) nos recomienda que utilicemos las metodologías ágiles para el desarrollo del aplicativo móvil, garantizando el cumplimiento de metas o tareas.

Vicente et al. (2020) el propósito de la tesis es desarrollar una aplicación móvil que nos deje compartir la búsqueda de clientes o proveedores, esto quiere decir que funcionaría como una red social de clientes y proveedores. Para la valoración de los resultados de la implementación de la aplicación se utilizó una escala de valorización de tipo Liker, donde se realizó una encuesta para determinar el mercado hacia donde se está enfocando la investigación, esta encuesta determinó que el 41.2% del total encuestado se encuentran en un rango de 18 a 25 años de edad, también se realizó una valoración del 1 al 5 para el manejo de dispositivos móviles, arrojando que un 44% calificó con un 4 y un 7% con un 5, dándonos a conocer que la gran mayoría tiene un gran manejo de estas tecnologías. Vicente et al. (2020) concluyó que pudieron realizar una aplicación móvil, simulando una red social donde los actores son los clientes y/o proveedores, también que hay mucho camino en la modernización, abriendo el camino al desarrollo de nuevas aplicaciones para la difusión de estas tecnologías. Vicente et al. (2020) recomendó que para futuras investigaciones se pueda agregar la funcionalidad de chat entre clientes y proveedores.

Calva et al. (2020) hizo un estudio sobre el problema que tienen los usuarios de supermercados en el momento de seleccionar sus productos, con la finalidad de resolver este problema planteó desarrollar dos aplicaciones que permitan gestionar este problema, primero una aplicación móvil que permita en la

ubicación de productos y segundo una página web que permita gestionarlo. Calva et al. (2020) para lograr medir el grado de satisfacción, desarrollaron un prototipo al que le generaron una encuesta para medir el criterio de aceptación arrojando que a) el 91.59% de las personas encuestadas usan dispositivos móviles, b) 48.2% le es difícil de ubicar los productos, c) el 42.2% del gustaría contar con esta aplicación y d) que un 39% les gustaría tener recibir ofertas o promociones. Calva et al. (2020) llego a la conclusión que los usos de estas tecnologías ayudan a los supermercados en la satisfacción de sus clientes, mejorando así la experiencia de usuario y que el desarrollo de un cuadro de mando permite observar el comportamiento de los clientes. Calva et al. (2020) recomendó para futuras investigaciones, la implementación de un módulo de inventario para gestionar el abastecimiento de sus productos, establecer credenciales de acceso para las personas que van a gestionar el tablero de mando y, por último, renovar el servidor donde está alojada la base de datos y el sistema web, esto con la finalidad de mantenerlo siempre disponible.

Condori et al. (2021) hace un estudio sobre la viabilidad de implementar un plan de negocio llamado “mi mercado”, que mediante las aplicaciones móviles permitirá a los mercados de abastos brindar sus productos sin que el cliente salga de casa. Condori et al. (2021) para poder implementar el plan de negocio realizó entrevistas a expertos sobre el abastecimiento de productos, donde se consultó a quince expertos de diferentes rubros. También se realizaron encuestas a clientes para conocer las preferencias y hábitos que tienen en le de consumo de bienes, servicios y uso de la tecnología. Condori et al. (2021) obtuvo como resultado de las encuestas, que para el modelo de negocio que propusieron, tanto como para los comerciantes de abastos como para los clientes les resulta beneficioso el uso de aplicaciones móviles para realizar esta actividad desde la seguridad de sus casas u oficinas. Concluyendo que el plan de negocio propuesto beneficia gran parte de la sociedad al dar mayor alcance a los comerciantes de abastos en estos tiempos de pandemia y que los clientes puedan tener mayor seguridad en el momento que puedan realizar sus actividades diarias con el manejo de aplicaciones móviles. Condori et al. (2021) recomienda aplicar blockchain a los procesos de seguimiento de productos, realizar la ampliación de este modelo a mas ciudades del país, analizar

tendencias de los productos, mercados y proveedores para generar mayor rentabilidad, revisar el proceso del delivery cada 6 meses buscando mejorar este proceso.

Chira et al. (2021) implementó una aplicación móvil para las ventas electrónicas, con el objetivo de ampliar el estado de confort de los clientes en el momento de comprar los productos de una minimarket, conociendo sus hábitos y preferencias. Chira et al. (2021) realizó encuestas sobre los usuarios de la aplicación para medir la usabilidad y nivel de fiabilidad de la aplicación dando como resultado que el 85% de las personas calificó con siete y un 10% calificó por debajo de un 5% con respecto a la usabilidad. Con respecto a la fiabilidad marcó que un 43% de personas que son menores a los 50 años de edad muestran inseguridad con respecto al aplicativo móvil. Chira et al. (2021) concluyó que la aplicación móvil ayudará en el proceso de compra, según el índice de satisfacción que mostró el cliente, además se observó que las personas mayores a 50 años tienen a resistirse a las nuevas tecnologías.

Domingo (2020) planteó el desarrollo de una aplicación móvil con arquitectura api_rest, con la finalidad de aprender las tecnologías que se usan para su implementación. Esta aplicación crea perfiles de tiendas que puedan publicar sus productos con ofertas y promociones para captar más clientes. Domingo (2020) concluyó que el aprendizaje del desarrollo de apis y aplicaciones móviles y otras tecnologías como php, mysql, y html fue exitoso. Recomendando que para complementar este proyecto sería agregar una capa más de seguridad, implementando HTTPS, además de la interfaz gráfica del aplicativo móvil, con el proposito de ser más intuitiva, otra mejora a recomendar sería la búsqueda de productos mediante códigos de barras.

Onecha (2020) desarrolló un aplicativo móvil que permitió la compra y venta de productos de segunda mano especializados en el rubro de la música. Como parte de esta investigación Onecha (2020) analizó los diferentes dispositivos móviles en el mercado tanto como sistemas operativos y versiones, con el propósito de llegar a la mayor cuota del mercado, quienes son los jóvenes entre 20 a 30 años de edad. Onecha (2020) llegó a la conclusión de que el sistema operativo más distribuido es android con la versión 5.0 lollipop y que la experiencia

desarrollando la aplicación fue satisfactorio tanto en tiempo como en funcionalidad. Onecha (2020) recomendó agregar la funcionalidad que se encargue de la gestión logístico y otro que permita la suscripción de alertas personalizadas.

Quirama (2019) diseñó y desarrolló una aplicación móvil que da a las pymes; listar, salvar y analizar una gran cantidad de datos referentes a sus posibles clientes con la finalidad de procesar estos datos se obtenga más ventas y experiencia de usuario (p.4). Concluyendo que se pudo aplicar el conocimiento aprendido y sumar con el conocimiento obtenido gracias a los desafíos que llevaron a este proyecto (Quirama, 2019, p.60).

Triviño (2020) realizó una aplicativo móvil para la venta online de articulos farmacéuticos, analizando la situación de la farmacéutica y la satisfacción de sus clientes, proponiendo estrategias de competitividad y herramientas tecnológicas (p.5-6).Triviño (2020) para la medición de la satisfacción, se realizó una encuesta a los clientes de la farmacéutica, registrando los siguientes resultados; primero que el 100% de los clientes poseen un dispositivo móvil, segundo que el 64% lo usa como entretenimiento, tercero que el 72% ha realizado algún tipo de compra online, cuarto que la mayoría con un 92% cree que es importante realizar compras desde los dispositivos móviles. Concluyendo según el diagnóstico de la farmacéutica, es necesario implementar una aplicación móvil que permita realizar las ventas online. (Triviño, 2020, p.110). Triviño (2020) recomendó agregar la funcionalidad de leer códigos QR para la búsqueda de productos.

Salas (2020) investigó la seguridad de un api_rest, donde se planteó como objetivo implementar un api_rest utilizando los verbos http, investigar el funcionamiento de JSON Web Tokens y la seguridad que brinda y por ultimo investigar el uso de Kong como GateWay. Salas (2020) para lograr su investigación implementó en un servidor Linux; Node.js PM2, JWT y MongoDB. Salas (2020) concluyó que aplicar JSON Web Tokens le da una capa de seguridad a las apis, pudiendo autenticarse cada vez que se consuma el api, además el uso de Kong permitió tener un mayor control de acceso a las apis con su dashboard integrado. Salas (2020) recomendó para futuras investigaciones, el estudio de la librería JOSE para Node.js ya que esta librería contiene los

estándares JWT, JWS y JWE, y por ultimo investigar los plugins logging, acl, rate limiting, proxy cache, analytics y monotoring en el gateWay Kong.

Hernández (2021) investigó el uso de diferentes tecnologías o arquitecturas que usan las empresas para desarrollar sus aplicaciones web, donde se propuso demostrar, cuál de estas cumple con las expectativas de las empresas. Para esto consideró estudiar al sector privado de la república mexicana, donde la información se manejó de manera parcial y objetiva para el cálculo de las muestras. Hernández (2021) se planteó como objetivo, analizar tres tecnologías o arquitecturas (REST, SOA, GraphQL), dando como resultado que durante los últimos 10 años REST se encuentra casi en la mayor parte de empresas. Hernández (2021) concluyó que aun las grandes empresas siguen implementadas SOA y no invierten en una arquitectura con mayor rendimiento e integración como es REST, y a pesar que la gran mayoría usa REST, GraphQL está tomando mayor importancia en las empresas.

Hernando (2021) analizó la seguridad del api rest, donde se propuso analizar las opciones que existe en la seguridad de las apis. Frente a las amenazas que existen actualmente en las implementaciones de las apis rest se propuso estudiar las diferentes capas de seguridad de las mismas. Hernando (2021) se planteó como objetivo estudiar los siguientes puntos; ¿qué es una api? gobierno de una API REST, Api Managers, productos de API Management o API Gateway (Kong,TYK, WSO2 api manager). Hernando (2021) concluyó que las Api Management son una herramienta muy útil para el gobierno y protección de las apis, WSO2 api manager es un producto funcional, ya que nos da muchas opciones de configuración y a pesar que HTTPS es una capa de seguridad que resuelve muchos problemas, la combinación con JWT (JSON Web Tokens) hacen una opción más escalable sin perder la robustez y que OAUTH nos brinda autenticación y autorización para las apis rest. Hernando (2021) recomendó para futuras investigaciones, probar otros mecanismos de autenticación y autorización, también apoyar la trazabilidad de WSO2 api manager con alguna herramienta de gestión de eventos.

Ruiz (2019) observando a los pequeños comerciantes y la experiencia de sus clientes, planteó una propuesta de desarrollo de una aplicación móvil que permita

una atención más completa y personalizada, esta aplicación también tendrá la posibilidad de manejar los pedidos y gestionar las entregas. Ruiz (2019) concluyó que las ayudas de las metodologías ágiles permiten desarrollar aplicaciones escalables y de gran impacto para el usuario y dejando gran expectativa para su implementación.

En esta segunda sección del marco teórico, correspondiente a las teorías relacionadas, se buscó apoyar la investigación con los conceptos de gestión de compras, seguridad en el api rest, marcos de trabajo. Para cumplir con el objetivo de disminuir el tiempo del proceso de compra, es importarte conocerlo conceptualmente y como este cambia en las pymes, también tenemos que aprender sobre marcos de trabajo que se ajustan más a las condiciones del objetivo.

Romero (2013) define a la orden de compra como un instrumento que realizan las empresas, este instrumento es unilateral y posee un valor jurídico, el cual es emitido por un comprador hacia un vendedor, en este documento se plasma los productos y/o servicios, donde y cuando, y el valor de esta transacción.

Parrales (2017) nos menciona el importante rol que cumple el proceso de compra en todas las instituciones, donde el análisis constante de la actividad de cada proceso permite reaccionar a las falencias y plantear soluciones, concluyendo que la estructura organizacional es de vital importancia, como la elección de un buen líder y por último el poseer un manual de funciones y/o procedimientos que enmarque el proceso de orden de compra.

Fernandez (2020) nos conceptualiza que es una arquitectura que permite definir cualquier interfaz entre distintos sistemas. El termino REST proviene de sus siglas en ingles Representacional State Transfer y se dio a conocer en una tesis doctoral en el 2000 por Roy Fielding (Fernandez, 2020, p. 3).

El beneficio de un api rest propia o de terceros, permite ampliar el ámbito de las organizaciones, podemos poner el ejemplo de las redes sociales como facebook quienes mediante esto pueden llegar a más personas y a las organizaciones les permite poder seguir a sus clientes. (Fernandez, 2020, p. 7).

¿Qué es un API? Application Programing Interface traduciéndolo sería una interfaz de programación de aplicaciones, este concepto surgió a principios de la informática, donde se usaba como una biblioteca para los sistemas operativos, permitiendo comunicar a dos aplicaciones de manera local, a principios de los 2000 ya eran una tecnología que permitía integrar datos de manera remota. Actualmente se puede conceptualizar como un conjunto de definiciones y protocolos que permiten desarrollar aplicaciones que puedan comunicarse con otros sistemas. (Fernandez, 2020, p.14).

¿Qué metodología usar? Si bien existen muchas metodologías o marcos de trabajo, necesitamos elegir el correcto para nuestra investigación, donde Sangama (2020) hace una comparación de las metodologías, donde destaca a las metodologías ágiles y dentro de ellas a Scrum, XP y Scrumban que mejores resultados brindan con respecto a los objetivos funcionales, generando calidad en el producto y en el menor tiempo.

Sangama (2020) nos define a scrum como un marco de trabajo que realiza iteraciones en toda la vida del desarrollo del software, enfocándose en entregables testeables. También define que las historias de usuario son el núcleo de scrum, donde se engloban en sprint de corto plazo y son inmutables, esto quiere decir que un sprint no tiene modificaciones.

En esta tercera sección del marco teórico, correspondiente al marco conceptual, se trata de conocer la importancia que debemos generar en el usuario de la aplicación.

Fernandes (2020) en su investigación concluyó que es importante brindar al cliente una experiencia de compra que interactúe con otras atmosferas digitales, adicionando la facilidad de uso, contenido informativo, confianza y señales de entretenimiento, aumentando el valor percibido por el usuario. Este valor creado aumenta la satisfacción el uso de la aplicación.

III. METODOLOGÍA

Para el desarrollo de este documento, se definió como aplicada la investigación, con una perspectiva cuantitativa y de diseño pre-experimental. Teniendo como variable la mejora del proceso de compra, reduciendo el tiempo y la satisfacción que genera el uso del mismo. La muestra tuvo 30 comercios (pymes) en el rubro de la indumentaria; además para recolectar los datos, usamos la técnica Liker y la razón con el objetivo de lograr cuantificar las dimensiones de tiempo y satisfacción respectivamente, planteados en la hipótesis.

3.1 Tipo y diseño de investigación

Donde la presente investigación fue de tipo aplicada. Este tipo de investigación pretende promover o generar conocimiento de manera directa y en un mediano plazo en la sociedad. Con este tipo de investigaciones logramos generar un valor agregado con respecto al manejo de conocimiento proveniente de la investigación básica. Gracias a esto podemos generar progreso en el sector productivo y rentabilidad por la diversificación. Mejorando de manera directa en el nivel de vida de la comunidad (Lozada, 2014, p.35).

Las investigaciones de tipo aplicada ayudan a las organizaciones en sus decisiones al brindar información y conocimiento. Estas experimentan situaciones que afectan el normal desarrollo de sus actividades y otras que favorecen o impiden el crecimiento de la actividad. Dichas situaciones son conceptualizadas como problemas de acción. En la medida en que las situaciones son conceptualizadas como problemas, la adecuada definición de estos problemas es clave para poder encontrar soluciones (Baudean, 2015, p.5)

De acuerdo a lo mencionado inicialmente, podemos decir que esta investigación es de tipo aplicada. Porque se pretende mejorar los tiempos en la generación de órdenes de compra usando la aplicación móvil con arquitectura api-rest y con ello agilizar sus procesos logrando la satisfacción en el usuario.

Esta investigación tiene enfoque cuantitativo. Hernández y Mendoza (2018) mencionan que para estimar las ocurrencias o magnitudes de fenómenos es apropiado tener un enfoque cuantitativo para probar una hipótesis (p.6). Para ello se recolectó una serie de datos los cuales han sido analizados mediante las

técnicas de este enfoque, utilizando con ello diferentes herramientas estadísticas, los cuales permitió probar las hipótesis establecidas.

La investigación experimental, Murillo (2011) nos dice que el autor de una investigación llega a manipular una o más variables de la investigación, con el propósito de dirigir el aumento o disminución de las mismas y el impacto en su comportamiento observado. Esta parte se desarrolla estrictamente en condiciones preparadas, con el propósito de detallar la causa o el modo que se comporta en una situación o momento en particular (Murillo, 2011, p.5)

En la investigación pre-experimental existe un reducido grado de control, ya que se trabaja con un solo grupo y no se asignan aleatoriamente las unidades de análisis. Este diseño consta en administrar un estímulo o tratamiento en modo de posprueba o en una preprueba correlativamente a una posprueba conjuntamente (Baray, 2006, p.69).

En el desarrollo de la investigación se realizó una preprueba y una posprueba correspondiente al manejo del aplicativo móvil con arquitectura api-rest a un pequeño grupo de mypes con el objetivo de probar, si se cumple las hipótesis formuladas en la investigación.

3.2 Variables y operacionalización

Definimos como variable de estudio al efecto de utilización aplicativo móvil para reducir el tiempo del proceso de compra mejorando el mismo, Observando el Anexo 4 “la matriz de operacionalización de la variable”. Podemos detallar lo siguiente:

1. Definición conceptual:

   Actualmente el mundo se encuentra inmerso en la tecnología, donde la tecnología móvil ha facilitado las iteraciones, comunicaciones y transacciones diarias del ser humano. Estas tecnologías móviles les facilitan a los pequeños comercios a administrar sus procesos de manera eficaz y efectiva. Justificando el bajo costo y simple uso de los dispositivos móviles. (Vera,2018, p.5; Pérez et al.,2018, p.15)

2. Definición operacional:

   La idea principal fue mejorar el proceso de orden de compra, reduciendo el tiempo y la satisfacción del proceso. Para lograr este objetivo, se fragmentó el proceso de orden de compra y se implementó un servicio rest, con el propósito de extender la funcionalidad en un dispositivo móvil que permita agilizar los trámites de orden de compra. Para esto se analizó los resultados a través de encuestas.

3. Dimensiones:

   - Tiempo: Los clientes valoran el tiempo y la seguridad que pueden ganar al realizar compras, mejorando el proceso de estos. (Condori et al. 2021, p.55; Chira et al. 2021, p30).

   - Satisfacción: Los dispositivos móviles son una tecnología que nos permite realizar tareas diarias de manera fácil y eficientemente, brindando satisfacción al ser intuitivo y de mayor distribución en el globo. (Calva et al.,2020, p.124; Condori et al., 2021, p.53; Chira et al., 2021, p.30; Triviño,2020, p.110).

4. Indicadores:

   - Reducción de tiempo (Rivera, 2020, p.17)

   - Incremento de satisfacción (Fernández, 2020, p.17)

5. Escala de medición:

- La Razón: Este tipo de medición es uno de los más completos, teniendo en cuenta que maneja una escala de intervalos dentro de sus propiedades, y también incluyendo el cero absoluto. Donde el valor cero no es arbitrario, ya que representa la ausencia total de la magnitud que se está midiendo. Con este tipo de escala nos permite lograr cualquier operación lógica, como las operaciones aritméticas, la comparación y el ordenamiento. A iguales diferencias entre los números asignados corresponden iguales diferencias en el grado de atributo presente en el objeto de estudio. Ejemplos: peso, longitud, distancia, precios, ingresos (Orlandoni, 2012, p.246).

- Likert: Elejabarrieta (2008) nos dice que esta escala es una de las más usadas para la medición de actitudes, siendo un método sencillo por ser simple en la construcción y aplicación. Esta técnica, además de ubicar al individuo en un punto determinado, lo que es muy común en otras escalas, poniendo énfasis en la cuenta de amplitud y la consistencia de las respuestas actitudinales (p.25).

3.3 Población, muestra y muestreo

Arias et al. (2016) nos explica que la población de la investigación es un grupo de casos que se encuentran definidos, limitados y accesibles, donde el investigador agrupará y seleccionará la muestra, y que cumple con una serie de criterios terminados. Es indispensable definir la población a estudiar dado que, al culminar la investigación basado en la muestra, nos permitirá generalizar o extrapolar los resultados obtenidos en la investigación hacia el total de la población (p.202).

Para esta investigación, se estableció una población de 257,280 mypes cuya actividad es el comercio en la zona de Lima (Instituto Nacional de Estadística e Informática, 2018).

Para Hernández et al. (2017) la muestra esencialmente pertenece a una cantidad pequeña de la población. Donde podemos definir a la muestra como un grupo de subconjunto de elementos con características pertenecientes a la población (p.175).

La técnica de muestreo fue no probabilística, usando el método por conveniencia, dicha muestra estuvo constituida por un grupo de 30 mypes cuya actividad es el comercio. Al respecto Arias et al. (2016) nos dice que la selección de la muestra se realiza de manera aleatoria donde las características son semejantes a la población objetivo. Otra opción que tiene el investigador es la posibilidad de seleccionar de forma directa e intencionadamente los elementos de la población. Los casos más frecuentes que donde el investigador selecciona su muestra es cuando se tiene un mejor acceso a estos. En general, este método es muy útil cuando se pretende investigar un fenómeno de una población o cuando no existe un tamaño de muestra definido (p.206).

Una vez definida la población, se debe detallar las condiciones o criterios que los participantes deben cumplir. Arias et al. (2016) nos dicen que estos criterios deben detallar las características que la población debe tener. Estos criterios son los que van a delimitar a la población elegible (p.204).

Criterio de inclusión:

- MYPES
- Pertenezcan a la actividad del comercio
- Que colaboren con la investigación
- Que cuenten con un sistema comercial
- Que posean dispositivos móviles con el sistema operativo Android con versiones superiores a Lollipop.

Criterios de exclusión:

- Quienes no completen adecuadamente las encuestas previas.
- Quienes no tengan dispositivos móviles con el sistema Android.
- Quienes ya definido una gestión de compras dentro de su empresa.

3.4 Técnicas e instrumentos de recolección de datos

En la presente investigación se utilizó como técnica de recolección a la encuesta y como instrumento se utilizó el cuestionario. Morone (2013), nos dice que los instrumentos y procedimientos son las técnicas que utilizaremos para poder obtener conocimiento. Entrevistas, encuestas, observaciones y todo lo que se deriva de ellas (p.3).

Para el investigador el instrumento de medición es un recurso que permite recolectar datos o información obtenidos de la variable de estudio (Hernández et al., 2014, p.199). Existen diferentes tipos de instrumentos, para esta investigación se aplicó el cuestionario. Para Hernández et al. (2014) el cuestionario es un grupo de preguntas basadas en una o más variables, el cual tiene que ser consecuente con lo planteado en la hipótesis o el problema a resolver (p.217).

Esta investigación usó la escala Likert para poder medir la variable, donde Hernández et al. (2014) nos demuestra que esta escala es un conglomerado de afirmaciones que permiten medir la reacción del sujeto en tres, cinco o siete categorías (p.238). La finalidad de utilizar esta escala, fue de obtener los datos respecto al grado de satisfacción con los aplicativos móviles basado en una pregunta con tres alternativas: (1) poco, (2) regular y (3) mucho.

A partir de lo mencionado líneas arriba, en esta investigación se utilizó la validez de contenido. Este refiere al juicio sobre el grado que el instrumento representa la variable objeto de medición, esto quiere decir, el nivel de representación del universo con respecto a la variable objeto de estudio (Bernal, 2010, p.248)

Respecto al nivel de confianza, Hernández et al. (2014) nos aclara sobre la confiabilidad que posee un instrumento de medición radica en los resultados iguales que genera un mismo individuo en la aplicación repetida del mismo (p.200), que para lograr alcanzar el nivel de confianza deseado es necesario el complemento de error máximo aceptable, porcentaje de acierto en la representatividad de la muestra (p.179). Podemos decir entonces que el intervalo

de confianza de esta investigación es del 95%, el cual es el más utilizado sobre los indicadores para las pruebas estadísticos.

3.5 Procedimientos

Se seleccionó a 30 empresas mypes cuya actividad es el comercio, que estén dispuestos a participar de la investigación, los cuales deben tener dispositivo móvil el cual es una herramienta esencial para el estudio.

El aplicativo móvil para mejorar el proceso de órdenes de compra, ayudo a la mejora significativa de sus procesos, ya que el usuario interactuó de manera directa. Esto ocasiono que exista un gran estímulo permitiéndole reducir sus tiempos en la ejecución de este proceso y con ello optimizar sus recursos. Por ello el tener la información de los productos on demand permitieron mejorar los tiempos de respuesta del comerciante en la calle en tomar decisiones con respecto a los productos que compraron.

En relación a la medición de los indicadores reducción de tiempo e incremento de satisfacción, se aplicó un cuestionario antes y después del uso del aplicativo móvil, con la finalidad de corroborar los datos obtenidos y con ello realizar la validación de la hipótesis.

Detalle de los pasos ejecutados:

- Consentimiento firmado por parte del usuario
- Aplicación del cuestionario preprueba
- Instalación y registro del aplicativo móvil en el móvil del usuario
- Aplicación del cuestionario posprueba

3.6 Método de análisis de datos

Para poder realizar el análisis de los datos se consideró los niveles de la medición de la variable y usando herramientas estadísticas para obtener los resultados (Hernández et al.,2014, p.271). Donde 30 empresas fueron la muestra, a quienes se les realizó un cuestionario antes y después de utilizar el

aplicativo móvil, cuyo intervalo de confianza es del 95% con un error aceptable de 5%.

La elección del test para nuestra investigación fue el de shapiro wilk el cual se enfoca en el contraste de la normalidad de un conjunto de datos. Este tipo de test se aplica para muestras inferiores a 50. Este tipo de prueba detecta en la normalidad desviaciones debido a la asimetría o curtosis (Razali, 2011, p,25)

Para realizar el análisis de datos se seleccionó el programa IBM SPSS (Statistical Package for the Social Sciences) y para probar la hipótesis se usó la prueba de Wilcoxon. Este tipo de prueba a menudo se usa para analizar medias o medianas de un par de conjuntos independientes, probablemente con distribución no normal, este tipo de prueba pertenece al grupo de pruebas estadísticas no paramétricas, es esencial la diferencia de dispersión de datos de un grupo con respecto a otro (Turcios, 2015, p.21)

3.7 Aspectos éticos

Todos los conocimientos aprendidos de las diferentes fuentes como libros, artículos científicos y otros estudios relacionados con la investigación, se utilizó la normativa ISO 690:2010 para las citas.

La investigación es de autoría propia presentando originalidad en su redacción, el cual acata con lo descrito en el artículo 9 del Código de Ética de Investigación 2020 de la Universidad César Vallejo, en donde hace mención a la política de anti plagio, y para ello la investigación se pasa por un software para determinar su similitud en relación a otros trabajos de investigación ya existentes. Hago mención que existe total transparencia en los datos obtenidos, así como el respeto por la propiedad privada.

Se hace mención que todos los procedimientos realizados en esta investigación han respetado lo estipulado por el Colegio de Ingenieros del Perú en lo referente al Código de Ética, en el artículo 18 menciona que se debe respetar las leyes y disposiciones vigentes y se debe actuar con honradez y moralidad.

IV. RESULTADOS

Para el desarrollo de este cuarto capítulo, se describió detalladamente los resultados que se obtuvo en la investigación apoyándose en indicadores de “reducción de tiempo” e “incremento de satisfacción”. Posteriormente se realizó el procesamiento de los indicadores mencionados, donde se plantearon como cuestionarios. Y para finalizar se procedió al análisis con el programa IBM SPSS Statistics, debido a que la investigación es de tipo pre-experimental.

Prueba de la hipótesis específica 1

En esta prueba detallamos los datos estadísticos descriptivos de acuerdo a lo planteado en la prueba de entrada y salida.

Con estos datos obtenidos al implementar la aplicación móvil, me permitió medir el tiempo que les toma a los comerciantes o pymes realizar sus compras y poder observar la reducción del mismo.

Tabla 1: Indicadores estadísticos sobre la reducción del tiempo de la gestión de orden de compra utilizando la aplicación móvil.

Reducción de Tiempo			
		Estadístico	Error estándar
HE1 Prueba de entrada	Media	3,30	0,119
HE1 Prueba de salida	Media	2,67	0,121

Tabla 1 Indicador estadístico sobre la reducción del tiempo

En la tabla 1 podemos observar que hubo una reducción del tiempo con respecto a la gestión de las órdenes de compra en el momento de utilizar la aplicación móvil. Donde la media que arrojó la prueba de entrada es de 3.30 y el de la salida es 12.67, este resultado nos permite visualizar que los comerciantes presentaron mejores resultados con una reducción del tiempo del 23.60%.

A continuación, mostramos la formula aplicada.
RT = Reducción de tiempo
PS = Prueba de salida
PE =Prueba de entrada

$$\mathrm{RT} = \frac{(\mathrm{PE} - \mathrm{PS})}{\mathrm{PS}} \mathrm{x}\ 100\%$$

$$\mathrm{RT} = \frac{(3{,}30 - 2{,}67)}{2{,}67} \mathrm{x}\ 100\%$$

$$\mathrm{RT} = 23{,}60\%$$

Prueba de Normalidad

Debido a que contamos con 30 comerciantes como muestra, el método que optó para realizar la prueba de normalidad es la de Shapiro-Wilk.

Tabla 2: Prueba de normalidad sobre la reducción del tiempo de la gestión de orden de compra utilizando la aplicación móvil.

Shapiro-Wilk			
	Estadístico	gl	Sig.
HE1 Prueba de entrada	0,774	30	0,00002322
HE1 Prueba de salida	0,771	30	0,00002018

Tabla 2 Prueba de normalidad sobre la reducción del tiempo

En la tabla 2 podemos observar al método de Shapiro-Wilk de la prueba de entrada y la prueba de salida arrojándonos como resultado los siguientes datos estadísticos de 0.774 y 0.771 respectivamente.

Donde:

Prueba de entrada

Una vez obtenidos los resultados de la prueba de normalidad, se puede observar que el nivel de significancia es menor a 0.05, lo cual demuestra que la prueba no sigue una distribución normal.

Prueba de salida

Observando los resultados que arrojó la prueba de normalidad, podemos mencionar que el nivel de significancia es menor a 0.05, lo cual nos indica que la muestra no sigue una distribución normal.

Una vez obtenidos los resultados de la prueba de normalidad, se puede observar que el nivel de significancia es menor a 0.05, lo cual prueba que la muestra no sigue una distribución normal.

Prueba de hipótesis

Hipótesis específica HE1

H0: Implementar la aplicación móvil no mejoró el proceso de compra, reduciendo el tiempo del mismo, sobre la gestión de orden de compra en las mypes.

Ha: Implementado la aplicación móvil mejoró el proceso de compra, reduciendo el tiempo del mismo, sobre la gestión de orden de compra en las mypes.

Prueba de Wilcoxon

Como pudimos observar que la muestra no sigue una distribución normal, se aplicó la prueba no paramétrica Wilcoxon.

Tabla 3: Prueba de rangos con signos de Wilcoxon.

Rangos				
		N	Rango promedio	Suma de rangos
HE1 Prueba de salida - HE1 Prueba de entrada	Rangos negativos	16[a]	10,16	162,50
	Rangos positivos	3[b]	9,17	27,50
	Empates	11[c]		
	Total	30		
a. HE1 Prueba de salida < HE1 Prueba de entrada				
b. HE1 Prueba de salida > HE1 Prueba de entrada				
c. HE1 Prueba de salida = HE1 Prueba de entrada				

Tabla 3 Prueba de rangos con signos de Wilcoxon sobre el indicador de reducción del tiempo.

Tabla 4: Estadístico de prueba Z - reducción del tiempo de la gestión de orden de compra utilizando la aplicación móvil.

Estadísticos de prueba[a]	
	HE1 Prueba de salida - HE1 Prueba de entrada
Z	-2,804[b]
Sig. asin. (bilateral)	0,005
a. Prueba de rangos con signo de Wilcoxon	
b. Se basa en rangos positivos.	

Tabla 4 Estadístico de prueba Z sobre el indicador reducción del tiempo.

Analizando los datos en el sistema IBM SPSS y en el estadístico de prueba, dando como resultado -2.804, donde si el valor P (0.005) es inferior al nivel de significancia (0.05), es rechazada la hipótesis nula y se acepta la hipótesis alternativa con un nivel de confianza del 95%, esto nos quiere decir que "la aplicación móvil reduce en un 23.60% el tiempo de gestión de órdenes de compra en las pymes".

Prueba de la hipótesis específica 2

Para el análisis de esta segunda prueba se tomó la misma muestra de 30 comerciantes. Los cuales usando la aplicación móvil se le realizó una encuesta para poder medir la satisfacción del mismo. Esta encuesta se basó en dos preguntas y valoradas en el rango de; poco satisfecho (1), regular satisfecho (2) y muy satisfecho (3).

Tabla 5: Indicadores estadísticos sobre el incremento de satisfacción en la gestión de orden de compra utilizando la aplicación móvil.

Incremento de Satisfacción			
		Estadístico	Error estándar
HE2 Prueba de entrada	Media	1,43	0,092
HE2 Prueba de salida	Media	2,17	0,128

Tabla 5 Indicadores estadísticos sobre el incremento de la satisfacción.

Observando la tabla 5 podemos notar el incremento de la satisfacción del uso de la aplicación móvil. Arrojando en la prueba de entrada una media de 1.43 y la prueba de salida una media de 2.17, como podemos observar, se puede afirmar que los comerciantes presentaron mejores resultados con un incremento en la satisfacción del 51.75%.

A continuación, se muestra la aplicación de la formula.
IS = Incremento de satisfacción
PS = Prueba de salida
PE =Prueba de entrada

$$\mathrm{RT} = \frac{(\mathrm{PS} - \mathrm{PE})}{\mathrm{PE}} \text{ x } 100\%$$

$$\mathrm{RT} = \frac{(2.17 - 1.43)}{1.43} \text{ x } 100\%$$

$$\mathrm{RT} = \ 51.75\%$$

Prueba de Normalidad

Para poder realizar esta prueba, se utilizó a Shapiro-Wilk como método, debido al tamaño de nuestra muestra de 30 comerciantes.

Tabla 6: Prueba de normalidad sobre el incremento del nivel de satisfacción de la gestión de orden de compra utilizando la aplicación móvil.

Shapiro-Wilk			
	Estadístico	gl	Sig.
HE2 Prueba de entrada	0,632	30	0,00000018
HE2 Prueba de salida	0,800	30	0,00006617

Tabla 6 Prueba de normalidad sobre el incremento del nivel de satisfacción.

En la tabla 6 se utilizó el método de Shapiro-Wilk de la prueba de entrada y la prueba de salida obteniendo los siguientes datos estadísticos de 0.632 y 0.800 respectivamente.

Donde:

Prueba de entrada

Una vez aplicado la prueba de normalidad, se pudo apreciar que el nivel de significancia es menor a 0.05, lo cual prueba que la muestra no sigue una distribución normal.

Prueba de salida

Una vez aplicado la prueba de normalidad, se pudo apreciar que el nivel de significancia es menor a 0.05, lo cual prueba que la muestra no sigue una distribución normal.

Prueba de hipótesis

Hipótesis específica HE2

H0: El efecto del uso de la aplicación móvil no promueve la satisfacción en el proceso de orden de compra en las mypes.

Ha: El efecto del uso de la aplicación móvil promueve la satisfacción en el proceso de orden de compra en las mypes.

Prueba de Wilcoxon

Como pudimos observar que la muestra no sigue una distribución normal, se aplicó la prueba no paramétrica Wilcoxon.

Tabla 7: Prueba de rangos con signos de Wilcoxon – incremento de satisfacción de la gestión de orden de compra utilizando la aplicación móvil.

Rangos				
		N	Rango promedio	Suma de rangos
HE2 Prueba de salida - HE2 Prueba de entrada	Rangos negativos	1[a]	8,50	8,50
	Rangos positivos	19[b]	10,61	201,50
	Empates	10[c]		
	Total	30		
a. HE2 Prueba de salida < HE2 Prueba de entrada				
b. HE2 Prueba de salida > HE2 Prueba de entrada				
c. HE2 Prueba de salida = HE2 Prueba de entrada				

Tabla 7 Prueba de rangos con signos de Wilcoxon sobre el incremento de satisfacción.

Tabla 8: Estadístico de prueba Z – incremento de satisfacción de la gestión de orden de compra utilizando la aplicación móvil.

Estadísticos de prueba[a]	
	HE2 Prueba de salida - HE2 Prueba de entrada
Z	-3,841[b]
Sig. asin. (bilateral)	0,000
a. Prueba de rangos con signo de Wilcoxon	
b. Se basa en rangos positivos.	

Tabla 8 Estadístico de prueba Z sobre el incremento de satisfacción.

Se revisaron los datos con el sistema IBM SPSS y en el estadístico de prueba dio un resultado de -3.841 en donde, si el valor P (0.000) es menor al nivel de significancia (0.05), se rechazó la hipótesis nula y se aceptó la hipótesis alternativa con un nivel de confianza del 95%, la cual indica que “la aplicación

móvil incrementa en un 51.75% la satisfacción de la gestión de órdenes de compra en las pymes".

Prueba de la hipótesis general

En la aplicación de la prueba de normalidad, obtuvimos un nivel de significancia menor al 0.05, esto nos quiere decir que la muestra no está dentro de una distribución normal, por lo que podemos aceptar la hipótesis alterna.

HG0: El uso de la aplicación móvil no mejoró el proceso de compra de las pymes.

HG1: El uso de la aplicación móvil mejoró el proceso de compra de las pymes.

Resumen

Tabla 9: Resumen de los resultados de aceptación o rechazo de las hipótesis planteadas en la investigación:

Cód.	Hipótesis	Resultado (Aceptada o Rechazada)
HE1	Implementar la aplicación móvil reduce el tiempo de gestión de órdenes de compra en las pymes.	Aceptada
HE2	El efecto del uso de la aplicación móvil promueve la satisfacción en el proceso de orden de compra en las mypes.	Aceptada
HG	El uso de la aplicación móvil mejoró el proceso de compra de las pymes.	Aceptada

Tabla 9 Resumen de los resultados de aceptación o rechazo de la hipótesis.

De acuerdo a los resultados obtenidos se comprobó que las hipótesis alternas planteadas fueron aceptadas, cumpliendo con el objetivo genera y específicos. Generándose una reducción de tiempo en el proceso de compra del 23.60% y un incremento de la satisfacción del 51.75%.

V. DISCUSIÓN

Analizando los resultados en general podemos atribuir que la reducción del tiempo en la gestión de compra se debe a que pudimos extender esta función atreves del aplicativo móvil y es así que pudimos observar en los resultados un aumento de satisfacción con respecto al uso habitual de la aplicación que ya disponían. Llegando a obtener los siguientes resultados: La reducción de tiempo en un 23.60% y el incremento del nivel de satisfacción en un 51.75%, en relación al uso de la aplicación móvil. Con respecto a estos resultados, se pudo demostrar que el manejo del aplicativo móvil mejoró el proceso de compra de las pymes, incrementando así la satisfacción de los usuarios.

Los resultados que se obtuvieron en el análisis de las pruebas de entrada y salida en el indicador tiempo nos arrojaron una media de 3.30 y 2.67 respectivamente, observándose en una reducción del 23.60 % con respecto al tiempo que toma realizar el proceso de compra. Con relación a ello, Castañeda et al. (2016) determinó que el principal problema en una gestión de compra, se encuentra en la aprobación de requerimientos y con la propuesta de su investigación logró eliminar tiempos de retraso y estos ayudaron a aumentar la productividad. Además de agregar la disponibilidad de los datos de las órdenes de compra, a los usuarios, también se mostraron datos del producto como stock, imágenes, precios y la posibilidad de agregar nuevos productos al sistema.

Asimismo, el análisis de los resultados en las pruebas que midieron la satisfacción hacia el uso del aplicativo móvil, entrada y salida, nos indicaron una media de 0.632 y 0.800, respectivamente, mostrando un incremento del 51,75% con respecto a la satisfacción. Con relación a ello, Chira et al. (2021) como parte de su investigación analizó la usabilidad y la fiabilidad de las tecnologías móviles, concluyendo que este tipo de tecnologías tienen un gran impacto en personas menores de 50 años.

Triviño (2020) demostró en su investigación que el 100% de sus clientes posee un dispositivo móvil y que el 92% cree que es importante realizar compras online desde aplicaciones móviles, concluyendo que la satisfacción de sus usuarios con el uso de estas tecnologías para la compra de productos es muy alta.

VI. CONCLUSIONES

Los análisis de los resultados de la investigación nos permitieron hacer las siguientes conclusiones:

1. Según los resultados de la investigación, pudimos obtener una reducción de tiempo del 23.60%. El aplicativo móvil permitió que el comerciante tenga la posibilidad de gestionar la orden de compra en tiempo real. Esto quiere decir que podía ingresar los productos al stock y/o modificarlos.

2. En relación al nivel de satisfacción, este fue del 51.75% después de uso del aplicativo móvil por parte de los comerciantes, esto debido que se visualiza en tiempo real el stock, precio de compra, precio de venta e imágenes del producto. También permite crear productos nuevos para poder generar orden de compra de los mismos.

3. Analizando los resultados en general podemos afirmar que el uso de la aplicación móvil permitió reducir el tiempo de gestión de orden de compra incrementando así la satisfacción del uso del mismo.

VII. RECOMENDACIONES

En este capítulo se recomienda algunas pautas para otras investigaciones:

1. Agregar dos capas de seguridad, primero, implementar el protocolo https al servicio de api-rest implementado, donde Domingo (2020) recomendó el uso de este protocolo con la finalidad de que los datos de identidad no puedan ser suplantados. Segundo, si bien el sistema api-rest implementado cuenta con token de identificación se recomienda usar un estándar más robusto el cual es JSON Web Tokens, donde Salas (2020) investigando sobre la seguridad de api-rest concluyó que, agregando este sistema de seguridad, brinda una mayor seguridad a las apis-rest.
2. Que la aplicación móvil permita tener un medio de comunicación entre el que programa las compras y el responsable de compra, con la finalidad de poder tener mayor comunicación entre los dos actores de la gestión de compra, donde Vicente et al. (2020) al implementar una aplicación móvil, simulando una red social, entre los proveedores y los clientes, obteniendo un gran porcentaje de aceptación, recomendó el agregar un chat de comunicación para la interacción de los mismos.
3. Que la aplicación móvil permita tener un sistema de notificación que informe al usuario (responsable de compra) que tiene una orden de compra en el momento que le asignaron.

REFERENCIAS

ARIAS-GÓMEZ, Jesús; VILLASÍS-KEEVER, Miguel Ángel; NOVALES, María Guadalupe Miranda. El protocolo de investigación III: la población de estudio. Revista Alergia México, 2016, vol. 63, no 2, p. 201-206.

BAUDEAN, Marcos. Introducción a la investigación aplicada.

BARAY, Héctor Luis Ávila. *Introducción a la metodología de la investigación*. Juan Carlos Martínez Coll, 2006. Recuperado de https://www.eumed.net/libros-gratis/2006c/203/

BERNAL, CÉSAR A. Metodología de la investigación. Tercera edición PEARSON EDUCACIÓN, Colombia, 2010 ISBN: 978-958-699-128-5

CALVA NAVARRO, Mayra Lissette; GUERRERO YAGUAL, Maritza Raquel. Desarrollo de un prototipo de sistema Web y aplicación móvil asistente de ubicación e identificación de productos en un supermercado utilizando Android Studio y Herramienta. Net para optimizar y agilitar el proceso de compras de los clientes. 2020. Tesis de Licenciatura. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales. Recuperado de http://repositorio.ug.edu.ec/handle/redug/48878

CASTAÑEDA MORETO, Renato Arturo; DÍAZ RODRÍGUEZ, Edgard Javier. Propuesta de mejora en el proceso de gestión de compras, para incrementar la productividad en la empresa agroindustrial Casa Grande SA. 2016. Recuperado de https://hdl.handle.net/11537/10242

CHALLIOL, Cecilia; LLITERAS, Alejandra Beatriz; GORDILLO, Silvia Ethel. Diseño de aplicaciones móviles basadas en posicionamiento: un framework conceptual. En Congreso Argentino de Ciencias de la Computación. 2017. recuperado de https://digital.cic.gba.gob.ar/handle/11746/9042

CHIRA TORRES, Ernesto Oswaldo; CORDOVA PUGLIANINI, Cristian Nicolas. Aplicación móvil para el proceso de compra electrónica en el Minimarket Mass. 2019. Recuperado de https://hdl.handle.net/20.500.12867/3794.

CONDORI CABANA, David Godofredo; MONZÓN ESTRELLA, Antony Abel; PEÑA BARRIENTOS, Paul. Plan de negocio para una plataforma tecnológica de intermediación de compras en los mercados de abastos de Lima Metropolitana. 2021. Recuperado de https://hdl.handle.net/20.500.12640/2305

DEEMER, Pete, et al. Información básica de SCRUM. California: Scrum Training Institute, 2009. recuperado de http://libroslibres.uls.edu.sv/informatica/informacion_basica_scrum.pdf

DOMINGO BARRERO, Carlos. Desarrollo de una aplicación de ayuda al comercio local para dispositivos Android. 2020. Tesis Doctoral. Recuperado de https://riunet.upv.es/handle/10251/152374

ELEJABARRIETA, F.; IÑIGUEZ, L. Construcción de escalas de actitud, tipo Thurstone y Likert. La Sociología en sus escenarios, 2008, no 17.

ESCOBAR JAÉN, Ricardo. Desarrollo de soluciones de marketing móvil a través de un asistente de compras en centros comerciales. 2021. Tesis de Licenciatura. Recuperado de https://hdl.handle.net/10983/25365

ESPINOZA-GALICIA, Carlos, et al. Implementación de plataforma Web y aplicaciones móviles mediante buenas prácticas usando tecnología .NET. Revista de Tecnologías de la Información y Comunicaciones, 2017, vol. 1, no 1, p. 42-49. Extraido de http://sebasoft.ddns.net:86/repo/revista_web_movil_net.pdf#page=49

FERNANDES, Naquita; BARFKNECHT, Catherine. Keep customers coming back: Enhancing value and satisfaction in a mobile shopping application context. Cogent Business & Management, 2020, vol. 7, no 1, p. 1788874. Recuperado de https://doi.org/10.1080/23311975.2020.1788874

FERNÁNDEZ ALONSO, Isaac, et al. “REST API Plataforma colaborativa. 2020.

HERNÁNDEZ, Luis Miguel Alamilla, et al. Arquitectura REST para el desarrollo de aplicaciones web empresariales. Revista Electrónica sobre Tecnología,

Educación y Sociedad, 2021, vol. 8, no 15. Recuperado de https://www.ctes.org.mx/index.php/ctes/article/view/748/909

HERNÁNDEZ-SAMPIERI, Roberto; TORRES, Christian Paulina Mendoza. Metodología de la investigación. México^ eD. F DF: McGraw-Hill Interamericana, 2014.

HERNÁNDEZ-SAMPIERI, Roberto; FERNÁNDEZ-COLLADO, R.; BAPTISTA-LUCIO, Pilar. Selección de la muestra. 2017.

HERNANDO CALLEJA, Daniel. Protección de APIs REST. 2021. Recuperado de http://hdl.handle.net/10609/132467

LAZA, Carmen Arenal. Gestión de compras en el pequeño comercio. MF2106. Tutor Formación, 2016. extraido de https://books.google.com.ar/books?id=mgEKDQAAQBAJ&pg=PA124&source=gbs_selected_pages&cad=3#v=onepage&q&f=false

LOZADA, José. Investigación aplicada: Definición, propiedad intelectual e industria. CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica, 2014, vol. 3, no 1, p. 47-50.

MORONE, Guillermo. Métodos y técnicas de la investigación científica. Documento de trabajo. Valparaíso, Chile: Pontificia Universidad Católica de Valparaíso. Sistema de Biblioteca, 2013.

MURILLO, Javier. Métodos de investigación de enfoque experimental. Recuperado el, 2011, vol. 2.

ONECHA BLANCO, Jorge. Desarrollo de una aplicación móvil para compra-venta de instrumentos musicales. 2020. Recuperado de http://hdl.handle.net/10609/117566

ORLANDONI, Giampaolo. Escalas de medición en Estadística. Telos: Revista de Estudios Interdisciplinarios en Ciencias Sociales, 2010, vol. 12, no 2, p. 243-243.

PALACIO, Alberto Montoya. Conceptos modernos de administración de compras. Editorial Norma, 2002. extraido de https://books.google.es/books?hl=es&lr=&id=gJ9pNIMDbsoC&oi=fnd&pg=PR13&dq=montoya+2002&ots=1ixZupvLVq&sig=sCTy6evd-8qZSWdsau1vb78NLLg#v=onepage&q=montoya%202002&f=false

PÁRRAGA GANCHOZO, José Xavier; SOLÓRZANO VERA, Quinche Noemí. Aplicación móvil de venta y entrega de productos del Supermarket Supercito de la ciudad de Calceta. 2021. Tesis de Licenciatura. Calceta: ESPAM MFL. Recuperado de http://repositorio.espam.edu.ec/handle/42000/1370

PARRALES YÁNEZ, María Gabriela. Diseño de un manual de procedimientos para la gestión de compras en shoe store–GUAYAQUIL. 2017. Tesis de Licenciatura. Universidad de Guayaquil. Recuperado de http://repositorio.ug.edu.ec/handle/redug/17747

PÉREZ, Laura Silvia Vargas, et al. Tecnología móvil para evaluar la calidad de las herramientas de diseño rápido para generar sistemas de información. Pistas educativas, 2018, vol. 38, no 120. recuperado de http://www.itc.mx/ojs/index.php/pistas/article/view/630/559

QUIRAMA CAÑAVERAL, Viviana Marcela. AuthentiCity: desarrollo de una aplicación móvil para la dinamización de los comercios de proximidad.2019. Recuperado de http://hdl.handle.net/10609/95766

RAZALI, Nornadiah Mohd, et al. Comparaciones de potencia de las pruebas shapiro-wilk, kolmogorov-smirnov, lilliefors y anderson-darling. Journal of statistical modeling and analytics,2011, vol. 2, no 1, p. 21-33.

RIVERA JIMENEZ, Hector Enrique. Propuesta de aplicativo móvil de la gestión del conocimiento para mejorar la optimización del tiempo en una empresa de TI, Lima, 2020. 2020. Recuperado de http://repositorio.uwiener.edu.pe/handle/123456789/3928

ROMERO SANTIBÁÑEZ, Sebastián Adolfo. La orden de compra y sus requisitos, considerando los actuales mecanismos electrónicos en que se efectúan las transacciones mercantiles. 2013. Recuperado de http://repositorio.uchile.cl/handle/2250/114377

RUIZ CORDÓN, Adrián. Desarrollo de una app para mejorar la experiencia de compra al por menor. 2019. Tesis de Licenciatura. Universitat Politècnica de Catalunya. recuperado de http://hdl.handle.net/2117/131617

SALAS GONZÁLEZ, Eduardo Marcelo. Aplicando seguridad a una API REST recuperado de http://hdl.handle.net/10609/126696

SANGAMA OÑATE, Abel Fernando. Metodologías ágiles Scrum, XP, SLeSS, Scrumban, HME, Mobile-D y MASAN empleadas en la industria de dispositivos móviles: Un contraste en favor de la industria del desarrollo móvil. 2020. recuperado de http://hdl.handle.net/20.500.12840/3906

TOVAR CARDOZO, Daniel, et al. Propuesta de diseño de una aplicación móvil para la gestión y control de inventarios en la empresa Deluxe Business Group. 2021. recuperado de https://hdl.handle.net/10983/25365

TRIVIÑO RON, Gonzalo Darío. DISEÑO DE UNA APLICACIÓN MÓVIL DE VENTAS ONLINE DE PRODUCTOS FARMACÉUTICOS DE LA FARMACIA “EXIFARMA” DE LA CIUDAD DE BALZAR EN EL AÑO 2020. 2021. Tesis de Licenciatura. Instituto Superior Universitario Bolivariano de Tecnología. Recuperado de http://repositorio.itb.edu.ec/handle/123456789/2669

TURCIOS, RA Sánchez. Prueba de Wilcoxon-Mann-Whitney: mitos y realidades. Rev Mex Endocrinol Metab Nutr, 2015, vol. 2, p. 18-21.

VERA, Iván Darío González. Market Cart App. Aplicación móvil para la gestión de compra de víveres en línea. Tecnología Investigación y Academia, 2018, vol. 6, no 1, p. 3-17. Recuperado de https://revistas.udistrital.edu.co/index.php/tia/article/view/8660

VICENTE MACIÁN, Alejandro Javier, et al. *Aplicación móvil para facilitar la búsqueda de responsables de compras y proveedores de empresas*. 2020. Tesis Doctoral. Recuperado de http://hdl.handle.net/10251/150480

Anexo 1: Declaratoria de autenticidad del autor

DECLARATORIA DE AUTENTICIDAD DEL AUTOR

Yo Torres Nuñez, Wilder, alumno de la Facultad Ingeniería y Arquitectura y Escuela Profesional Ingeniería de Sistemas de la Universidad César Vallejo Lima Este, declaro bajo juramento que todos los datos e información que acompañan a la tesis titulado: “Aplicación móvil con servicio api-resp para mypes” son:

1. De mi autoría
2. El presente Trabajo de Investigación / Tesis no ha sido plagiado ni total, ni parcialmente.
3. El Trabajo de Investigación / Tesis no ha sido publicado ni presentado anteriormente.
4. Los resultados presentados en el presente Trabajo de Investigación /Tesis son reales, no han sido falseados, ni duplicados, ni copiados.

Lima, 05 de diciembre del 2021

……..

Torres Nuñez Wilder

DNI: 43621579

Anexo 2: Declaratoria de autenticidad del asesor

DECLARATORIA DE AUTENTICIDAD DEL ASESOR

Yo, Dr. Hilario Falcón, Francisco Manuel, docente de la Facultad Ingeniería y Arquitectura y Escuela Profesional Ingeniería de Sistemas de la Universidad César Vallejo Lima Este, revisor del trabajo de tesis titulado: “Aplicación móvil con servicio api-resp para mypes” del estudiante Torres Nuñez, Wilder, constato que la investigación tiene un índice de similitud de …….% verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y he concluido que cada una de las coincidencias detectadas no constituyen plagio. En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Lima, 05 de diciembre del 2021

……..

Dr. Hilario Falcón, Francisco Manuel

DNI: …….

Anexo 3: Matriz de consistencia

Matriz de Consistencia

PROBLEMA	OBJETIVO	HIPOTESIS	VARIABLE	DIMENSION	INDICADORES
General	**General**	**General**			
¿Cómo podemos mejorar el proceso de gestión de ordenes compra de las pymes?	Determinar si el manejo del aplicativo móvil brindando accesibilidad y disponibilidad puede mejorar el proceso de compra en las pymes	Se mejoró el proceso de compra de las mypes, reduciendo el tiempo del mismo, causando satisfacción en el manejo del aplicativo móvil	-	-	-
Específicos	**Específicos**	**Específicos**			
¿Cómo podemos mejora el proceso de orden de compra?	Determinar si la implementación de la aplicación móvil reduce el tiempo del proceso de orden de compra en las pymes.	El uso de la aplicación móvil mejoró el proceso de orden de compra de la pymes	Efecto del uso de la aplicación móvil para mejorar el proceso de orden de compras(Péres et al.,2018,p.15;Vera,2018,p.16;Sangama,2020,p.17;Párraga et al,2021,73; Salas,2020,p.56)	Tiempo	Reducción de Tiempo (Condori et al.,2021,p.55; Chira et al.,2021p.30)
¿ Se podrá causar un efecto de satisfacción en el uso de la aplicación móvil?	Determinar el efecto del uso de la aplicación móvil en el nivel de satisfacción del proceso de orden de compra	El uso de la aplicación móvil incrementó el nivel de satisfacción del proceso de orden de compra		Satisfacción	Incremento de Satisfacción (Calva et al.,2020,p.124; Condori et al., 2021,p.53;Chira et al.,2021,p.30;Triviño,2020,p.110)

Tabla 10: Matriz de consistencia.

Anexo 4: Matriz de operacionalización de variables

Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicadores	Instrumento	Escala de Medición
Efecto del uso del aplicativo móvil para mejorar el proceso de orden de compras(Pérez et al.,2018,p.15;Vera,2018,p.16;Sangama,2020,p.17;Párraga et al,2021,73; Salas,2020,p.56)	Actualmente el mundo se encuentra inmerso en la tecnología, donde la tecnología móvil ha facilitado las iteraciones, comunicaciones y transacciones diarias del ser humano. Estas tecnologías móviles les facilitan a los pequeños comercios a administrar sus procesos de manera eficaz y efectiva. Justificando el bajo costo y simple uso de los dispositivos móviles. (Vera,2018,p.5;Pérez et al.,2018,p.15)	La idea principal fue mejorar el proceso de orden de compra, reduciendo el tiempo y mejorando la satisfacción del proceso. Para lograr este objetivo, se fragmento el proceso de generación de orden de compra y se implementó un servicio rest, con el propósito de extender la funcionalidad en un dispositivo móvil que permita agilizar los trámites de orden de compra. Para esto se analizó los resultados atreves de encuestas.	Tiempo Los clientes valoran el tiempo y la seguridad que pueden ganar al realizar compras, mejorando el proceso de estos. (Condori et al. 2021,p.55;Chira et al.2021,p30)	Reducción de tiempo (Rivera, 2020, p,17)	Encuesta	Razón
			Satisfacción Los dispositivos móviles son una tecnología que nos permite realizar tareas diarias de manera fácil y eficientemente, brindando satisfacción al ser intuitivo y de mayor distribución en el globo. (Calva et al.,2020,p.124; Condori et al., 2021,p.53;Chira et al.,2021,p.30;Triviño,2020,p.110)	Incremento de satisfacción (Fernandes, 2020,p.17)	Encuesta	Likert

Tabla 11: Matriz de operacionalización de variables.

Anexo 5: Instrumento de recolección de datos

Evaluación de administración del tiempo y satisfacción de gestión de órdenes de compra

El siguiente cuestionario es confidencial y solo va ser utilizado para una tesis de investigación

*Obligatorio

1. ¿Tiene algún sistema de información? *

   Marca solo un óvalo.

   - Sí
   - No
   - Tal vez

2. ¿Este sistema tiene algún proceso de gestión de compra? *

   Marca solo un óvalo.

   - Sí
   - No
   - Tal vez

3. ¿Usa el proceso de gestión de compra que propone el sistema? *

   Marca solo un óvalo.

   - Si
   - No
   - Tal vez

4. ¿Cuánto tiempo le demora actualmente poder realizar una compra de sus productos? *

   Marca solo un óvalo.

   - 2 Horas
   - 3 Horas
   - 4 Horas
   - 5 Horas

5. ¿Qué tipo de negocio comercial tiene? *

Marca solo un óvalo.

- ◯ Venta de ropa
- ◯ Venta de calzado
- ◯ Venta de artículos tecnológicos
- ◯ Venta de abarrotes
- ◯ Otros

6. ¿Qué parte del proceso de compra le toma más tiempo?

Marca solo un óvalo.

- ◯ Tener el requerimiento de compra
- ◯ Cotizar los productos
- ◯ Seleccionar al proveedor
- ◯ Generar la orden de compra

7. ¿Estarías dispuesto a utilizar un dispositivo móvil para obtener información de sus productos? *

Marca solo un óvalo.

- ◯ Si
- ◯ No
- ◯ Tal vez

8. ¿Qué tan satisfecho se siente con el sistema que maneja actualmente para la gestión de compra?

Marca solo un óvalo.

- ◯ Poco satisfecho
- ◯ Regular satisfecho
- ◯ Muy satisfecho

Evaluación de administración del tiempo y satisfacción de gestión de órdenes de compra

El siguiente cuestionario es confidencial y solo va ser utilizado para una tesis de investigación

*Obligatorio

1. ¿Le pareció útil la aplicación móvil para generar una orden de compra? *

   Marca solo un óvalo.

   - ◯ Sí
   - ◯ No
   - ◯ Tal vez

2. ¿Cuánto tiempo le tomó gestionar la orden de compra? *

   Marca solo un óvalo.

   - ◯ 2 Horas
   - ◯ 3 Horas
   - ◯ 4 Horas
   - ◯ 5 Horas

3. ¿Qué le pareció el nuevo esquema de gestión de compra? *

   Marca solo un óvalo.

   - ◯ Bueno
   - ◯ Regular
   - ◯ Malo

4. ¿Sintió que tuvo la información necesaria para realizar la orden de compra? *

   Marca solo un óvalo.

   - ◯ Si
   - ◯ No
   - ◯ Tal vez

5. ¿Qué satisfecho quedó con la utilización de la aplicación? *

   Marca solo un óvalo.

   - ◯ Poco satisfecho
   - ◯ Regular satisfecho
   - ◯ Muy satisfecho

Anexo 6: Desarrollo de la metodología

A continuación, aplicaremos la metodología Scrum, la cual se eligió para esta investigación en el capítulo segundo, para el desarrollo del aplicativo móvil se usará Android studio, para el api_resp se usará php y para la base de datos mysql.

1. Equipo Scrum

Debido que esta investigación contiene a un solo autor, definimos los tres roles al autor.

Persona	Contacto	Rol
Wilder Torres N.	wildertn@gmail.com	Product Owner Scrum Master Team Member

Fuente: Elaboración Propia

2. Historias de Usuario

Como se estable en scrum definimos las historias de usuario que necesitará el aplicativo móvil, estas deben contener una descripción, una estimación (cabe aclarar que la estimación no está relacionada con el tiempo sino con la dificultad, y esta a su vez puede referirse a variables como incertidumbre o lo laborioso que es desarrollarlo), una prioridad y sus criterios de aceptación.

Historia de Usuario 1	
ID	H1
Nombre	Ingresar al aplicativo
Prioridad	1
Estimación	9
Descripción: Brindar al usuario un acceso con usuario y contraseña a la aplicación.	
Criterios de Aceptación: Se deben crear el esquema que permita gestionar los usuarios conectados desde el aplicativo móvil, como accesos o restricciones al mismo, también se debe crear un token de seguridad para el uso del api.	

Fuente: Elaboración propia

Historia de Usuario 2	
ID	H2
Nombre	Brindar acceso a los productos
Prioridad	3
Estimación	15
Descripción: Brindar al usuario acceso a la información de los productos, con una búsqueda inteligente que permita mostrar datos como el stock, precios, características del producto (marca, color y tallas)y por ultimo a las imágenes o fotografías que permitan identificarlo.	
Criterios de Aceptación: Dar acceso a la información de los productos con una búsqueda inteligente que permita ver las características del producto como son los precios, stock, marca, color, tallas y por ultimo permitir ver las imágenes o fotografías que posea el producto para la identificación del mismo.	

Fuente: Elaboración propia

Historia de Usuario 3	
ID	H3
Nombre	Ingresar nuevos productos
Prioridad	2
Estimación	23
Descripción: Brindar al usuario la opción de poder ingresar, modificar y listar nuevos productos al sistema, esto con la finalidad de poder realizar órdenes de compra sobre productos nuevos.	
Criterios de Aceptación: El usuario debe tener la posibilidad de ingresar nuevos productos junto con sus imágenes, también debe poder modificarlos antes de su aprobación o en el caso contrario poder eliminarlos para poder volver a cargarlos (esto debe permitir limpiar las imágenes cargados en el servidor bajo ese identificador de producto) y por ultimo debe permitir listar los productos nuevos que están a espera de su aprobación.	

Fuente: Elaboración propia

Historia de Usuario 4	
ID	H4
Nombre	Acceso a las Órdenes de Compra
Prioridad	1
Estimación	20
Descripción: Brindar al usuario acceso a la información de las órdenes de compra que se le asignaron, como también la opción de modificarlo, esto con la finalidad de existir un producto faltante poder cambiarlo.	
Criterios de Aceptación: El usuario debe tener la posibilidad de ver las órdenes de compra asignados y contar con las opciones de modificar, eliminar (anular) o enviar (es la confirmación de la compra). La confirmación de la orden de compra espera un proceso de aprobación, esta aprobación activa un proceso que ingresa al stock los productos para luego ser reflejados en los listados de productos.	

Fuente: Elaboración propia

3. Tareas

Listado de las Tareas según sus Historias			
Historia	**Tarea**	**Descripción**	**Estimación**
H1	T1	Esquema de ingreso del usuario en la base de datos.	2
	T2	Proceso en base de datos que permite validar al usuario y devolver el token de seguridad.	3
	T3	Api que permita validar con el proceso T2 y devolviendo el token de seguridad.	2
	T4	Login de acceso al sistema del aplicativo móvil.	2
H2	T5	Proceso para la búsqueda inteligente.	3
	T6	Proceso que devuelva datos del stock general según el producto.	1
	T7	Proceso que devuelva datos del stock según el producto y la marca.	1
	T8	Proceso que devuelva datos del stock según el producto, la marca y el color.	1
	T9	Proceso que devuelva datos del stock según el producto, la marca, el color y la talla.	1
	T10	Proceso que devuelva datos de la ubicación de la imagen según el producto.	2
	T11	Api que devuelva los datos según T5 al T10.	3
	T12	Listado de productos en el aplicativo móvil.	3
H3	T13	Esquema de Producto Nuevo	2
	T14	Función para verificar el token.	1

	T15	Api para ingresar un producto nuevo.	2
	T16	Api para ingresar imágenes asociados al producto nuevo.	2
	T17	Api para modificar un producto nuevo.	2
	T18	Api para eliminar un producto nuevo.	2
	T19	Api para listar un producto nuevo.	2
	T20	Modelo de producto nuevo en el aplicativo	1
	T21	Ingresar producto nuevo desde el aplicativo móvil.	3
	T22	Modificar o eliminar producto nuevo desde el aplicativo móvil.	3
	T23	Proceso de aprobación y asociación del producto nuevo con el listado de productos.	3
H4	T24	Esquema de orden de compra.	1
	T25	Función de verificación del token.	1
	T26	Api para listar las órdenes de compra asignadas	2
	T27	Api para modificar las cantidades de los productos según su marca, color y talla.	3
	T28	Api para anular las órdenes de compra.	1
	T29	Api para confirmar la compra de los productos.	1
	T30	Modelo de orden de compra y productos en el aplicativo móvil.	3
	T31	Listar Orden de Compra en el aplicativo móvil.	2
	T32	Modificar Orden de Compra en el aplicativo móvil	3
	T33	Proceso de aprobación de la orden de compra para el ingreso de productos al stock.	3

Fuente: Elaboración propia

4. Pila de Producto (Product Backlog)

Tarea	Descripción	Estimación	Prioridad
T1	Esquema de ingreso del usuario en la base de datos.	2	1
T2	Proceso en base de datos que permite validar al usuario y devolver el token de seguridad.	3	1
T3	Api que permita validar con el proceso T2 y devolviendo el token de seguridad.	2	1
T4	Login de acceso al sistema del aplicativo móvil.	2	1
T5	Proceso para la búsqueda inteligente.	3	2
T6	Proceso que devuelva datos del stock general según el producto.	1	3

T7	Proceso que devuelva datos del stock según el producto y la marca.	1	3
T8	Proceso que devuelva datos del stock según el producto, la marca y el color.	1	3
T9	Proceso que devuelva datos del stock según el producto, la marca, el color y la talla.	1	3
T10	Proceso que devuelva datos de la ubicación de la imagen según el producto.	2	3
T11	Api que devuelva los datos según T5 al T10.	3	2
T12	Listado de productos en el aplicativo móvil.	3	2
T13	Esquema de Producto Nuevo	2	2
T14	Función para verificar el token.	1	2
T15	Api para ingresar un producto nuevo.	2	2
T16	Api para ingresar imágenes asociados al producto nuevo.	2	2
T17	Api para modificar un producto nuevo.	2	3
T18	Api para eliminar un producto nuevo.	2	3
T19	Api para listar un producto nuevo.	2	3
T20	Modelo de producto nuevo en el aplicativo	1	2
T21	Ingresar producto nuevo desde el aplicativo móvil.	3	2
T22	Modificar o eliminar producto nuevo desde el aplicativo móvil.	3	2
T23	Proceso de aprobación y asociación del producto nuevo con el listado de productos.	3	2
T24	Esquema de orden de compra.	1	1
T25	Función de verificación del token.	1	1
T26	Api para listar las órdenes de compra asignadas	2	1
T27	Api para modificar las cantidades de los productos según su marca, color y talla.	3	1
T28	Api para anular las órdenes de compra.	1	1
T29	Api para confirmar la compra de los productos.	1	1
T30	Modelo de orden de compra y productos en el aplicativo móvil.	3	1
T31	Listar Orden de Compra en el aplicativo móvil.	2	1
T32	Modificar Orden de Compra en el aplicativo móvil	3	1
T33	Proceso de aprobación de la orden de compra para el ingreso de productos al stock.	3	1

Fuente: Elaboración propia

5. Listado del Sprint (Sprint Backlog)

En cada sprint se tiene que realizar un entregable funcional por lo que se tratará de introducir las tareas a tal que puedan realizar un entregable, cabe mencionar que cada sprint consta de 2 semanas respetando las ceremonias de review y planning, obviando las ceremonias de dayli y refinamiento por contar con un solo actor.

Sprint	Periodo	Historia	Tarea	Puntos de Historia
1	23/08/2021 – 03/09/2021	H1	T1	2
			T2	3
			T3	2
			T4	2
		H2	T5	3
Total de Puntos de Historia				12

Fuente: Elaboración Propia

Sprint	Periodo	Historia	Tarea	Puntos de Historia
2	06/09/2021 – 17/09/2021	H2	T6	1
			T7	1
			T8	1
			T9	1
			T10	2
			T11	3
			T12	3
Total de Puntos de Historia				12

Fuente: Elaboración Propia

Sprint	Periodo	Historia	Tarea	Puntos de Historia
3	20/09/2021 – 01/10/2021	H3	T13	2
			T14	1
			T15	2
			T16	2

			T17	2
			T18	2
			Total de Puntos de Historia	11

Fuente: Elaboración Propia

Sprint	**Periodo**	**Historia**	**Tarea**	**Puntos de Historia**
4	04/10/2021 – 15/10/2021	H3	T19	2
			T20	1
			T21	3
			T22	3
			T23	3
			Total de Puntos de Historia	12

Fuente: Elaboración Propia

Sprint	**Periodo**	**Historia**	**Tarea**	**Puntos de Historia**
5	18/10/2021 – 29/10/2021	H4	T24	1
			T25	1
			T26	2
			T27	3
			T28	1
			T29	1
			T30	3
			Total de Puntos de Historia	12

Fuente: Elaboración Propia

Sprint	**Periodo**	**Historia**	**Tarea**	**Puntos de Historia**
6	01/11/2021 – 12/11/2021	H4	T31	2
			T32	3
			T33	3
			Total de Puntos de Historia	8

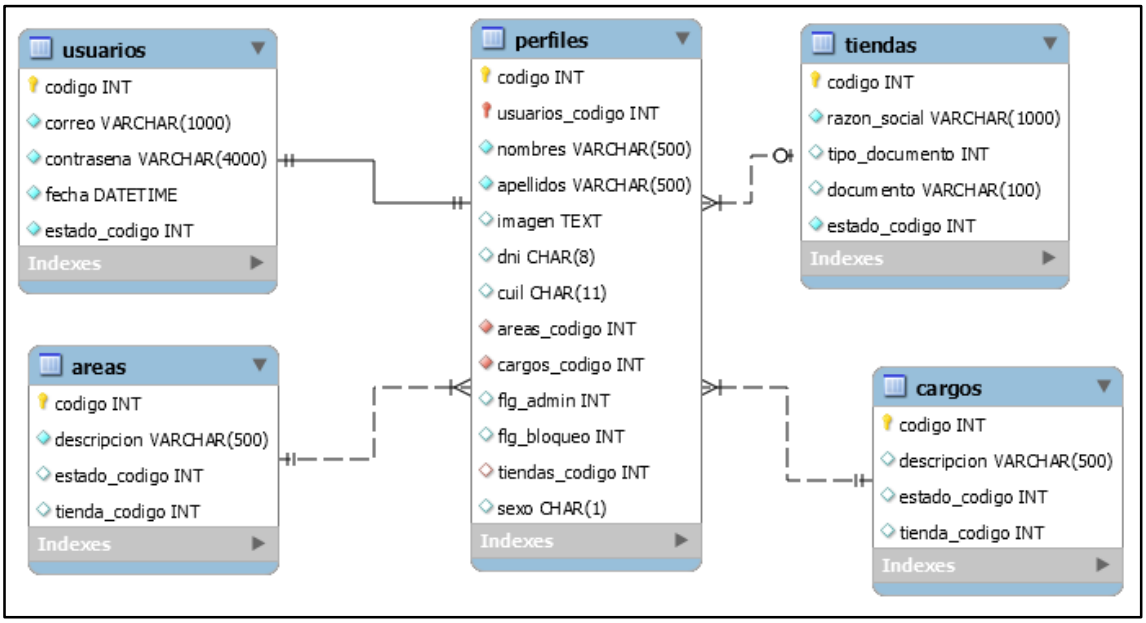
Fuente: Elaboración Propia

6. Desarrollo de Sprint 1

6.1. Tareas T1

Crear el esquema del usuario en la base de datos.

6.1.1. Diseño del esquema de usuario para el acceso



Fuente: Elaboración propia

Figura 3: Esquema de usuario en la base de datos.

6.2. Tarea T2

Crear el proceso de base de datos que permita validar el usuario y contraseña que se envía desde el, también tiene que devolver el token de seguridad.

6.2.1. Diseño del Proceso

Crear el proceso con el nombre de sp_iniciar_sesion_android que tiene que recibir como parámetro el email del usuario con la contraseña encriptada, devolviendo un cursor que tenga los datos correctos “1” o incorrecto “-1”,”-2”,”-3”,”-4”. El -1 hace referencia que el usuario y contraseña son incorrectos, el -2 que el usuario se encuentra bloqueado por varios intentos, el -3 es cuando intentó varias veces con error y esto activa el bloqueo automático y el -4 que nos dice que es el bloqueo realizado por el administrador del sistema.

6.2.2. Código Fuente

```
CREATE DEFINER=`vadmin`@`%` PROCEDURE `sp_iniciar_sesion_android`(
    in p_email varchar(4000),
    in p_pass varchar(4000))
BEGIN
    declare bloqueo,idusuario int;
    declare bloqueo_user,tienda int;
    if (select exists (select   1
                       from     perfiles a,usuarios b
                       where    a.usuarios_codigo = b.codigo
                                and b.correo = p_email)) then
        select  estado_codigo
        into    bloqueo_user
        from    usuarios
        where   correo = p_email;
        if (bloqueo_user = 4) then
            select -2 resp;
        elseif (bloqueo_user = 2) then
            select -4 resp;
        elseif(select exists (select    1
                              from      perfiles a,usuarios b
                              where     a.usuarios_codigo = b.codigo
                                        and b.correo = p_email
                                        and contrasena = p_pass)) then
            select  ifnull(b.flg_bloqueo,0),b.tiendas_codigo,a.codigo
            into    bloqueo,tienda,idusuario
            from    usuarios a left join
                    perfiles b on a.codigo = b.usuarios_codigo
            where   a.correo = p_email;
            if (bloqueo = 1) then
                select -3 resp;
            else
                select idusuario resp,tienda;
            end if;
        else
            select -1 resp;
        end if;
    else
        select -1 resp;
    end if;
END
```

Fuente: Elaboración propia

Figura 4: Store Procedure de validación de usuario.

6.3. Tarea T3

Crear el api de validación

6.3.1. Diseño

Crear el api que permita validar el proceso de validación de usuario y contraseña de base de datos (T2). En el caso de que la validación sea correcta, devolver el id del usuario y el token de seguridad.

6.3.2. Código Fuente

auth.php

```php
<?php
require_once 'clases/auth.class.php';
require_once 'clases/respuestas.class.php';

$_auth = new auth;
$_respuestas = new respuestas;

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){
    //recibimos los datos
    $postBody = file_get_contents("php://input");

    //enviamos los datos al servidor
    $datosArray = $_auth->login($postBody);

    //devolvemos una respuesta
    if(isset($datosArray['result']['error_id'])){
        $code = $datosArray['result']['error_id'];
        http_response_code($code);

    }else {
        http_response_code(200);

    }
    echo json_encode($datosArray);
}else {
    echo json_encode($_respuestas->error_405());
}

?>
```

Fuente: Elaboración Propia

Figura 5: Código fuente del api de validación de usuario.

6.4. Tarea T4

Crear el login de acceso al aplicativo móvil

6.4.1. Diseño del Login de Acceso

Crear el login de acceso que permita usar el api de validación y en el caso de ser correcto guardar en una variable global el token de acceso.

Fuente: Elaboración Propia

Figura 6: Login de acceso del aplicativo móvil.

6.4.2. Código Fuente del login(Clases)

- Clases de configuración

```java
package com.sebasoft.tienda11.db;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import androidx.annotation.Nullable;

public class Usuario extends dbHelper {
    Context context;
    String  sQuery,token;
    private int idtienda;
    private int iduser;
    public Usuario(@Nullable Context context) {
        super(context);
        this.context = context;
    }

    public void _iniciarDatosUsuario(){
        Cursor cur = null;
        try{
            dbHelper DBHelper = new dbHelper(context);
            SQLiteDatabase db = DBHelper.getWritableDatabase();
            // sQuery = "select idtienda,idusuario,token from " + TABLE_USUARIO ;
            sQuery = "select b.idtienda,b.idusuario,b.token\n" +
                    "from "+TABLE_SESSION+" a,"+TABLE_USUARIO+" b\n" +
                    "where a.idtienda = b.idtienda and a.idusuario = b.idusuario";

            cur = db.rawQuery(sQuery, null);
            if (cur.moveToFirst()) {
                do{
                    this.idtienda = cur.getInt(0);
                    this.iduser = cur.getInt(1);
                    this.token = cur.getString(2);
                }while (cur.moveToNext());
            }
        }catch (Exception ex){
            ex.getMessage();
        }
    }
    public int getTienda(){
        return this.idtienda;
    }
    public int getUsuario(){
        return this.iduser;
    }
    public String getToken(){ return this.token; }

```

```
49      public void setUsuario(int idtienda,int iduser, String token){
50
51          if (existeSesion()){
52              updateSesion(idtienda,iduser);
53          }else{
54              insertSession(idtienda,iduser);
55          }
56          if(existeConfig(idtienda,iduser)){
57              updateUsuario(idtienda,iduser,token);
58          }else{
59              insertUsuario(idtienda,iduser,token);
60          }
61      }
62
63      private void updateSesion(int idtienda, int iduser){
64          try{
65
66              dbHelper DBHelper = new dbHelper(context);
67              SQLiteDatabase db = DBHelper.getWritableDatabase();
68              ContentValues cv = new ContentValues();
69              cv.put("idusuario",iduser);
70              cv.put("idtienda",idtienda);
71              db.update(TABLE_SESSION, cv, null,null);
72          }catch (Exception ex){
73              ex.toString();
74          }
75      }
76
77      private long insertSession(int idtienda, int iduser){
78          long id =0;
79          try {
80              dbHelper DBHelper = new dbHelper(context);
81              SQLiteDatabase db = DBHelper.getWritableDatabase();
82              ContentValues values = new ContentValues();
83              values.put("idtienda", idtienda);
84              values.put("idusuario", iduser);
85              id =  db.insert(TABLE_SESSION, null, values);
86          }catch (Exception ex){
87              ex.toString();
88          }
89          return id;
90      }
91
92      private long insertUsuario(int idtienda, int iduser, String token){
93          long id =0;
94          try {
95              dbHelper DBHelper = new dbHelper(context);
96              SQLiteDatabase db = DBHelper.getWritableDatabase();
97              ContentValues values = new ContentValues();
98              values.put("idtienda", idtienda);
99              values.put("idusuario", iduser);
100             values.put("token",token);
101             id =  db.insert(TABLE_USUARIO, null, values);
102         }catch (Exception ex){
103             ex.toString();
104         }
105         return id;
106     }
```

```
107
108     private void updateUsuario(int idtienda, int iduser, String token){
109         try{
110             dbHelper DBHelper = new dbHelper(context);
111             SQLiteDatabase db = DBHelper.getWritableDatabase();
112             ContentValues cv = new ContentValues();
113             cv.put("token",token);
114             db.update(TABLE_USUARIO, cv, "idusuario = ? and idtienda = ?",
115                     new String[]{Integer.toString(iduser),Integer.toString(idtienda)});
116         }catch (Exception ex){
117             ex.toString();
118         }
119     }
120
121
122     public boolean existeConfig(int idtienda,int iduser){
123         Cursor cur = null;
124         int total = 0;
125         try {
126             dbHelper DBHelper = new dbHelper(context);
127             SQLiteDatabase db = DBHelper.getWritableDatabase();
128             sQuery = "select count(*) from " + TABLE_USUARIO +
129                     " where idtienda = ? and idusuario = ?";
130             cur = db.rawQuery(sQuery, new String[]{Integer.toString(idtienda),Integer.toString(iduser)});
131             if (cur.moveToFirst()) {
132                 do{
133                     total = cur.getInt(0);
134                 }while (cur.moveToNext());
135             }
136             cur.close();
137
138             if (total == 0) {
139                 return false;
140             }
141         }catch (Exception ex){
142             ex.toString();
143         }
144         return true;
145     }
146
147     public boolean existeSesion(){
148         Cursor cur = null;
149         int total = 0;
150         try {
151             dbHelper DBHelper = new dbHelper(context);
152             SQLiteDatabase db = DBHelper.getWritableDatabase();
153             sQuery = "select count(*) from " + TABLE_SESSION;
154             cur = db.rawQuery(sQuery, null);
155             if (cur.moveToFirst()) {
156                 do{
157                     total = cur.getInt(0);
158                 }while (cur.moveToNext());
159             }
160             cur.close();
161             if (total == 0) {
162                 return false;
163             }
164         }catch (Exception ex){
165             ex.toString();
166         }
167         return true;
168     }
169 }
```

Fuente: Elaboración propia

Figura 7: Código fuente modelo usuario del aplicativo móvil.

```java
package com.sebasoft.tienda11.db;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import androidx.annotation.Nullable;

public class dbconfiguracion extends dbHelper{
    Context context;
    public dbconfiguracion(@Nullable Context context) {
        super(context);
        this.context = context;
    }

    public void setDBconfig(String servidor,String version){
        if (existeConfig())
            updateDBconfig(servidor,version);
        else
            setservidor(servidor,version);
    }

    public long setservidor(String servidor,String version){
        long id = 0;
        try {

            dbHelper DBHelper = new dbHelper(context);
            SQLiteDatabase db = DBHelper.getWritableDatabase();

            ContentValues values = new ContentValues();
            values.put("servidor", servidor);
            values.put("version", version);

            id =  db.insert(TABLE_CONF, null, values);
        }catch (Exception ex){
            ex.toString();
        }
        return  id;
    }
    public String getServidor(){
        String  Query,Servidor;
        Cursor cur = null;
        Servidor = "";
        try {
            dbHelper DBHelper = new dbHelper(context);
            SQLiteDatabase db = DBHelper.getWritableDatabase();
            Query = "select servidor from " + TABLE_CONF +
                    " where id = 1";
            cur = db.rawQuery(Query, null);
            if (cur.moveToFirst()) {
                do {
                    Servidor = cur.getString(0);
                }while (cur.moveToNext());
            }
        }catch (Exception ex){
            ex.toString();
        }
        return Servidor;
    }
    public void updateDBconfig(String servidor,String version){
        try{
            dbHelper DBHelper = new dbHelper(context);
            SQLiteDatabase db = DBHelper.getWritableDatabase();
            ContentValues cv = new ContentValues();
            cv.put("servidor", servidor);
```

```
            cv.put("version", version);
            db.update(TABLE_CONF, cv, "id = ?", new String[]{"1"});
        }catch (Exception ex){
            ex.toString();
        }
    }

    public boolean existeConfig(){
        Cursor cur = null;
        int total = 0;
        try {
            dbHelper DBHelper = new dbHelper(context);
            SQLiteDatabase db = DBHelper.getWritableDatabase();
            String Query = "select count(*) from " + TABLE_CONF +
                    " where id = 1";
            cur = db.rawQuery(Query, null);
            if (cur.moveToFirst()) {
                do{
                    total = cur.getInt(0);

                }while (cur.moveToNext());
            }
            cur.close();
            if (total == 0) {

                return false;
            }
        }catch (Exception ex){
            ex.toString();
        }

        return true;
    }
}
```

Fuente: Elaboración propia

Figura 8: Código fuente configuración base de datos.

```
package com.sebasoft.tienda11.db;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class dbHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 7;
    private static final String DATABASE_NOMBRE = "comercio.db";
    public  static final String TABLE_CONF = "config";
    public  static final String TABLE_USUARIO = "usuario";
    public  static final String TABLE_SESSION = "sesion";
    public dbHelper(@Nullable Context context) {
        super(context, DATABASE_NOMBRE, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String Query;
        try {
```

```
24              Query = "CREATE TABLE " + TABLE_CONF + " (" +
25                      "id INTEGER PRIMARY KEY AUTOINCREMENT," +
26                      "servidor TEXT not null," +
27                      "version TEXT not null)";
28
29              sqLiteDatabase.execSQL(Query);
30              Query = "CREATE TABLE " + TABLE_USUARIO + " (" +
31                      "id INTEGER PRIMARY KEY AUTOINCREMENT," +
32                      "idtienda INTEGER not null," +
33                      "idusuario INTEGER not null," +
34                      "token TEXT not null)";
35              sqLiteDatabase.execSQL(Query);
36              Query = "CREATE TABLE " + TABLE_SESSION + " (" +
37                      "id INTEGER PRIMARY KEY AUTOINCREMENT," +
38                      "idtienda INTEGER not null," +
39                      "idusuario INTEGER not null)";
40              sqLiteDatabase.execSQL(Query);
41
42          }catch (Exception ex){
43              ex.getMessage();
44          }
45      }
46
47      @Override
48      public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
49
50          sqLiteDatabase.execSQL("drop table if exists "+TABLE_CONF);
51          sqLiteDatabase.execSQL("drop table if exists "+TABLE_USUARIO);
52          sqLiteDatabase.execSQL("drop table if exists "+TABLE_SESSION);
53          onCreate(sqLiteDatabase);
54      }
55 }
```

Fuente: Elaboración Propia

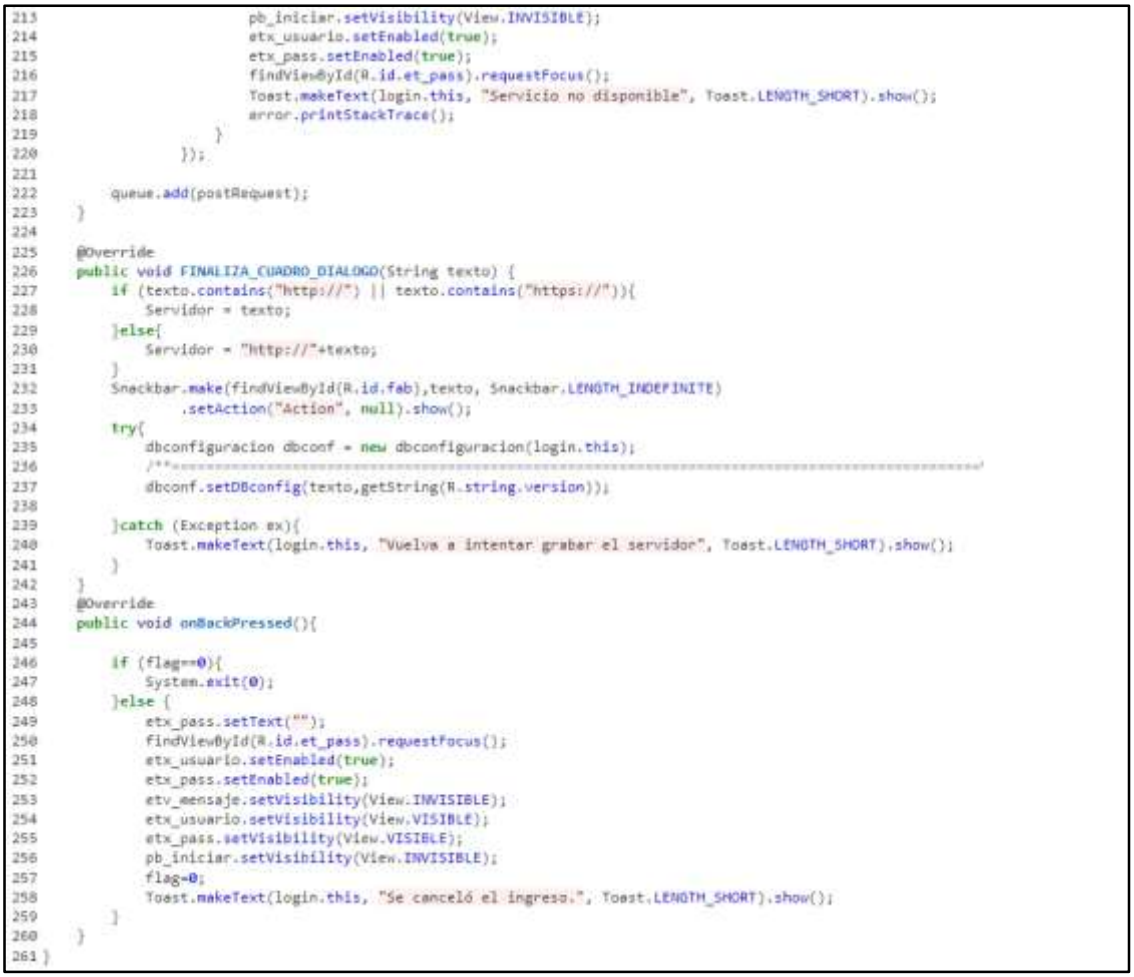
Figura 9: Código fuente de configuración inicial de base de datos.

- Clase login

```
1 package com.sebasoft.tienda11.login;
2
3 import androidx.appcompat.app.AlertDialog;
4 import androidx.appcompat.app.AppCompatActivity;
5
6 import android.content.DialogInterface;
7 import android.content.Intent;
8 import android.os.Bundle;
9 import android.os.Vibrator;
10 import android.view.KeyEvent;
11 import android.view.View;
12 import android.widget.EditText;
13 import android.widget.ProgressBar;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 import com.android.volley.Request;
18 import com.android.volley.RequestQueue;
19 import com.android.volley.Response;
20 import com.android.volley.VolleyError;
21 import com.android.volley.toolbox.JsonArrayRequest;
22 import com.android.volley.toolbox.JsonObjectRequest;
23 import com.android.volley.toolbox.StringRequest;
24 import com.android.volley.toolbox.Volley;
25 import com.google.android.material.floatingactionbutton.FloatingActionButton;
26 import com.google.android.material.snackbar.Snackbar;
27 import com.sebasoft.tienda11.MainActivity;
```

```java
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.db.Usuario;
import com.sebasoft.tienda11.db.dbconfiguracion;
import com.sebasoft.tienda11.ui.fragment.asignar_servidor;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class login extends AppCompatActivity implements asignar_servidor.nuevocursorlistener,EditText.OnEditorActionListener{
    String Servidor;
    EditText etx_usuario;
    EditText etx_pass;
    TextView etv_mensaje;
    ProgressBar pb_iniciar;
    Intager flag= 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        etx_usuario = (EditText)  findViewById(R.id.et_usuario);
        etx_pass = (EditText) findViewById(R.id.et_pass);
        etv_mensaje = (TextView) findViewById(R.id.tv_mensaje);
        pb_iniciar = (ProgressBar) findViewById(R.id.pb_iniciar);
        etx_pass.setOnEditorActionListener(this);

        //Regresamos el nombre del servidor
        dbconfiguracion dbconf = new dbconfiguracion(login.this);

        Servidor = null;
        if (dbconf.existeConfig()){
            Servidor = dbconf.getServidor();
        }


        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {

            // Aca vamos agregar el codigo
            @Override
            public void onClick(final View view) {
                AlertDialog.Builder dialogo = new AlertDialog.Builder(view.getContext());
                dialogo.setTitle("SERVIDOR");
                dialogo.setMessage("¿Ver o Modificar Servidor?");
                dialogo.setCancelable(false);

                dialogo.setPositiveButton("VER",new DialogInterface.OnClickListener(){
                    public void onClick(DialogInterface dialogo, int id){
                        String Resp;
                         if (Servidor==null){
                             Resp = "No hay un servidor asignado";
                         }else{
                             Resp = Servidor;
                         }
                        Snackbar.make(view, Resp, Snackbar.LENGTH_INDEFINITE)
                                .setAction("Action", null).show();
                    }

                });
                dialogo.setNegativeButton("MODIFICAR",new DialogInterface.OnClickListener(){
                    public void onClick(DialogInterface dialogo, int id){
                        asignar_servidor aserv = new asignar_servidor(Servidor);
                        aserv.show(getFragmentManager(),"Personal");

                        android.app.Fragment frag = getFragmentManager().findFragmentByTag("personal");
                        if (frag!=null){
                            getFragmentManager().beginTransaction().remove(frag).commit();
                        }return;
                    }

                });
                dialogo.show();
            }
        });
    }
    @Override
    public boolean onEditorAction(TextView textView, int i, KeyEvent keyEvent) {
        if (etx_usuario.getText().toString().length()>0){
            if (etx_pass.getText().toString().length()>0){
                String Resp;
                if (Servidor==null){
                    Toast.makeText(login.this,"No hay un servidor asignado.", Toast.LENGTH_LONG).show();
                }else{
                    try {
                        Conectar();
                    } catch (JSONException e) {
                        e.printStackTrace();
                        Toast.makeText(login.this,"El error en el servicio.", Toast.LENGTH_LONG).show();
```

```
                }
            }
        }else {
            Toast.makeText(login.this,"Ingresa una contraseña.", Toast.LENGTH_LONG).show();
        }
    }else{
        Toast.makeText(login.this,"Ingresa un usuario.", Toast.LENGTH_LONG).show();
    }
    return true;

}

public void Conectar() throws JSONException {
    String user,password,servicio;
    Map<String, String> data = new HashMap<>();

    user = etx_usuario.getText().toString();
    password = etx_pass.getText().toString();

    //Creamos el json que vamos a enviar
    data.put("user", user);
    data.put("password", password);
    JSONObject jsonData = new JSONObject(data);

    //Accionamos los controles
    etv_mensaje.setVisibility(View.VISIBLE);
    pb_iniciar.setVisibility(View.VISIBLE);
    etx_usuario.setEnabled(false);
    etx_pass.setEnabled(false);
    etv_mensaje.setText("Conectando...");
    flag = 1;
    Vibrator vibrator = (Vibrator) getSystemService(login.this.VIBRATOR_SERVICE);

    //consumimos el api
    servicio = Servidor + "/auth";


    RequestQueue queue = Volley.newRequestQueue(this);
    JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.POST, servicio,
            jsonData,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    int idtienda,idusuario;
                    String  token;
                    JSONObject Obj,Result;
                    Obj = response;
                    String  status;

                    try {
                        status = Obj.getString("status");
                        Result = Obj.getJSONObject("result");
                        if (status.equals("ok")){
                            // Post conexion
                            idtienda = Integer.parseInt(Result.getString("idtienda"));
                            idusuario = Integer.parseInt(Result.getString("idusuario"));
                            token = Result.getString("token");
                            Usuario usuario = new Usuario(login.this);
                            usuario.setUsuario(idtienda,idusuario,token);

                            startActivity(new Intent(login.this, MainActivity.class));
                            etv_mensaje.setText("");
                            etx_usuario.setText("");
                            etx_pass.setText("");
                            etv_mensaje.setVisibility(View.INVISIBLE);
                            etx_usuario.setEnabled(true);
                            etx_pass.setEnabled(true);
                            pb_iniciar.setVisibility(View.INVISIBLE);
                        }else{
                                vibrator.vibrate(400);
                                etv_mensaje.setText("Usuario o Contraseña Incorrecto");
                                pb_iniciar.setVisibility(View.INVISIBLE);
                                etx_usuario.setEnabled(true);
                                etx_pass.setEnabled(true);
                                findViewById(R.id.et_pass).requestFocus();
                        }
                    } catch (JSONException e) {
                        //El servio no trajo nada
                        e.printStackTrace();
                        vibrator.vibrate(400);
                        etv_mensaje.setText("Error en el Servicio API");
                        pb_iniciar.setVisibility(View.INVISIBLE);
                        etx_usuario.setEnabled(true);
                        etx_pass.setEnabled(true);
                        findViewById(R.id.et_pass).requestFocus();
                    }
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    vibrator.vibrate(400);
                    etv_mensaje.setText("Servicio no disponible");
```



```
                    pb_iniciar.setVisibility(View.INVISIBLE);
                    etx_usuario.setEnabled(true);
                    etx_pass.setEnabled(true);
                    findViewById(R.id.et_pass).requestFocus();
                    Toast.makeText(login.this, "Servicio no disponible", Toast.LENGTH_SHORT).show();
                    error.printStackTrace();
                }
            });

        queue.add(postRequest);
    }

    @Override
    public void FINALIZA_CUADRO_DIALOGO(String texto) {
        if (texto.contains("http://") || texto.contains("https://")){
            Servidor = texto;
        }else{
            Servidor = "http://"+texto;
        }
        Snackbar.make(findViewById(R.id.fab),texto, Snackbar.LENGTH_INDEFINITE)
                .setAction("Action", null).show();
        try{
            dbconfiguracion dbconf = new dbconfiguracion(login.this);
            /**=============================================================================================*/
            dbconf.setDBconfig(texto,getString(R.string.version));

        }catch (Exception ex){
            Toast.makeText(login.this, "Vuelva a intentar grabar el servidor", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onBackPressed(){

        if (flag==0){
            System.exit(0);
        }else {
            etx_pass.setText("");
            findViewById(R.id.et_pass).requestFocus();
            etx_usuario.setEnabled(true);
            etx_pass.setEnabled(true);
            etv_mensaje.setVisibility(View.INVISIBLE);
            etx_usuario.setVisibility(View.VISIBLE);
            etx_pass.setVisibility(View.VISIBLE);
            pb_iniciar.setVisibility(View.INVISIBLE);
            flag=0;
            Toast.makeText(login.this, "Se canceló el ingreso.", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Fuente: Elaboración propia

Figura 10: Código fuente de la clase login.

6.5. Tarea T5

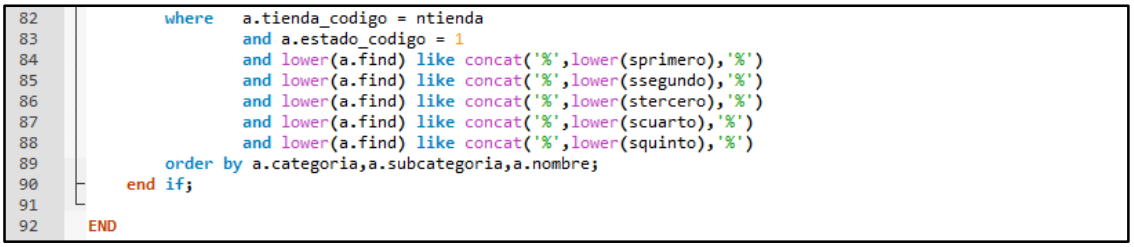
Crear proceso en la base de datos que permita realizar una búsqueda inteligente.

6.5.1. Diseño

Se necesita crear un procedimiento almacenado que reciba como parámetro una búsqueda, esta puede constar de una sola palabra o varias palabras separados por espacio, y este tiene que retornar un listado de productos.

6.5.2. Código fuente del procedimiento almacenado

```
CREATE DEFINER=`vadmin`@`%` PROCEDURE `sp_buscar_producto`(
    in cadena varchar(4000),
    in ntienda int)
BEGIN
    declare sprimero varchar(1000);
    declare ssegundo varchar(1000);
    declare stercero varchar(1000);
    declare scuarto varchar(1000);
    declare squinto varchar(1000);

    set sprimero = substring_index(cadena, ' ', 1);
    set ssegundo = substring_index(
                   trim(SUBSTR( cadena, length(sprimero) +1)) , ' ', 1);
    set stercero = substring_index(
                   trim(SUBSTR( cadena, length(sprimero)+
                   length(ssegundo)+2)) , ' ', 1);
    set scuarto = substring_index(
                   trim(SUBSTR( cadena, length(sprimero)+
                   length(ssegundo)+length(stercero)+3)) , ' ', 1);
    set squinto = substring_index(
                   trim(SUBSTR( cadena, length(sprimero)+length(ssegundo)+
                   length(stercero)+length(scuarto)+4)) , ' ', 1);

    if ssegundo = '' then
        if f_isnumber(sprimero) then
            select  a.codigo,a.categorias_codigo,a.subcategoria_codigo,a.categoria,a.subcategoria,a.nombre
                ,a.observacion,a.unidad_medida,a.estado,a.tienda_codigo,a.find,
                concat('/Imagenes/Productos/',cast(a.tienda_codigo as char),
                '/',cast(b.categorias_codigo as char),'/',
                cast(b.subcategoria_codigo as char),'/',b.direccion) img_url
                ,f_string_stock(a.categorias_codigo,a.subcategoria_codigo,a.codigo) stock_general
            from    vproductos a left join
                    imagenes b on b.categorias_codigo = a.categorias_codigo
                                and b.subcategoria_codigo = a.subcategoria_codigo
                                and b.productos_codigo = a.codigo
                                and b.codigo = (select min(x.codigo) from imagenes x
                                                where x.subcategoria_codigo = b.subcategoria_codigo
                                                and x.categorias_codigo = b.categorias_codigo
                                                and x.productos_codigo = b.productos_codigo)
            where   a.tienda_codigo = ntienda
                    and a.estado_codigo = 1
                    and concat(cast(a.categorias_codigo as char),cast(a.subcategoria_codigo as char),
                    cast(a.codigo as char)) = cadena
            order by a.categoria,a.subcategoria,a.nombre;
        else
            select  a.codigo,a.categorias_codigo,a.subcategoria_codigo,a.categoria,a.subcategoria,a.nombre
                ,a.observacion,a.unidad_medida,a.estado,a.tienda_codigo,a.find,
                concat('/Imagenes/Productos/',cast(a.tienda_codigo as char),
                '/',cast(b.categorias_codigo as char),'/',
                cast(b.subcategoria_codigo as char),'/',b.direccion) img_url
                ,f_datos_adicionales(a.categorias_codigo,a.subcategoria_codigo,a.codigo) stock_general
            from    vproductos a left join
                    imagenes b on b.categorias_codigo = a.categorias_codigo
                                and b.subcategoria_codigo = a.subcategoria_codigo
                                and b.productos_codigo = a.codigo
                                and b.codigo = (select min(x.codigo) from imagenes x
                                                where x.subcategoria_codigo = b.subcategoria_codigo
                                                and x.categorias_codigo = b.categorias_codigo
                                                and x.productos_codigo = b.productos_codigo)
            where   a.tienda_codigo = ntienda
                    and a.estado_codigo = 1
                    and lower(a.find) like concat('%',lower(sprimero),'%')
            order by a.categoria,a.subcategoria,a.nombre;

        end if;
    else
        select  a.codigo,a.categorias_codigo,a.subcategoria_codigo,a.categoria,a.subcategoria,a.nombre
                ,a.observacion,a.unidad_medida,a.estado,a.tienda_codigo,a.find,
                concat('/Imagenes/Productos/',cast(a.tienda_codigo as char),
                '/',cast(b.categorias_codigo as char),'/',
                cast(b.subcategoria_codigo as char),'/',b.direccion) img_url
                ,concat(f_string_stock(a.categorias_codigo,a.subcategoria_codigo,a.codigo),
                '\n',f_string_marcas(a.categorias_codigo,a.subcategoria_codigo,a.codigo))stock_general
        from    vproductos a left join
                imagenes b on b.categorias_codigo = a.categorias_codigo
                            and b.subcategoria_codigo = a.subcategoria_codigo
                            and b.productos_codigo = a.codigo
                            and b.codigo = (select min(x.codigo) from imagenes x
                                            where x.subcategoria_codigo = b.subcategoria_codigo
                                            and x.categorias_codigo = b.categorias_codigo
                                            and x.productos_codigo = b.productos_codigo)
```



```
82          where   a.tienda_codigo = ntienda
83                  and a.estado_codigo = 1
84                  and lower(a.find) like concat('%',lower(sprimero),'%')
85                  and lower(a.find) like concat('%',lower(ssegundo),'%')
86                  and lower(a.find) like concat('%',lower(stercero),'%')
87                  and lower(a.find) like concat('%',lower(scuarto),'%')
88                  and lower(a.find) like concat('%',lower(squinto),'%')
89          order by a.categoria,a.subcategoria,a.nombre;
90      end if;
91
92  END
```

Fuente: Elaboración propia

Figura 11: Código fuente del procedimiento almacenado para la búsqueda de productos.

7. Desarrollo de Sprint 2

7.1. Tarea T6

Crear proceso que devuelve los datos del stock general según el producto.

7.1.1. Diseño

Crear una función que permita recibir los parámetros del producto devolviendo el stock general junto con la unidad de medida.

7.1.2. Código

```sql
CREATE DEFINER=`vadmin`@`%` FUNCTION `f_string_stock`(an_cat int,
an_subcat int, an_pro int)
RETURNS varchar(1000) CHARSET utf8
BEGIN
    DECLARE finished INTEGER DEFAULT 0;
    DECLARE string_stock,stock varchar(500) DEFAULT "";
    DECLARE curstock
        CURSOR FOR
            select
                concat(replace(cast(sum(a.cantidad) as char),'.00',''),' ',
                ifnull(CONCAT(UPPER(LEFT(b.descripcion,1)),
                SUBSTR(lower(b.descripcion),2)),'sin U.M.')) stock
            from Stock a left join
                unidad_medida b on a.unidad_medida_codigo = b.codigo
            where a.categorias_codigo = an_cat
                    and a.subcategoria_codigo = an_subcat
                    and a.productos_codigo = an_pro
            group by b.descripcion;

    -- declare NOT FOUND handler
    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET finished = 1;

    OPEN curstock;
    getstock: LOOP
        FETCH curstock INTO stock;
        IF finished = 1 THEN
            LEAVE getstock;
        END IF;
        -- build email list
        SET string_stock = CONCAT(stock,", ",string_stock);
    END LOOP getstock;
    CLOSE curstock;
    set @stringstock = if(trim(string_stock)='','sin stock',
    substring(string_stock, 1, length(string_stock) - 2));
    return @stringstock;
END
```

Fuente: Elaboración propia

Figura 12: función que retorno el stock general.

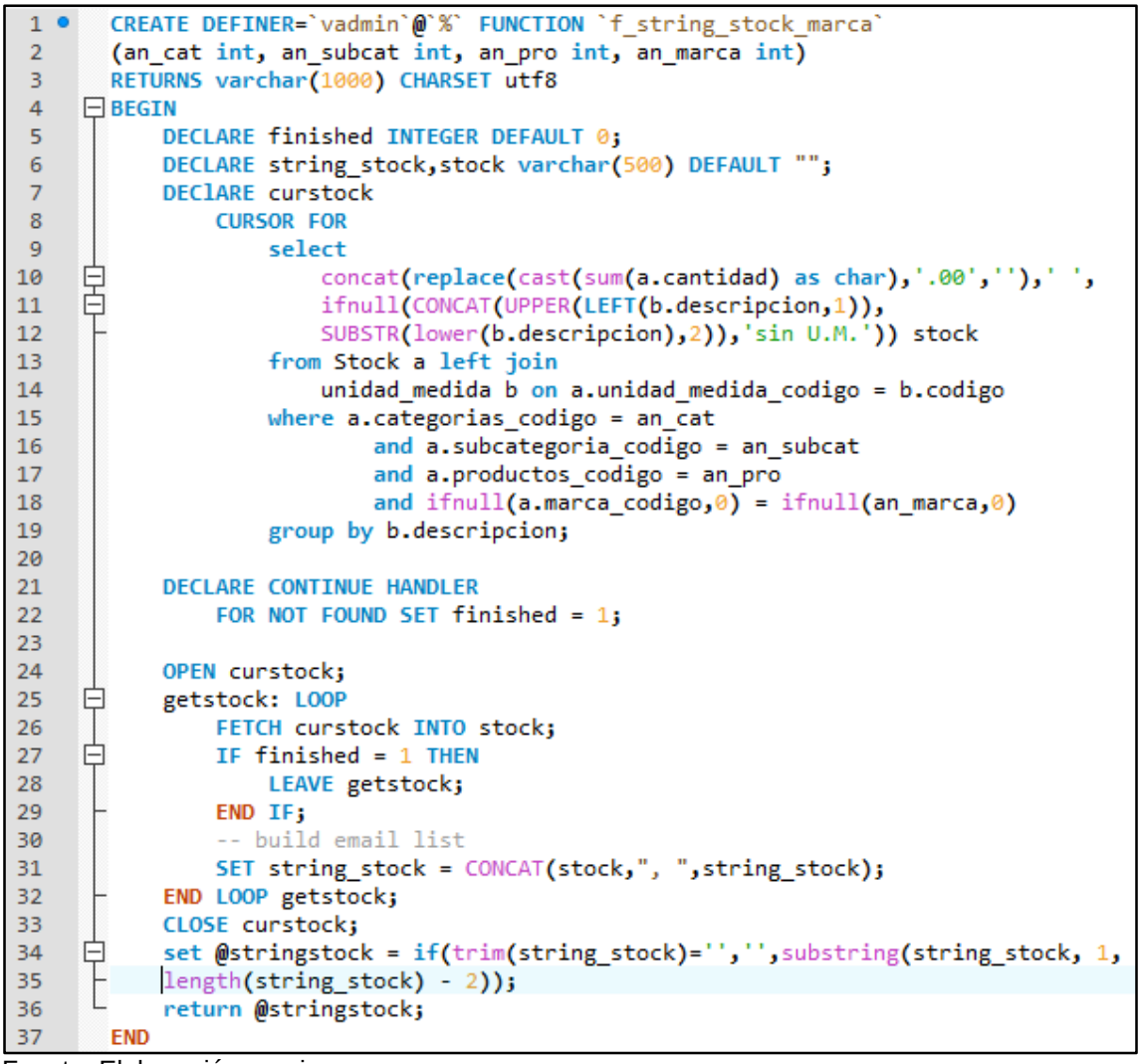
7.2. Tarea T7

Crear proceso que devuelva datos del stock según el producto y la marca.

7.2.1. Diseño

Crear una función que permita recibir los parámetros del producto y la marca devolviendo un cursor con los datos del producto.

7.2.2. Código



```
CREATE DEFINER=`vadmin`@`%` FUNCTION `f_string_stock_marca`
(an_cat int, an_subcat int, an_pro int, an_marca int)
RETURNS varchar(1000) CHARSET utf8
BEGIN
    DECLARE finished INTEGER DEFAULT 0;
    DECLARE string_stock,stock varchar(500) DEFAULT "";
    DEClARE curstock
        CURSOR FOR
            select
                concat(replace(cast(sum(a.cantidad) as char),'.00',''),' ',
                ifnull(CONCAT(UPPER(LEFT(b.descripcion,1)),
                SUBSTR(lower(b.descripcion),2)),'sin U.M.')) stock
            from Stock a left join
                unidad_medida b on a.unidad_medida_codigo = b.codigo
            where a.categorias_codigo = an_cat
                    and a.subcategoria_codigo = an_subcat
                    and a.productos_codigo = an_pro
                    and ifnull(a.marca_codigo,0) = ifnull(an_marca,0)
            group by b.descripcion;

    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET finished = 1;

    OPEN curstock;
    getstock: LOOP
        FETCH curstock INTO stock;
        IF finished = 1 THEN
            LEAVE getstock;
        END IF;
        -- build email list
        SET string_stock = CONCAT(stock,", ",string_stock);
    END LOOP getstock;
    CLOSE curstock;
    set @stringstock = if(trim(string_stock)='','',substring(string_stock, 1,
    length(string_stock) - 2));
    return @stringstock;
END
```

Fuente: Elaboración propio

Figura 13: Código fuente de la función que devuelve el stock según producto y marca.

7.3. Tarea T8

Crear proceso que devuelva datos del stock según el producto, la marca y el color.

7.3.1. Diseño

Crear una función que permita recibir los parámetros del producto, la marca y el color devolviendo un cursor con los datos del producto.

7.3.2. Código

```sql
CREATE DEFINER=`vadmin`@`%` FUNCTION `f_string_stock_marca_color`
(an_tienda int,an_cat int, an_subcat int, an_pro int, an_marca int, an_color int)
RETURNS varchar(1000) CHARSET utf8
BEGIN
    DECLARE finished INTEGER DEFAULT 0;
    DECLARE string_stock,stock varchar(500) DEFAULT "";
    DECLARE curstock
        CURSOR FOR
            select  concat(replace(cast(sum(a.cantidad) as char),'.00','')
            ,' ',b.abreviatura)
            from    Stock a left join
                    unidad_medida b on a.tienda_codigo = if(b.tienda_codigo = 0,
                    a.tienda_codigo,b.tienda_codigo)
                                    and a.unidad_medida_codigo = codigo
            where   a.tienda_codigo = an_tienda
                    and a.categorias_codigo = an_cat
                    and a.subcategoria_codigo = an_subcat
                    and a.productos_codigo = an_pro
                    and ifnull(a.marca_codigo,0) = ifnull(an_marca,0)
                    and a.color_codigo = an_color
            group by b.abreviatura;

    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET finished = 1;

    OPEN curstock;
    getstock: LOOP
        FETCH curstock INTO stock;
        IF finished = 1 THEN
            LEAVE getstock;
        END IF;
        SET string_stock = CONCAT(stock,", ",string_stock);
    END LOOP getstock;
    CLOSE curstock;
    set @stringstock = if(trim(ifnull(string_stock,''))='','Sin Color',
    substring(string_stock, 1, length(string_stock) - 2));
    return @stringstock;
END
```

Fuente: Elaboración propia

Figura 14: Código fuente de la función que devuelve stock según el productos, marca y color.

7.4. Tarea T9

Crear proceso que devuelva datos del stock según el producto, la marca, el color y la talla.

7.4.1. Diseño

Crear una función que permita recibir los parámetros del producto, la marca, el color y la talla devolviendo un cursor con los datos del producto.

7.4.2. Código

```sql
CREATE DEFINER=`vadmin`@`%` FUNCTION `f_string_stock_marca_talle`(
    an_tienda int,
    an_cat int,
    an_subcat int,
    an_pro int,
    an_marca int,
    an_color int) RETURNS varchar(1000) CHARSET utf8
BEGIN
    DECLARE finished INTEGER DEFAULT 0;
    DECLARE string_stock,stock varchar(500) DEFAULT "";
    DECLARE curstock
        CURSOR FOR
            select  concat(b.abreviatura,'(',replace(cast(sum(a.cantidad)
            as char),'.00',''),')')
            from    Stock a left join
                    tallas b on a.tallas_codigo = b.codigo
                                and a.categorias_codigo =
                                b.categorias_codigo
                                and a.subcategoria_codigo =
                                b.subcategoria_codigo
                                and a.tienda_codigo =
                                if(b.tienda_codigo=0,a.tienda_codigo,b.tienda_codigo)
            where   a.tienda_codigo = an_tienda
                    and a.categorias_codigo = an_cat
                    and a.subcategoria_codigo = an_subcat
                    and a.productos_codigo = an_pro
                    and ifnull(a.marca_codigo,0) = ifnull(an_marca,0)
                    and a.color_codigo = an_color
            group by b.abreviatura,b.codigo
            order by b.codigo;

    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET finished = 1;

    OPEN curstock;
    getstock: LOOP
        FETCH curstock INTO stock;
        IF finished = 1 THEN
            LEAVE getstock;
        END IF;
        -- build email list
        SET string_stock = CONCAT(stock," - ",string_stock);
    END LOOP getstock;
    CLOSE curstock;
    set @stringstock = if(trim(ifnull(string_stock,''))='','Sin Talle',
    substring(string_stock, 1, length(string_stock) - 3));
    return @stringstock;
END
```

Fuente: Elaboración propia

Figura 15: Código fuente de la función que devuelve el stock según producto, marca, color y talle.

7.5. Tarea T10

Proceso que devuelva datos de la ubicación de la imagen según el producto.

7.5.1. Diseño

Creamos el api que permite devolver el listado de imágenes que posee un producto.

7.5.2. Código fuente del Api de Imágenes

imagenes.php

```php
<?php
require_once 'clases/respuestas.class.php';
require_once 'clases/imagenes.class.php';

$_respuestas = new respuestas;
$_imagenes = new imagenes;

header("Content-Type: application/json");
if($_SERVER['REQUEST_METHOD'] == 'GET'){
    if(isset($_GET['tienda']) && isset($_GET['subcategoria']) &&
        isset($_GET['producto']) && isset($_GET['categoria'])){
        $resultado = $_imagenes->getImagenes($_GET['tienda'],
        $_GET['categoria'],$_GET['subcategoria'],$_GET['producto']);
        echo json_encode($resultado);
    }else{
        echo json_encode($_respuestas->error_400());
    }
}else{
    echo json_encode($_respuestas->error_405());
}

?>
```

Fuente: Elaboración propia

Figura 16: Código fuente que controla las peticiones de las imágenes.

clases > imagenes.class.php

```php
<?php
require_once 'conexion/conexion.php';

class imagenes extends conexion{

    public function getImagenes($tienda,$categoria,$subcategoria,$producto){
        $query = "select    concat('/Imagenes/Productos/',:tienda,'/',
                            cast(b.categorias_codigo as char),'/',
                            cast(b.subcategoria_codigo as char),'/',b.direccion) url_imagen
                  from      imagenes b
                  where     b.categorias_codigo = :categoria
                            and b.subcategoria_codigo = :subcategoria
                            and b.productos_codigo = :producto
                            and b.estado_codigo = 1";
        $params = [
            'tienda' => $tienda,
            'categoria' => $categoria,
            'subcategoria' => $subcategoria,
            'producto' => $producto
        ];
        $sql = parent::getParamDatos($query,$params);
        return $sql;
    }
}

?>
```

Fuente: Elaboración propia

Figura 17: Código fuente con la función de listar las imágenes.

7.6. Tarea T11

Crear el api que devuelva los datos del producto según las tareas 5,6,7,8,9,10.

7.6.1. Diseño

Creamos el api que va contener el get, post, put y delete que va a contener las funciones creadas para las tareas del 5 al 10.

7.6.2. Código fuente del Api Productos

productos.php

```php
<?php
require_once 'clases/respuestas.class.php';
require_once 'clases/productos.class.php';

$_respuesta = new respuestas;
$_producto = new productos;


header("Content-Type: application/json");
switch($_SERVER['REQUEST_METHOD']){
    /*=============================================================================================
    case 'POST':
        $postBody = file_get_contents('php://input');
        $resp = $_producto->insertProducto($postBody);
        echo json_encode($resp);

        break;
    /*=============================================================================================
    case 'GET':

        if (isset($_GET['tienda']) && isset($_GET['usuario']) && isset($_GET['producto'])){
            $resultado = $_producto->getProducto($_GET['tienda'],
                                                 $_GET['usuario'],
                                                 $_GET['producto']);
            echo json_encode($resultado);

        }elseif (isset($_GET['page']) && isset($_GET['tienda'])){
            $resultado = $_producto->getPageProductos($_GET['page'],$_GET['tienda']);
            echo json_encode($resultado);

        }elseif (isset($_GET['tienda']) && isset($_GET['buscar'])){
            $resultado = $_producto->getbuscarProducto($_GET['tienda'],$_GET['buscar']);
            echo json_encode($resultado);

        }elseif (isset($_GET['tienda']) && isset($_GET['usuario'])){
            $resultado = $_producto->getProductos($_GET['tienda'],$_GET['usuario']);
            echo json_encode($resultado);
        }

        break;

    /*=============================================================================================
    case 'PUT':

        $postBody = file_get_contents('php://input');
        $resp = $_producto->updateProducto($postBody);
        echo json_encode($resp);
        break;
```

```php
    /*=============================================================================================
    case 'DELETE':
        $postBody = file_get_contents('php://input');
        $resp = $_producto->deleteProducto($postBody);
        echo json_encode($resp);

        break;
}

?>
```

Fuente: Elaboración propia

Figura 18: Código fuente del api de gestiona las peticiones para los productos.

clases > productos.class.php

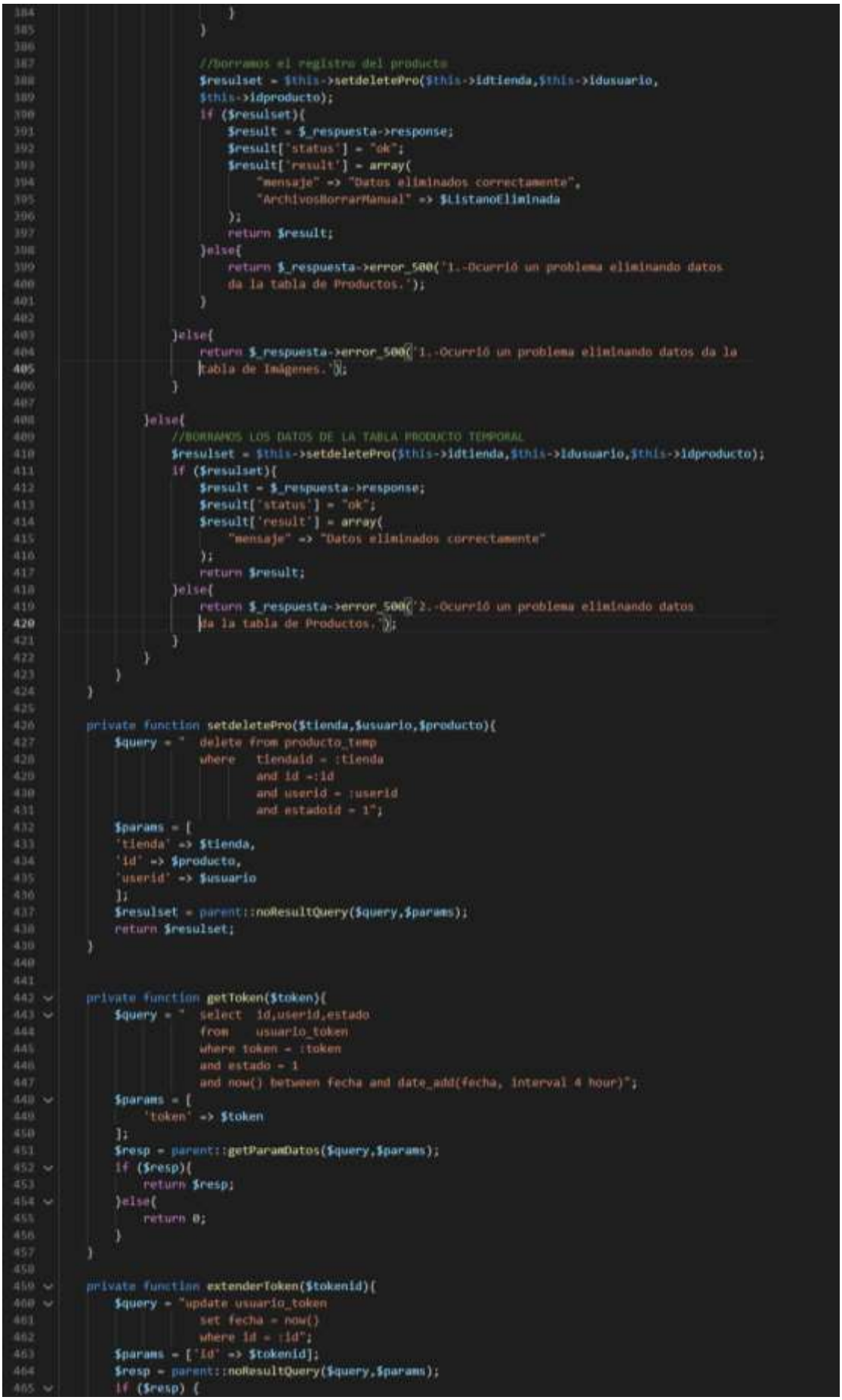
```php
<?php
require_once 'conexion/conexion.php';
require_once 'respuestas.class.php';
require_once 'clases/categoria.class.php';
require_once 'clases/subcategoria.class.php';

$_respuesta = new respuestas;

class productos extends conexion{

    private $idtienda ="";
    private $idcategoria ="";
    private $idsubcategoria ="";
    private $idproducto ="";
    private $producto ="";
    private $modelo ="";
    private $observacion ="";
    private $idusuario ="";


    //Muestra una Lista de Productos de la tabla original divididos en páginas de 100
    public function getPageProductos($pagina,$tienda){
        $inicio = 0;
        $cantidad = 100;
        if($pagina > 1){
            $inicio = ($cantidad * ($pagina - 1)) + 1;
            $cantidad = $cantidad * $pagina;
        }

        $query = "select    a.codigo,a.categorias_codigo,a.subcategoria_codigo,a.categoria,
        a.subcategoria,a.nombre,a.observacion,a.unidad_medida,a.estado,a.stock
                    from    vproductos a
                    where   a.tienda_codigo = :tienda
                    limit $inicio,$cantidad";
        $params = [
            'tienda' => $tienda
        ];
        $sql = parent::getParamDatos($query,$params);
        return $sql;
    }

    //Muestra un listado de la tabla temporal de Productos
    public function getProductos($tienda,$usuario){
        $query = "select    a.id,a.catid,a.subcatid,b.nombre categoria,c.nombre subcategoria,
        a.dsc_prod,a.dsc_obs,a.dsc_modelo
                    from    producto_temp a left join
                            categorias b on b.codigo = a.catid left join
                            subcategoria c on c.codigo = a.subcatid
                                              and c.categorias_codigo = a.catid
                where
                    a.tiendaid = :tienda
                    and a.userid = :usuario
                    and a.estadoid = 1";
        $params = [
            'tienda' => $tienda,
            'usuario' => $usuario
        ];
        $sql = parent::getParamDatos($query,$params);

```

```
        $cab = array();
        foreach ($sql as $cmarca => $vmarca) {
            $det = array();
            $detalle = array();
            foreach($vmarca as $c => $v){
                $det += [$c => $v];

                switch ($c) {
                    case "catid":
                        $categoria = $v;
                        break;

                    case 'subcatid':
                        $subcategoria = $v;
                        break;

                    case 'id':
                        $producto = $v;
                        break;
                }
            }
            $det += ["imagenes" => $this->getProductoImagen($tienda,$categoria,
            $subcategoria,$producto)];
            $cab += [$cmarca => $det];
        }

        return $cab;


    }


    public function getProductoImagen($tienda,$cat,$subcat,$pro){
        $query = "select    a.id,if(length(trim(a.dsc_dir))=0,null,concat('/Imagenes/Productos/',
        cast(:tienda as char),'/',cast(a.catid as char),'/',cast(a.subcatid as char),a.dsc_dir)) dsc_dir
                    from    imagenes_temp a
                    where   a.catid = :cat
                            and a.subcatid = :subcat
                            and a.prodid = :pro";
        $params = [
            'tienda' => $tienda,
            'cat' => $cat,
            'subcat' => $subcat,
            'pro' => $pro
        ];
        $sql = parent::getParamDatos($query,$params);
        return $sql;


    }

    //Retorna el Producto de la Tabla Temporal
    public function getProducto($tienda,$usuario,$producto){
        $query = "  select  a.id,a.catid,a.subcatid,b.nombre categoria,c.nombre subcategoria,
        a.dsc_prod,a.dsc_obs,a.dsc_modelo
                    from    producto_temp a left join
                                categorias b on b.codigo = a.catid left join
                                subcategoria c on c.codigo = a.subcatid
                                                    and c.categorias_codigo = a.catid
                    where
                        a.tiendaid = :tienda
                        and a.userid = :usuario
                        and a.id = :producto";
        $params = [
            'tienda' => $tienda,
            'producto' => $producto,
            'usuario' => $usuario
        ];
        $sql = parent::getParamDatos($query,$params);
        $cab = array();
        foreach ($sql as $columna => $valor) {
            $det = array();
            $detalle = array();
            foreach($valor as $c => $v){
                $det += [$c => $v];

                switch ($c) {
                    case "catid":
                        $categoria = $v;
```

```
                        break;

                    case 'subcatid':
                        $subcategoria = $v;
                        break;

                    case 'id':
                        $producto = $v;
                        break;
                }
            }
            $det += ["imagenes" => $this->getProductoImagen($tienda,$categoria,$subcategoria,
            $producto)];
            $cab += [$columna => $det];
        }

        return $cab;
    }

    //Busca de la tabla orginal de Producto, una coincidencia.
    public function getbuscarProducto($tienda,$buscar){
        $query = "CALL sp_buscar_producto(:buscar,:tienda)";
        $params = [
            'tienda' => $tienda,
            'buscar' => $buscar
        ];
        $sql = parent::getParamDatos($query,$params);
        return $sql;

    }

    //Obtiene el nuevo código para insertar en la tabla temporal de Productos
    private function getIDProducto($tienda){
        $query = "  select ifnull(max(id),0)+1 idproducto
                    from producto_temp
                    where
                        tiendaId = :tienda";
        $params = ['tienda' => $tienda];
        $sql = parent::getParamDatos($query,$params);
        return (int)$sql[0]['idproducto'];

    }

    //Inserta en la tabla temporal de Prodcuto, esperando ser aprobado desde el sistema principal
    public function insertProducto($json){
        $_respuesta = new respuestas;
        $datos = json_decode($json,true);
        //----------------------------------------------------------------------------INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401();
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
            }
        }
        //----------------------------------------------------------------------------FIN-TOKEN


        if (!isset($datos['idtienda']) || !isset($datos['categoria']) ||
        !isset($datos['subcategoria']) ||  !isset($datos['producto']) || !isset($datos['idusuario'])) {
            return $_respuesta->error_400();
        }else {
            $this->idtienda     = $datos['idtienda'];
            $categoria          = $datos['categoria'];
            $subcategoria       = $datos['subcategoria'];
            $this->producto     = $datos['producto'];
            $this->idusuario    = $datos['idusuario'];

            //Buscamos los id de la categoria y la subcategoria
            $_categoria = new categoria;
            $_subcategoria = new subcategoria;
            $resultado = $_categoria->getIdCategoria($this->idtienda,$categoria);
            if($resultado){
                $this->idcategoria = $resultado[0]['id'];
            }else{
                return $_respuesta->error_200('La categoria no es válida.');
            }

            $resul = $_subcategoria->getidSubCategoria($this->idtienda,$this->idcategoria,
            $subcategoria);
```

```
            if($resul){
                $this->idsubcategoria = $resul[0]['id'];
            }else{
                return $_respuesta->error_200('La subcategoria no es válida o no coincide
                con la categoria.');
            }

            if(isset($datos['modelo'])){
                $this->modelo = $datos['modelo'];
            }
            if(isset($datos['observacion'])){
                $this->observacion = $datos['observacion'];
            }
            $query = "insert into producto_temp(tiendaid,id,catid,subcatid,dsc_prod,dsc_obs,
                dsc_modelo,estadoid,userid,fecha)
            values(:tienda,:id,:catid,:subcatid,:dsc_pro,:dsc_obs,:dsc_modelo,1,:userid,now())";
            $idpro = $this->getIDProducto($this->idtienda);
            $params = [
                'tienda' => $this->idtienda,
                'id' => $idpro,
                'catid' => $this->idcategoria,
                'subcatid' => $this->idsubcategoria,
                'dsc_pro' => $this->producto,
                'dsc_obs' => $this->observacion,
                'dsc_modelo' => $this->modelo,
                'userid' => $this->idusuario
            ];

            $resulset = parent::noResultQuery($query,$params);

            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "idcat" => $this->idcategoria,
                    "idsubcat" => $this->idsubcategoria,
                    "idpro" => $idpro,
                );
                return $result;
            }else{
                return $_respuesta->error_500('Ocurrió un problema enviando datos al servidor.');
            }
        }
    }

    //Actualiza los datos de la tabla temporal de productos.
    public function updateProducto($json){

        $datos = json_decode($json,true);
        //=============================================================================INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401();
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado:");
            }
        }
        //=============================================================================FIN-TOKEN

        if (!isset($datos['idtienda']) || !isset($datos['idusuario']) || !isset($datos['idproducto']) ||
        !isset($datos['idcategoria']) || !isset($datos['idsubcategoria']) ||
        !isset($datos['producto']) || !isset($datos['modelo']) || !isset($datos['observacion'])) {
            return $_respuesta->error_400();
        }else {
            $this->idtienda         = $datos['idtienda'];
            $this->idusuario        = $datos['idusuario'];
            $this->idproducto       = $datos['idproducto'];

            $this->idcategoria      = $datos['idcategoria'];
            $this->idsubcategoria   = $datos['idsubcategoria'];

            $this->producto         = $datos['producto'];
            $this->observacion      = $datos['observacion'];
            $this->modelo           = $datos['modelo'];



            $query = "  update  producto_temp
                        set     dsc_prod = :dsc_pro,
                                dsc_obs = :dsc_obs,
```

```
                                dsc_modelo = :dsc_modelo,
                                fecha = now()
                        where   tiendaId = :tienda
                                and id = :id
                                and userid = :userid
                                and estadoid = 1";
            $params = [
                'tienda' => $this->idtienda,
                'id' => $this->idproducto,
                'dsc_pro' => $this->producto,
                'dsc_obs' => $this->observacion,
                'dsc_modelo' => $this->modelo,
                'userid' => $this->idusuario
            ];
            $resulset = parent::noResultQuery($query,$params);
            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "mensaje" => "Datos guardados correctamente"
                );
                return $result;
            }else{
                return $_respuesta->error_500('Ocurrió un problema enviando datos al servidor.');
            }
        }
    }
    //elimina un producto de la tabla temporal Productos
    public function deleteProducto($json){
        $_respuesta = new respuestas;
        $datos = json_decode($json,true);
        //----------------------------------------------------------------------------INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401('No autorizado wilder '.$datos['idusuario']);
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
            }
        }
        //----------------------------------------------------------------------------FIN-TOKEN

        if (!isset($datos['idtienda']) || !isset($datos['idcategoria']) || !isset($datos['idsubcategoria'])
        || !isset($datos['idproducto']) || !isset($datos['idusuario'])) {
            return $_respuesta->error_400();
        }else {
            $this->idtienda         = $datos['idtienda'];
            $this->idusuario        = $datos['idusuario'];
            $this->idproducto       = $datos['idproducto'];
            $this->idcategoria      = $datos['idcategoria'];
            $this->idsubcategoria   = $datos['idsubcategoria'];
            // Buscamos las imagenes que contiene este producto.
            $imagenes = $this->getProductoImagen($this->idtienda,$this->idcategoria,$this->idsubcategoria,
            $this->idproducto);

            if ($imagenes){
                //borramos el registro de imagenes asociado al producto
                $query = "delete from   imagenes_temp
                        where   catId = :categoria
                                and subcatId = :subcategoria
                                and prodId = :producto";
                $params = [
                    'categoria' => $this->idcategoria,
                    'subcategoria' => $this->idsubcategoria,
                    'producto' => $this->idproducto
                ];
                $resulset = parent::noResultQuery($query,$params);
                if ($resulset){
                    //ELIMINAMOS LOS ARCHIVOS QUE TIENEN LA TABLA DE IMAGENES
                    $ListanoEliminada = array();
                    foreach ($imagenes as $columna => $valor) {
                        $noEliminada = array();
                        foreach($valor as $c => $v){
                            if ($c == "dsc_dir"){
                                if (!unlink(".".$v)){
                                    $noEliminada += [$c => $v];
                                }
                            }
                        }
                        if($noEliminada){
                            $ListanoEliminada += [$columna => $noEliminada];
```



```
                    }
                )

                //Borramos el registro del producto
                $resulset = $this->setdeletePro($this->idtienda,$this->idusuario,
                $this->idproducto);
                if ($resulset){
                    $result = $_respuesta->response;
                    $result['status'] = "ok";
                    $result['result'] = array(
                        "mensaje" => "Datos eliminados correctamente",
                        "ArchivosBorrarManual" => $ListanoEliminada
                    );
                    return $result;
                }else{
                    return $_respuesta->error_500('1.-Ocurrió un problema eliminando datos
                    da la tabla de Productos.');
                }

            }else{
                return $_respuesta->error_500('1.-Ocurrió un problema eliminando datos da la
                tabla de Imágenes.');
            }

        }else{
            //BORRAMOS LOS DATOS DE LA TABLA PRODUCTO TEMPORAL
            $resulset = $this->setdeletePro($this->idtienda,$this->idusuario,$this->idproducto);
            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "mensaje" => "Datos eliminados correctamente"
                );
                return $result;
            }else{
                return $_respuesta->error_500('2.-Ocurrió un problema eliminando datos
                da la tabla de Productos.');
            }
        }
    }
}

private function setdeletePro($tienda,$usuario,$producto){
    $query = "  delete from producto_temp
                where   tiendaid = :tienda
                        and id =:id
                        and userid = :userid
                        and estadoid = 1";
    $params = [
    'tienda' => $tienda,
    'id' => $producto,
    'userid' => $usuario
    ];
    $resulset = parent::noResultQuery($query,$params);
    return $resulset;
}


private function getToken($token){
    $query = "  select  id,userid,estado
                from    usuario_token
                where token = :token
                and estado = 1
                and now() between fecha and date_add(fecha, interval 4 hour)";
    $params = [
        'token' => $token
    ];
    $resp = parent::getParamDatos($query,$params);
    if ($resp){
        return $resp;
    }else{
        return 0;
    }
}

private function extenderToken($tokenid){
    $query = "update usuario_token
                set fecha = now()
                where id = :id";
    $params = ['id' => $tokenid];
    $resp = parent::noResultQuery($query,$params);
    if ($resp) {
```

```
            return $resp;
        }else{
            return 0;
        }
    }


}

?>
```

Fuente: Elaboración propia

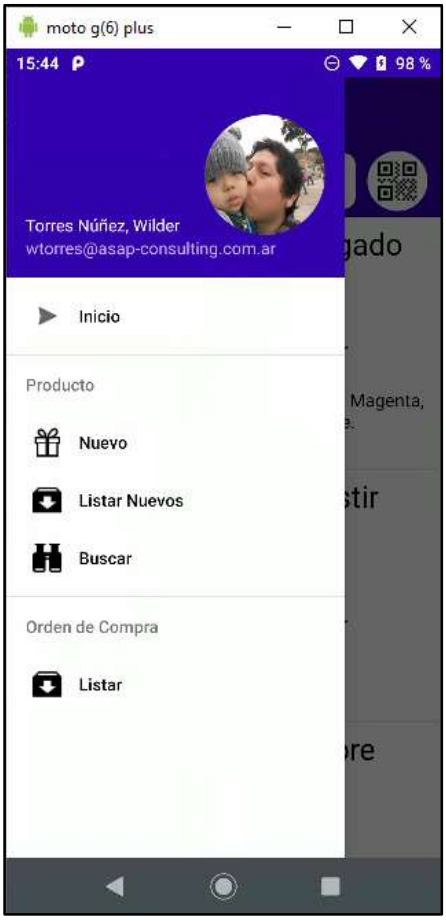
Figura 19: Código fuente del api que contiene las funciones para los productos.

7.7. Tarea T12

Crear la pantalla que muestra el listado de productos en la aplicación móvil.

7.7.1. Diseño

Se va a crear la pantalla que contenga el listado de los productos, estos deben contener las funciones de búsqueda inteligente, información sobre el stock general y los colores que tenga según el stock que posea, también debe permitir ver todas las imágenes referentes a los productos como el detalle de existencias que posea el mismo.



Fuente: Elaboración propia

Figura 20: Menú listar productos (Buscar).

Fuente: Elaboración propia

Figura 21: Pantalla principal de búsqueda de productos.

Fuente: Elaboración propia

Figura 22: Pantalla emergente que muestra las imágenes de los productos.

Fuente: Elaboración propia

Figura 23: Pantalla emergente detallando la información del producto.

7.7.2. Código fuente de las ventanas de productos

```java
package com.sebasoft.tienda11.ui.fragment;

import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.db.Usuario;
import com.sebasoft.tienda11.db.dbconfiguracion;
import com.sebasoft.tienda11.esquema.Aplicacion;
import com.sebasoft.tienda11.esquema.Producto;
import com.sebasoft.tienda11.ui.controller.adapter_Articulo;
import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

public class fragment_BuscarArticulo extends Fragment implements View.OnClickListener{

    private EditText Codigo;
    private ImageButton scanBtn;
    private ProgressBar pb_buscar;
    private ListView lista;
    private String Servidor;
    private int idtienda,iduser;
    private Usuario cuser;
    FrameLayout fragmentContainer;

    private fragment_CrearArticulo.OnFragmentInteractionListener mListener;
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {

        View rootView = null;
        Aplicacion app = (Aplicacion) getActivity().getApplicationContext();

        if (app.getArticulo_info() == 1) {
            app.setArticulo_info(0);
        }
            rootView = inflater.inflate(R.layout.fragment_buscararticulo, container, false);
            dbconfiguracion dbconf = new dbconfiguracion(rootView.getContext());

            cuser = new Usuario(getContext());
            cuser._iniciarDatosUsuario();
            Servidor = dbconf.getServidor();
            idtienda = cuser.getTienda();
            iduser = cuser.getUsuario();

            fragmentContainer = (FrameLayout) rootView.findViewById(R.id.fragment_marca);
            scanBtn = (ImageButton) rootView.findViewById(R.id.ibt_scan);
            Codigo = (EditText) rootView.findViewById(R.id.et_coditem);
            pb_buscar = (ProgressBar) rootView.findViewById(R.id.pb_buscar);
            lista = (ListView) rootView.findViewById(R.id.lv_articulos);
            /*Forzamos cuando va el foco en el edittext codigo, se abra el teclado*/
            Codigo.requestFocus();
            InputMethodManager imm = (InputMethodManager) getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
            imm.toggleSoftInput(InputMethodManager.SHOW_FORCED, InputMethodManager.HIDE_IMPLICIT_ONLY);

            scanBtn.setOnClickListener(this);
            /*Buscaquamos los productos*/
            Codigo.setOnKeyListener(new View.OnKeyListener() {

                @Override
                public boolean onKey(View arg0, int arg1, KeyEvent arg2) {
                    String Item;
                    if (arg1 == KeyEvent.KEYCODE_ENTER) {
                        Item = Codigo.getText().toString();
                        InputMethodManager imm = (InputMethodManager) getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
                        imm.hideSoftInputFromWindow(Codigo.getWindowToken(), 0);
                        scanBtn.setClickable(false);
                        pb_buscar.setVisibility(View.VISIBLE);
                        onBuscarProducto(Item);

                        //listado.setVisibility(View.VISIBLE);
                        return true;
```

```
                }
                return false;
            }
        });

        return rootView;
    }



    @Override
    public void onClick(View view) {
        /*Aqui va el codigo del boton scan*/
        switch (view.getId()){
            case R.id.ibt_scan:
                //agregamos las acciones de presionar el boton de escanear
                break;
        }

    }

    public void onBuscarProducto(String item) {
        String tienda =Integer.toString(idtienda);
        String servicio = Servidor+"/"+"productos?tienda="+tienda+"&buscar="+item;

        RequestQueue queue = Volley.newRequestQueue(getContext());
        StringRequest postRequest = new StringRequest( Request.Method.GET, servicio,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        onCargarLista(new String(response));
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(getContext(),"Api sin respuestas",Toast.LENGTH_SHORT).show();
                    }
                });

        queue.add(postRequest);
    }

    public void onCargarLista(String response){
        Producto producto;

        ArrayList<Producto> aproductos = new ArrayList<Producto>();
        try {
            JSONArray jsonArray = new JSONArray(response);
            for(int i=0;i<jsonArray.length();i++){
                producto = new Producto(
                        i,
                        Integer.parseInt(jsonArray.getJSONObject(i).getString("tienda_codigo")),
                        Integer.parseInt(jsonArray.getJSONObject(i).getString("categorias_codigo")),
                        Integer.parseInt(jsonArray.getJSONObject(i).getString("subcategoria_codigo")),
                        Integer.parseInt(jsonArray.getJSONObject(i).getString("codigo")),
                        0,
                        0,
                        jsonArray.getJSONObject(i).getString("categoria"),
                        jsonArray.getJSONObject(i).getString("subcategoria"),
                        jsonArray.getJSONObject(i).getString("nombre"),
                        jsonArray.getJSONObject(i).getString("observacion"),
                        jsonArray.getJSONObject(i).getString("unidad_medida"),
                        jsonArray.getJSONObject(i).getString("estado"),
                        jsonArray.getJSONObject(i).getString("img_url"),
                        jsonArray.getJSONObject(i).getString("stock_general")
                        );
                aproductos.add(producto);
            }

            //creando el adapter personalizado
            adapter_Articulo adapter= new adapter_Articulo(getActivity(),aproductos,getParentFragmentManager());
            lista.setAdapter(adapter);
            scanBtn.setClickable(true);
            pb_buscar.setVisibility(View.INVISIBLE);

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }



    // TODO: Rename method, update argument and hook method into UI event
    public void onButtonPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof Fragment_CrearArticulo.OnFragmentInteractionListener) {
            mListener = (Fragment_CrearArticulo.OnFragmentInteractionListener) context;
        } else {
            throw new RuntimeException(context.toString()
```

```
            throw new RuntimeException(context.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }
    public interface OnFragmentInteractionListener {
        // TODO: Update argument type and name
        void onFragmentInteraction(Uri uri);
    }
}
```

Fuente: Elaboración propia

Figura 24: Código fuente de la ventana principal de búsqueda de productos.

```
package com.sebasoft.tienda11.ui.controller;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import com.bumptech.glide.Glide;
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.db.dbconfiguracion;
import com.sebasoft.tienda11.esquema.Aplicacion;
import com.sebasoft.tienda11.esquema.Producto;
import java.util.ArrayList;

public class adapter_Articulo extends BaseAdapter {
    protected Activity activity;
    protected ArrayList<Producto> producto;
    private String Servidor;
    FragmentManager fm_base;
    Aplicacion app;
    df_articulo_info fragment2;

    public adapter_Articulo(Activity activity,ArrayList<Producto> producto,FragmentManager fm_base){
        this.activity = activity;
        this.producto = producto;
        this.fm_base = fm_base;
        app = (Aplicacion) activity.getApplicationContext();
    }

    @Override
    public int getCount() {
        return producto.size();
    }

    @Override
    public Object getItem(int position) {
        return producto.get(position);
    }

    @Override
    public long getItemId(int position) {
        return producto.get(position).getId();
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        View v = view;
        if (view == null) {
            LayoutInflater inf = (LayoutInflater) activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            v = inf.inflate(R.layout.adapter_articulo, null);
        }
        final Producto item = (Producto) getItem(i);
        dbconfiguracion dbconf = new dbconfiguracion(v.getContext());
        Servidor = dbconf.getServidor();
        String ls_imagen_url,ls_anulado;



        ImageView iv_imagen = (ImageView) v.findViewById(R.id.iv_principal_articulo);
        TextView  tv_codigo = (TextView) v.findViewById(R.id.tv_codigo_articulo);
        TextView  tv_anulado = (TextView) v.findViewById(R.id.tv_anulado_articulo);
        TextView  tv_categoria = (TextView) v.findViewById(R.id.tv_categoria_articulo);
        TextView  tv_subcategoria = (TextView) v.findViewById(R.id.tv_subcategoria_articulo);
```

```
        TextView  tv_producto = (TextView) v.findViewById(R.id.tv_producto_articulo);
        TextView  tv_adicional = (TextView) v.findViewById(R.id.tv_adicional_articulo);
        LinearLayout ly_info = (LinearLayout) v.findViewById(R.id.ly_info);


        tv_codigo.setText("Código "+Integer.toString(item.getIdcategoria()) +
                                    Integer.toString(item.getIdsubcategoria()) +
                                    Integer.toString(item.getIdProducto()));

        ls_anulado = item.getEstado();
        if(!ls_anulado.equals("Vigente")){
            tv_anulado.setVisibility(View.VISIBLE);
        }

        tv_categoria.setText(item.getCategoria());
        tv_subcategoria.setText(item.getSubcategoria());
        tv_producto.setText(item.getProducto());
        tv_adicional.setText(item.getStock());
        ls_imagen_url = item.getUrl_imagen();
        if(!ls_imagen_url.equals("null")){
            Glide.with(v.getContext())
                    .load(Servidor+ls_imagen_url)
                    .into(iv_imagen);


        }else{
            iv_imagen.setImageResource(R.mipmap.tienda);
        }

        iv_imagen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                df_articulo_imagen articulo_imagen = new df_articulo_imagen(item);
                articulo_imagen.show(fm_base,"Imagenes");
                android.app.Fragment frag = activity.getFragmentManager().findFragmentByTag("Imagenes");
                if (frag!=null){
                    activity.getFragmentManager().beginTransaction().remove(frag).commit();
                }
            }
        });
        ly_info.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                    fragment2 = new df_articulo_info(item);
                    FragmentManager fragmentManager = fm_base;
                    FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.setCustomAnimations(android.R.anim.fade_in, android.R.animator.fade_out);
                    fragmentTransaction.replace(R.id.fragment_marca, fragment2);
                    fragmentTransaction.commit();
                    fragmentTransaction.addToBackStack(null);
                    app.setFragmenadd();

            }
        });



        return v;
    }

    private void showToast(String mensaje){
        Toast.makeText(activity,mensaje, Toast.LENGTH_SHORT).show();
    }
}
```

Fuente: Elaboración propia

Figura 25: Código fuente de la pantalla principal que devuelve los productos.

```
1 package com.sebasoft.tienda11.ui.controller;
2
3 import android.os.Bundle;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.Toast;
8 import androidx.fragment.app.DialogFragment;
9 import com.android.volley.Request;
10 import com.android.volley.RequestQueue;
11 import com.android.volley.Response;
12 import com.android.volley.VolleyError;
13 import com.android.volley.toolbox.StringRequest;
14 import com.android.volley.toolbox.Volley;
15 import com.sebasoft.tienda11.R;
16 import com.sebasoft.tienda11.db.Usuario;
17 import com.sebasoft.tienda11.db.dbconfiguracion;
18 import com.sebasoft.tienda11.esquema.Producto;
19 import org.imaginativeworld.whynotimagecarousel.ImageCarousel;
```

```java
import org.imaginativeworld.whynotimagecarousel.model.CarouselItem;
import org.json.JSONArray;
import org.json.JSONException;
import java.util.ArrayList;
import java.util.List;

public class df_articulo_imagen extends DialogFragment {

    Producto producto;
    private String Servidor,title_imagen;
    private int idtienda,iduser;
    private ImageCarousel carousel;

    public df_articulo_imagen(Producto producto){
        this.producto = producto;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstaceState) {
        View view = inflater.inflate(R.layout.df_articulo_imagen, container);
        dbconfiguracion dbconf = new dbconfiguracion(view.getContext());
        Usuario cuser = new Usuario(getContext());
        cuser._iniciarDatosUsuario();
        Servidor = dbconf.getServidor();
        idtienda = cuser.getTienda();
        iduser = cuser.getUsuario();
        title_imagen = producto.getProducto();
        carousel = view.findViewById(R.id.carousel);
        carousel.registerLifecycle(getLifecycle());
        List<CarouselItem> list = new ArrayList<>();
        onCargarImagenes(list);
        return view;
    }

    private void onCargarImagenes(List<CarouselItem> list) {
        String servicio;
        servicio = Servidor+"/imagenes?tienda="+Integer.toString(idtienda)+
                "&categoria="+Integer.toString(producto.getIdcategoria())+
                "&subcategoria="+Integer.toString(producto.getIdsubcategoria())+
                "&producto="+Integer.toString(producto.getIdProducto());

        RequestQueue queue = Volley.newRequestQueue(getContext());
        StringRequest postRequest = new StringRequest( Request.Method.GET, servicio,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        ArrayList<String> categorias;
                        onListado(new String(response),list);
                        carousel.setData(list);
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(getContext(),"No se pudo cargar imagenes",Toast.LENGTH_SHORT).show();
                    }
                });

        queue.add(postRequest);
    }
    public void onListado(String response,List<CarouselItem> list){
        try {
            JSONArray jsonArray = new JSONArray(response);
            for(int i=0;i<jsonArray.length();i++){
                String url_imagen = Servidor+jsonArray.getJSONObject(i).getString("url_imagen");
                list.add(new CarouselItem(url_imagen,title_imagen));
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

Fuente: Elaboración propia

Figura 26: Código fuente de la pantalla emergente de muestra las imágenes de los productos.

8. Desarrollo de Sprint 3

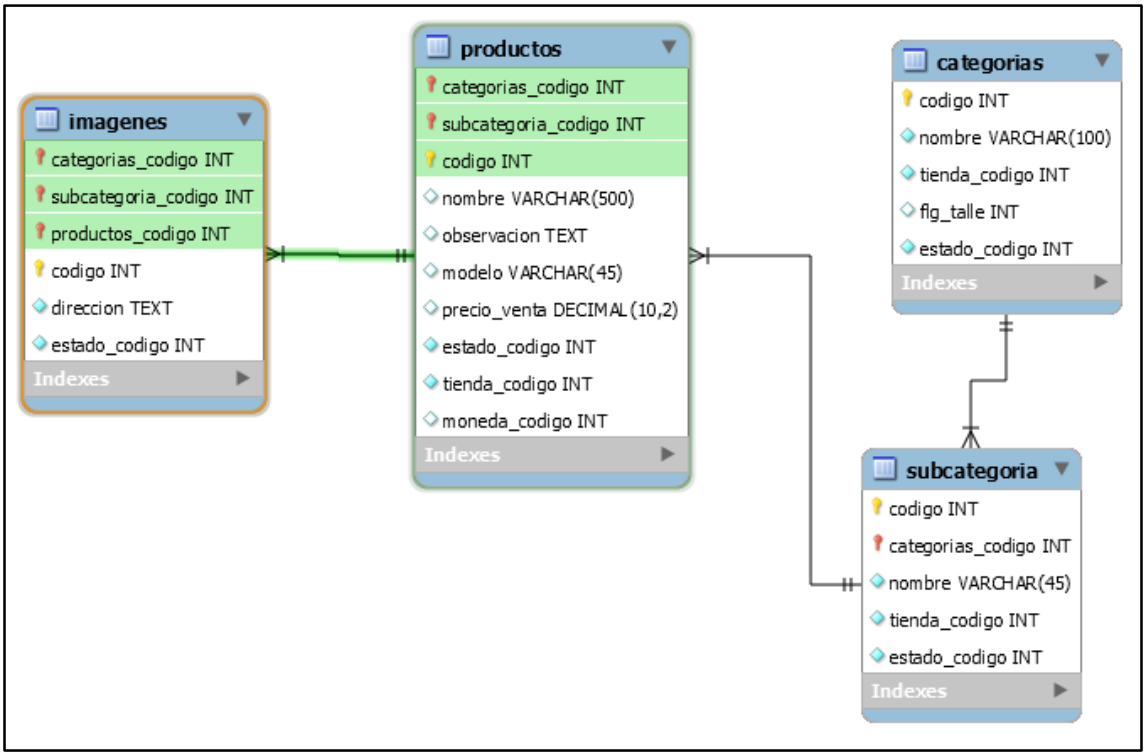
8.1. Tarea T13

Crear el esquema de producto nuevo.

8.1.1. Diseño

Se necesita crear la tabla de producto nuevo, la tabla de imagen de producto nuevo.

8.1.2. Código



Fuente: Elaboración propia

Figura 27: Esquema de los productos nuevos y sus imágenes.

8.2. Tarea T14

Crear función para verificar el token.

8.2.1. Diseño

Se necesita crear el api de producto nuevo la función que permita validar el token enviado por el aplicativo móvil.

8.2.2. Código

```
442     private function getToken($token){
443         $query = "  select  id,userid,estado
444                     from    usuario_token
445                     where token = :token
446                     and estado = 1
447                     and now() between fecha and date_add(fecha, interval 4 hour)";
448         $params = [
449             'token' => $token
450         ];
451         $resp = parent::getParamDatos($query,$params);
452         if ($resp){
453             return $resp;
454         }else{
455             return 0;
456         }
457     }
458 
459     private function extenderToken($tokenid){
460         $query = "update usuario_token
461                     set fecha = now()
462                     where id = :id";
463         $params = ['id' => $tokenid];
464         $resp = parent::noResultQuery($query,$params);
465         if ($resp) {
466             return $resp;
467         }else{
468             return 0;
469         }
470     }
471 
```

Fuente: Elaboración propia

Figura 28: Código fuente de verificación del token en el api de productos nuevos.

8.3. Tarea T15

Crear api para ingresar un producto nuevo.

8.3.1. Diseño

Se necesita crear el api para el ingreso de producto nuevo.

8.3.2. Código

productos.php

```php
<?php
require_once 'clases/respuestas.class.php';
require_once 'clases/productos.class.php';

$_respuesta = new respuestas;
$_producto = new productos;


header("Content-Type: application/json");
switch($_SERVER['REQUEST_METHOD']){
    /*=============================================================
    case 'POST':
        $postBody = file_get_contents('php://input');
        $resp = $_producto->insertProducto($postBody);
        echo json_encode($resp);

        break;

}

?>
```

Fuente: Elaboración propia

Figura 29: Código fuente del api que recibe peticiones de producto nuevo.

```php
//Inserta en la tabla temporal de Producto, esperando ser aprobado desde el sistema principal
public function insertProducto($json){
    $_respuesta = new respuestas;
    $datos = json_decode($json,true);
    //---------------------------------------------------------------------INI-TOKEN
    if(!isset($datos['token'])){
        return $_respuesta->error_401();
    }else {
        $tokenarray = $this->getToken($datos['token']);
        if(!$tokenarray){
            return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
        }
    }
    //---------------------------------------------------------------------FIN-TOKEN


    if (!isset($datos['idtienda']) || !isset($datos['categoria']) ||
    !isset($datos['subcategoria']) ||  !isset($datos['producto']) || !isset($datos['idusuario'])) {
        return $_respuesta->error_400();
    }else {
        $this->idtienda     = $datos['idtienda'];
        $categoria          = $datos['categoria'];
        $subcategoria       = $datos['subcategoria'];
        $this->producto     = $datos['producto'];
        $this->idusuario    = $datos['idusuario'];

        //Buscamos los id de la categoria y la subcategoria
        $_categoria = new categoria;
        $_subcategoria = new subcategoria;
        $resultado = $_categoria->getIdCategoria($this->idtienda,$categoria);
        if($resultado){
            $this->idcategoria = $resultado[0]['id'];
        }else{
            return $_respuesta->error_200('La categoria no es válida.');
        }

        $resul = $_subcategoria->getidSubCategoria($this->idtienda,$this->idcategoria,
```

```
            $subcategoria);
            if($resul){
                $this->idsubcategoria = $resul[0]['id'];
            }else{
                return $_respuesta->error_200('La subcategoría no es válida o no coincide
                con la categoría.');
            }

            if(isset($datos['modelo'])){
                $this->modelo = $datos['modelo'];
            }
            if(isset($datos['observacion'])){
                $this->observacion = $datos['observacion'];
            }
            $query = "insert into producto_temp(tiendaId,id,catId,subcatId,dsc_prod,dsc_obs,
                dsc_modelo,estadoId,userId,fecha)
            values(:tienda,:id,:catId,:subcatId,:dsc_pro,:dsc_obs,:dsc_modelo,1,:userId,now())";
            $idpro = $this->getIDProducto($this->idtienda);
            $params = [
                'tienda' => $this->idtienda,
                'id' => $idpro,
                'catId' => $this->idcategoria,
                'subcatId' => $this->idsubcategoria,
                'dsc_pro' => $this->producto,
                'dsc_obs' => $this->observacion,
                'dsc_modelo' => $this->modelo,
                'userId' => $this->idusuario
            ];

            $resulset = parent::noResultQuery($query,$params);

            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "idcat" => $this->idcategoria,
                    "idsubcat" => $this->idsubcategoria,
                    "idpro" => $idpro,
                );
                return $result;
            }else{
                return $_respuesta->error_500('Ocurrió un problema enviando datos al servidor.');
            }
        }
    }
```

Fuente: Elaboración propia

Figura 30: Código fuente, extraído del api procesos de productos.

8.4. Tarea T16

Crear api para ingresar imágenes asociados al producto nuevo.

8.4.1. Diseño

Se necesita crear el api que permita agregar imágenes a un producto recién creado.

8.4.2. Código

subir_imagen.php

```php
<?php

require_once 'clases/respuestas.class.php';
require_once 'clases/subir_imagen.class.php';

$_respuestas = new respuestas;
$_subir_imagen = new subir_imagen;

header("Content-Type: application/json");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $postBody = file_get_contents('php://input');
    $resp = $_subir_imagen->insertImagen($postBody);
    echo json_encode($resp);
}else{
    echo json_encode($_respuestas->error_405());
}


?>
```

Fuente: Elaboración propia

Figura 31: Código fuente del receptor de peticiones de subir imágenes.

clases > subir_imagen.class.php

```php
<?php
require_once 'conexion/conexion.php';
require_once 'respuestas.class.php';

class subir_imagen extends conexion{

    private $tienda = "";
    private $categoria = "";
    private $subcategoria = "";
    private $producto = "";
    private $imagen = "";
    private $id = "";
    private $usuario = "";
    private $archivo ="";

    public function insertImagen($json){
        $_respuestas = new respuestas;
        $datos = json_decode($json,true);
        //=============================================================================INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuestas->error_401();
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuestas->error_401("El token que ha enviado es inválido o ha caducado");
            }
        }
        //=============================================================================FIN-TOKEN

        if(!isset($datos['imagen']) || !isset($datos['idcat']) ||
        !isset($datos['idsubcat']) || !isset($datos['idpro']) ||
        !isset($datos['idtienda']) || !isset($datos['iduser'])){
            echo json_encode($_respuestas->error_400());
        }else{
            $this->tienda = $datos['idtienda'];
            $this->categoria = $datos['idcat'];
            $this->subcategoria = $datos['idsubcat'];
            $this->producto = $datos['idpro'];
            $this->usuario = $datos['iduser'];
            $this->imagen = $datos['imagen'];
            $this->id = $this->getIDImagen($this->categoria,
                                           $this->subcategoria,
                                           $this->producto);
```

```
44
45              /*Copiamos a los Archivos*/
46              $Path = "Imagenes/Productos";
47              $path_tienda = "/".$this->tienda;
48              $path_categoria = "/".$this->categoria;
49              $path_subcategoria = "/".$this->subcategoria;
50
51              $this->archivo = "/api_".$this->producto."_"
52                               .$this->id."_".date('d.m.Y.H.i.s').".jpeg";
53              /*Creamos la Carpeta si no Existe*/
54              if (!file_exists($Path.$path_tienda.
55                               $path_categoria.
56                               $path_subcategoria)) {
57                  mkdir($Path.$path_tienda.
58                        $path_categoria.
59                        $path_subcategoria, 0777, true);
60              }
61              /*Insertamos el Archivo*/
62              $data = base64_decode($this->imagen);
63              $filepath = $Path.$path_tienda.
64                          $path_categoria.
65                          $path_subcategoria.
66                          $this->archivo;
67
68              file_put_contents($filepath, $data);
69              chmod ($filepath, 0644);
70
71
72
73              /*Insertamos en la tabla*/
74              $query = "insert into imagenes_temp(catid,subcatid,prodid,id,dsc_dir,estadoid,userid,fecha)
75              values(:catid,:subcatid,:prodid,:id,:dsc_dir,1,:userid,now())";
76              $params = [
77                  'catid' => (int)$this->categoria,
78                  'subcatid' => (int)$this->subcategoria,
79                  'prodid' => (int)$this->producto,
80                  'id' => (int)$this->id,
81                  'dsc_dir' => $this->archivo,
82                  'userid' => (int)$this->usuario
83              ];
84              $resulset = parent::noResultQuery($query,$params);
85              if ($resulset){
86                  $result = $_respuestas->response;
87                  $result['status'] = "ok";
88                  $result['result'] = array(
89                      "mensaje" => "Se insertó"
90                  );
91                  return $result;
92              }else{
93                  return $_respuestas->error_500('Ocurrió un problema enviando datos al servidor.');
94              }
95          }
96
97      }
98      //Obtiene el nuevo código para insertar en la Tabla Temporal de Imagenes
99      private function getIDImagen($categoria,$subcategoria,$producto){
100         $query = "  select  ifnull(max(id),0)+1 idimagen
101                     from    imagenes_temp
102                     where   catid = :categoria
103                             and subcatid = :subcategoria
104                             and prodid = :producto";
105         $params = [
106             'categoria' => $categoria,
107             'subcategoria' => $subcategoria,
108             'producto' => $producto
109         ];
110         $sql = parent::getParamDatos($query,$params);
111         return $sql[0]['idimagen'];
112
113     }
114
115     private function getToken($token){
116         $query = "  select  id,userid,estado
```

```
                    from    usuario_token
                    where token = :token
                    and estado = 1
                    and now() between fecha and date_add(fecha, interval 4 hour)";
        $params = [
            'token' => $token
        ];
        $resp = parent::getParamDatos($query,$params);
        if ($resp){
            return $resp;
        }else{
            return 0;
        }
    }
}

?>
```

Fuente: Elaboración propia

Figura 32: Código fuente de las funciones en general del api de subir imágenes.

8.5. Tarea T17

Crear api para modificar un producto nuevo.

8.5.1. Diseño

Crear api que permita modificar el producto nuevo.

8.5.2. Código

productos.php

```php
<?php
require_once 'clases/respuestas.class.php';
require_once 'clases/productos.class.php';

$_respuesta = new respuestas;
$_producto = new productos;


header("Content-Type: application/json");
switch($_SERVER['REQUEST_METHOD']){
    /*------------------------------------------------------------

    case 'PUT':

        $postBody = file_get_contents('php://input');
        $resp = $_producto->updateProducto($postBody);
        echo json_encode($resp);
        break;
    /*------------------------------------------------------------
    case 'DELETE':
        $postBody = file_get_contents('php://input');
        $resp = $_producto->deleteProducto($postBody);
        echo json_encode($resp);

        break;
}

?>
```

Fuente: Elaboración propia

Figura 33: Código fuente del api peticiones de Productos nuevos.

```
    //Actualiza los datos de la tabla temporal de productos.
    public function updateProducto($json){

        $datos = json_decode($json,true);
        //=============================================================================INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401();
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado:");
            }
        }
        //=============================================================================FIN-TOKEN

        if (!isset($datos['idtienda']) || !isset($datos['idusuario']) || !isset($datos['idproducto']) ||
        !isset($datos['idcategoria']) || !isset($datos['idsubcategoria']) ||
        !isset($datos['producto']) || !isset($datos['modelo']) || !isset($datos['observacion'])) {
            return $_respuesta->error_400();
        }else {
            $this->idtienda         = $datos['idtienda'];
            $this->idusuario        = $datos['idusuario'];
            $this->idproducto       = $datos['idproducto'];

            $this->idcategoria      = $datos['idcategoria'];
            $this->idsubcategoria   = $datos['idsubcategoria'];

            $this->producto         = $datos['producto'];
            $this->observacion      = $datos['observacion'];
            $this->modelo           = $datos['modelo'];



            $query = "  update  producto_temp
                        set     dsc_prod = :dsc_pro,
                                dsc_obs = :dsc_obs,
                                dsc_modelo = :dsc_modelo,
                                fecha = now()
                        where   tiendaid = :tienda
                                and id = :id
                                and userid = :userid
                                and estadoid = 1";
            $params = [
                'tienda' => $this->idtienda,
                'id' => $this->idproducto,
                'dsc_pro' => $this->producto,
                'dsc_obs' => $this->observacion,
                'dsc_modelo' => $this->modelo,
                'userid' => $this->idusuario
            ];
            $resulset = parent::noResultQuery($query,$params);
            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "mensaje" => "Datos guardados correctamente"
                );
                return $result;
            }else{
                return $_respuesta->error_500('Ocurrió un problema enviando datos al servidor.');
            }
        }
    }
    //elimina un producto de la tabla temporal Productos
    public function deleteProducto($json){
        $_respuesta = new respuestas;
        $datos = json_decode($json,true);
        //=============================================================================INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401('No autorizado wilder '.$datos['idusuario']);
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
            }
        }
        //=============================================================================FIN-TOKEN

        if (!isset($datos['idtienda']) || !isset($datos['idcategoria']) || !isset($datos['idsubcategoria'])
        || !isset($datos['idproducto']) || !isset($datos['idusuario'])) {
            return $_respuesta->error_400();
        }else {
            $this->idtienda         = $datos['idtienda'];
            $this->idusuario        = $datos['idusuario'];
            $this->idproducto       = $datos['idproducto'];
            $this->idcategoria      = $datos['idcategoria'];
            $this->idsubcategoria   = $datos['idsubcategoria'];
```

```
        // Buscamos las imagenes que contiene este producto.
        $imagenes = $this->getProductoImagen($this->idtienda,$this->idcategoria,$this->idsubcategoria,$this->idproducto);

        if ($imagenes){
            //borramos el registro de imagenes asociado al producto
            $query = "delete from  imagenes_temp
                    where  catid = :categoria
                           and subcatid = :subcategoria
                           and prodid = :producto";
            $params = [
                'categoria' => $this->idcategoria,
                'subcategoria' => $this->idsubcategoria,
                'producto' => $this->idproducto
            ];
            $resulset = parent::noResultQuery($query,$params);
            if ($resulset){
                //ELIMINAMOS LOS ARCHIVOS QUE TIENEN LA TABLA DE IMAGENES
                $ListanoEliminada = array();
                foreach ($imagenes as $columna => $valor) {
                    $noEliminada = array();
                    foreach($valor as $c => $v){
                        if ($c == "dsc_dir"){
                            if (!unlink(".".$v)){
                                $noEliminada += [$c => $v];
                            }
                        }
                    }
                    if($noEliminada){
                        $ListanoEliminada += [$columna => $noEliminada];
                    }
                }

                //borramos el registro del producto
                $resulset = $this->setdeletePro($this->idtienda,$this->idusuario,$this->idproducto);
                if ($resulset){
                    $result = $_respuesta->response;
                    $result['status'] = "ok";
                    $result['result'] = array(
                        "mensaje" => "Datos eliminados correctamente",
                        "ArchivosBorrarManual" => $ListanoEliminada
                    );
                    return $result;
                }else{
                    return $_respuesta->error_500('1.-Ocurrió un problema eliminando datos da la tabla de Productos.');
                }

            }else{
                return $_respuesta->error_500('1.-Ocurrió un problema eliminando datos da la tabla de Imagenes.');
            }

        }else{
            //BORRAMOS LOS DATOS DE LA TABLA PRODUCTO TEMPORAL
            $resulset = $this->setdeletePro($this->idtienda,$this->idusuario,$this->idproducto);
            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "mensaje" => "Datos eliminados correctamente"
                );
                return $result;
            }else{
                return $_respuesta->error_500('2.-Ocurrió un problema eliminando datos da la tabla de Productos.');
            }
        }
    }
}
```

Fuente: Elaboración propia

Figura 34: Código fuente api actualizar productos nuevos.

8.6. Tarea T18

Crear api para eliminar un producto nuevo.

8.6.1. Diseño

Crear api que permita eliminar el producto nuevo.

8.6.2. Código

```
412     private function setdeletePro($tienda,$usuario,$producto){
413         $query = "  delete from producto_temp
414                     where   tiendaid = :tienda
415                             and id =:id
416                             and userid = :userid
417                             and estadoid = 1";
418         $params = [
419         'tienda' => $tienda,
420         'id' => $producto,
421         'userid' => $usuario
422         ];
423         $resulset = parent::noResultQuery($query,$params);
424         return $resulset;
425     }
```

Fuente: Elaboración propia.

Figura 35: Código fuente del api eliminar producto nuevo.

9. Desarrollo de Sprint 4

9.1. Tarea T19

Crear api para listar un producto nuevo.

9.1.1. Diseño

Crear api que permita listar los productos nuevo que ha generado el usuario logueado al aplicativo móvil.

Fuente: Elaboración propia

Figura 36: Pantalla que lista los productos nuevos ingresados

9.1.2. Código

```java
package com.sebasoft.tienda11.ui.fragment;

import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.esquema.Aplicacion;
import com.sebasoft.tienda11.esquema.Producto;
import com.sebasoft.tienda11.esquema.ProductoNuevo;
import com.sebasoft.tienda11.ui.controller.adapter_Articulo;
import com.sebasoft.tienda11.ui.controller.adapter_ProductoNuevo;
import com.sebasoft.tienda11.ui.controller.df_ProductoNuevo_Imagen;
import com.sebasoft.tienda11.ui.controller.df_ProductoNuevo_modificar;
import com.sebasoft.tienda11.ui.controller.df_articulo_info;

import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

public class fragment_ListarArticulo extends Fragment {
    private fragment_CrearArticulo.OnFragmentInteractionListener mListener;
    private ListView lv_producto;
    private Aplicacion app;
    private String Servidor;
    private int idtienda;
    private int iduser;
    private ArrayList<ProductoNuevo> asproductos;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_listararticulo, container, false);

        app = (Aplicacion) rootView.getContext().getApplicationContext();
        //.getApplicationContext()
        lv_producto = (ListView) rootView.findViewById(R.id.lv_listar_articulo_nuevo);
        Servidor = app.getServidor();
        idtienda = app.getIdtienda();
        iduser = app.getIduser();
        onListarProducto();

        lv_producto.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                ProductoNuevo productoNuevo = new ProductoNuevo();
                productoNuevo = (ProductoNuevo) adapterView.getItemAtPosition(i);

                Fragment fragment2 = new fragment_ProductoNuevo_modificar(productoNuevo);
                FragmentManager fragmentManager = getFragmentManager();
                FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.app_bar_main, fragment2);
                fragmentTransaction.commit();
                fragmentTransaction.addToBackStack(null);
                app.setFragmenadd();

```

```
            }
        });

        return rootView;
    }

    private void onListarProducto(){
        String tienda =Integer.toString(idtienda);
        String usuario =Integer.toString(iduser);
        String servicio = Servidor+"/"+"productos?tienda="+tienda+"&usuario="+usuario;

        RequestQueue queue = Volley.newRequestQueue(getContext());
        StringRequest postRequest = new StringRequest( Request.Method.GET, servicio,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        onCargarLista(new String(response));
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(getContext(),"Api sin respuestas",Toast.LENGTH_SHORT).show();
                    }
                });

        queue.add(postRequest);
    }

    public void onCargarLista(String response){
        ProductoNuevo producto;

        asproductos = new ArrayList<ProductoNuevo>();
        try {
            JSONArray jsonArray = new JSONArray(response);

            //ProductoNuevo(int id,String categoria,String subcategoria,String producto,String obse
            for(int i=0;i<jsonArray.length();i++){
                producto = new ProductoNuevo(
                        i,
                        jsonArray.getJSONObject(i).getString("categoria"),
                        jsonArray.getJSONObject(i).getString("subcategoria"),
                        jsonArray.getJSONObject(i).getString("id"),
                        jsonArray.getJSONObject(i).getString("catid"),
                        jsonArray.getJSONObject(i).getString("subcatid"),
                        jsonArray.getJSONObject(i).getString("dsc_prod"),
                        jsonArray.getJSONObject(i).getString("dsc_obs"),
                        jsonArray.getJSONObject(i).getString("dsc_modelo"),
                        jsonArray.getJSONObject(i).getJSONArray("imagenes")

                );
                asproductos.add(producto);
            }

            //creando el adapter personalizado
            adapter_ProductoNuevo adapter= new adapter_ProductoNuevo(getActivity(),asproductos);
            lv_producto.setAdapter(adapter);

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }



    // TODO: Rename method, update argument and hook method into UI event
    public void onButtonPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof fragment_CrearArticulo.OnFragmentInteractionListener) {
```

```
151             mListener = (fragment_CrearArticulo.OnFragmentInteractionListener) context;
152         } else {
153             throw new RuntimeException(context.toString()
154                     + " must implement OnFragmentInteractionListener");
155         }
156     }
157
158     @Override
159     public void onDetach() {
160         super.onDetach();
161         mListener = null;
162     }
163     public interface OnFragmentInteractionListener {
164         // TODO: Update argument type and name
165         void onFragmentInteraction(Uri uri);
166     }
167 }
```

Fuente: Elaboración propia

Figura 37: Código fuente de listar productos nuevos.

9.2. Tarea T20

Crear modelo de producto nuevo en el aplicativo.

9.2.1. Diseño

Se necesita crear el modelo del producto que recibirá la información del api.

9.2.2. Código

```
1  package com.sebasoft.tienda11.esquema;
2
3  import org.json.JSONArray;
4
5  public class ProductoNuevo {
6      private String categoria,subcategoria,producto,observacion,modelo,productoid;
7      private String categoriaid,subcategoriaid;
8      private JSONArray jimagenes;
9      private int id;
10
11     public ProductoNuevo(){
12
13     }
14     public ProductoNuevo(int id,String categoria,String subcategoria,String productoid,
15                          String categoriaid,String subcategoriaid,
16                          String producto,String observacion,String modelo,JSONArray jimagenes){
17         this.id = id;
18         this.categoria =categoria;
19         this.subcategoria = subcategoria;
20         this.producto = producto;
21         this.observacion = observacion;
22         this.modelo = modelo;
23         this.jimagenes = jimagenes;
24         this.productoid = productoid;
25         this.categoriaid = categoriaid;
26         this.subcategoriaid = subcategoriaid;
27     }
28
29     public String getCategoriaid() {
30         return categoriaid;
31     }
32
33     public String getSubcategoriaid() {
34         return subcategoriaid;
35     }
```

```
36
37      public void setCategoriaid(String categoriaid) {
38          this.categoriaid = categoriaid;
39      }
40
41      public void setSubcategoriaid(String subcategoriaid) {
42          this.subcategoriaid = subcategoriaid;
43      }
44
45      public void setProductoid(String productoid) {
46          this.productoid = productoid;
47      }
48
49      public String getProductoid() {
50          return productoid;
51      }
52
53      public int getId() {
54          return id;
55      }
56
57      public void setId(int id) {
58          this.id = id;
59      }
60
61      public String getCategoria() {
62          return categoria;
63      }
64
65      public void setCategoria(String categoria) {
66          this.categoria = categoria;
67      }
68
69      public void setSubcategoria(String subcategoria) {
70          this.subcategoria = subcategoria;
71      }
72
73      public void setProducto(String producto) {
74          this.producto = producto;
75      }
76
77      public void setObservacion(String observacion) {
78          this.observacion = observacion;
79      }
80
81      public String getSubcategoria() {
82          return subcategoria;
83      }
84
85      public String getProducto() {
86          return producto;
87      }
88
89      public String getObservacion() {
90          return observacion;
91      }
92
93      public JSONArray getJimagenes() {
94          return jimagenes;
95      }
96
97      public String getModelo() {
98          return modelo;
99      }
100
101     public void setJimagenes(JSONArray jimagenes) {
102         this.jimagenes = jimagenes;
103     }
104
105     public void setModelo(String modelo) {
106         this.modelo = modelo;
107     }
108 }
```

Fuente: Elaboración propia

Figura 38: Código fuente del modelo de producto nuevo.

9.3. Tarea T21

Crear pantalla de ingreso de producto nuevo en el aplicativo móvil

9.3.1. Diseño

Se necesita crear la pantalla de ingreso del producto nuevo.

Fuente: Elaboración propia

Figura 39: Pantalla de producto nuevo.

9.3.2. Código

```
1 package com.sebasoft.tienda11.ui.fragment;
2
3 import android.content.ClipData;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.graphics.Bitmap;
7 import android.graphics.BitmapFactory;
8 import android.net.Uri;
9 import android.os.Build;
10 import android.os.Bundle;
11 import android.os.Vibrator;
12 import android.util.Base64;
13 import android.view.LayoutInflater;
14 import android.view.View;
15 import android.view.ViewGroup;
16 import android.widget.AdapterView;
17 import android.widget.Button;
18 import android.widget.EditText;
19 import android.widget.GridView;
20 import android.widget.ProgressBar;
21 import android.widget.Spinner;
22 import android.widget.TabHost;
23 import android.widget.Toast;
24 import androidx.activity.result.ActivityResult;
25 import androidx.activity.result.ActivityResultCallback;
26 import androidx.activity.result.ActivityResultLauncher;
27 import androidx.activity.result.contract.ActivityResultContracts;
28 import androidx.annotation.NonNull;
```

```java
import androidx.annotation.RequiresApi;
import androidx.fragment.app.Fragment;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.api.apiConexion;
import com.sebasoft.tienda11.ui.controller.GridViewAdapter;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class fragment_CrearArticulo extends Fragment implements  AdapterView.OnItemSelectedListener{
    private OnFragmentInteractionListener mListener;
    Spinner sp_categoria,sp_subcategoria;
    EditText et_producto,et_modelo,et_observacion;
    ProgressBar pb_guardar;
    apiConexion apiconexion;
    String  ls_categoria,ls_subcategoria;
    private Uri imagenUri = null;
    private Button btnGaleria;
    private GridView gvImagenes;
    List<Uri> listaImagenes = new ArrayList<>();
    private String Servidor;
    GridViewAdapter baseAdapter;
    private boolean lbo_enviando = true;
    List<String> listaBase64Imagenes = new ArrayList<>();

    @RequiresApi(api = Build.VERSION_CODES.M)
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_creararticulo, container, false);
        TabHost tabs = (TabHost) rootView.findViewById(R.id.tabhost);
        tabs.setup();
        setHasOptionsMenu(true);
        TabHost.TabSpec spec = tabs.newTabSpec("tab1");
        spec.setContent(R.id.tab1);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_input_get));
        tabs.addTab(spec);
        spec = tabs.newTabSpec("tab2");
        spec.setContent(R.id.tab2);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_gallery));
        tabs.addTab(spec);
        spec = tabs.newTabSpec("tab3");
        spec.setContent(R.id.tab3);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_manage));
        tabs.addTab(spec);
        tabs.setCurrentTab(0);

        sp_categoria = (Spinner) rootView.findViewById(R.id.sp_categoria);
        sp_subcategoria = (Spinner) rootView.findViewById(R.id.sp_subcategoria);
        et_producto = (EditText) rootView.findViewById(R.id.et_producto);
        et_modelo = (EditText) rootView.findViewById(R.id.et_modelo);
        et_observacion = (EditText) rootView.findViewById(R.id.et_observacion);
        pb_guardar = (ProgressBar) rootView.findViewById(R.id.pb_guardar);
        btnGaleria = (Button) rootView.findViewById(R.id.btnGaleria);
        gvImagenes = (GridView) rootView.findViewById(R.id.gvImagenes);
        sp_categoria.setOnItemSelectedListener(this);
        sp_subcategoria.setOnItemSelectedListener(this);

        Vibrator vibrator = (Vibrator) getActivity().getSystemService(getContext().VIBRATOR_SERVICE);

        /*"LLENAMOS LOS DATOS DEL SPINER CON LA CLASE DE APIPRODUCTOS"*/
        apiconexion = new apiConexion(getContext());
        apiconexion.setSpinners("categoria",sp_categoria,"");
        Servidor = apiconexion.getServidor();

        FloatingActionButton fab = (FloatingActionButton) rootView.findViewById(R.id.fab_producto);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String servicio;
                Map<String, String> data = new HashMap<>();

                if (et_producto.getText().toString().length() == 0) {
                    vibrator.vibrate(400);
                    Toast.makeText(getContext(),"Ingrese una descripción general del producto",Toast.LENGTH_SHORT).show();
                } else {

                    data.put("idtienda", Integer.toString(apiconexion.getIdtienda()));
                    data.put("categoria", ls_categoria);
                    data.put("subcategoria", ls_subcategoria);
                    data.put("producto",et_producto.getText().toString());
                    data.put("observacion", et_observacion.getText().toString());
                    data.put("modelo", et_modelo.getText().toString());
                    data.put("idusuario", Integer.toString(apiconexion.getIduser()));
                    data.put("token", apiconexion.getToken());
                    JSONObject jsonData = new JSONObject(data);
                    //result = apiconexion.onEnviarPOST(jsonData, "productos");

                    servicio = Servidor+"/productos";

```

```
                /*Insertamos datos del producto*/
                habilitaProducto(false);
                RequestQueue queue = Volley.newRequestQueue(getContext());
                JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.POST, servicio,
                        jsonData,
                        new Response.Listener<JSONObject>() {
                            @Override
                            public void onResponse(JSONObject response) {
                                JSONObject Obj,Result;
                                Obj = response;
                                String  status,mensaje,sidcat,sidsubcat,sidpro;
                                try {
                                    status = Obj.getString("status");
                                    Result = Obj.getJSONObject("result");
                                    if (status.equals("ok")){
                                        //-----------------------------------------OK---------------------------------

                                        sidcat = Result.getString("idcat");
                                        sidsubcat = Result.getString("idsubcat");
                                        sidpro = Result.getString("idpro");
                                        /*Enviamos las Imágenes*/
                                        if (baseAdapter != null){
                                            subirImagenes(sidcat,sidsubcat,sidpro);
                                            baseAdapter.setclean();
                                            gvImagenes.setAdapter(null);
                                        }

                                        et_producto.setText("");
                                        et_modelo.setText("");
                                        et_observacion.setText("");
                                        habilitaProducto(true);
                                        rootView.findViewById(R.id.et_producto).requestFocus();
                                        tabs.setCurrentTab(0);

                                        Toast.makeText(getContext(),"Datos Guardados correctamente",Toast.LENGTH_LONG).show();
                                    }else{
                                        vibrator.vibrate(400);
                                        habilitaProducto(true);
                                        mensaje = Result.getString("error_message");
                                        Toast.makeText(getContext(),mensaje,Toast.LENGTH_LONG).show();
                                    }
                                } catch (JSONException e) {
                                    //El servio no trajo nada
                                    vibrator.vibrate(400);
                                    habilitaProducto(true);
                                    e.printStackTrace();
                                    Toast.makeText(getContext(),"El servicio retorno un error inesperado",Toast.LENGTH_LONG).show();
                                }
                            }
                        },
                        new Response.ErrorListener() {
                            @Override
                            public void onErrorResponse(VolleyError error) {
                                vibrator.vibrate(400);
                                habilitaProducto(true);
                                Toast.makeText(getContext(),"El servicio no está disponible",Toast.LENGTH_LONG).show();
                                error.printStackTrace();
                            }
                        });
                queue.add(postRequest);
            }
        }
    });

    btnGaleria.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            pickImage();

        }
    });

    return rootView;
}

public void setflag_enviand(boolean flag){
    this.lbo_enviando = flag;
}

public boolean getflag_enviando(){
    return this.lbo_enviando;
}

public void subirImagenes(String idcat,String idsubcat,String idpro){
    listaBase64Imagenes.clear();
    for(int i=0;i<listaImagenes.size(); i++){
        try {
            InputStream is = getActivity().getContentResolver().openInputStream(listaImagenes.get(i));
            Bitmap bitmap = BitmapFactory.decodeStream(is);
            String cadena = convertirUriToBase64(bitmap);
            enviarImagen(cadena,idcat,idsubcat,idpro);
            bitmap.recycle();
            if(getflag_enviando()){
                Thread.sleep(500); //Esperamos medio segundo para elviar el siguiente
            }
        }catch (Exception ex){
            ex.getMessage();
        }
    }
}
```

```
231     public void enviarImagen(final String cadena,final String idcat,final String idsubcat,final String idpro){
232         Map<String, String> data = new HashMap<>();
233         setflag_enviand(true);
234         data.put("idtienda", Integer.toString(apiconexion.getIdtienda()));
235         data.put("imagen", cadena);
236         data.put("idcat",idcat);
237         data.put("idsubcat",idsubcat);
238         data.put("idpro",idpro);
239         data.put("iduser",Integer.toString(apiconexion.getIduser()));
240         data.put("token", apiconexion.getToken());
241         JSONObject jsonData = new JSONObject(data);
242         String servicio = Servidor+"/subir_imagen";
243         RequestQueue queue = Volley.newRequestQueue(getContext());
244         JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.POST, servicio,
245                 jsonData,
246                 new Response.Listener<JSONObject>() {
247                     @Override
248                     public void onResponse(JSONObject response) {
249                         JSONObject Obj,Result;
250                         Obj = response;
251                         String  status,mensaje;
252                         try {
253                             status = Obj.getString("status");
254                             Result = Obj.getJSONObject("result");
255                             if (status.equals("ok")){
256                                 //=====================================OK=================================
257                                 setflag_enviand(false);
258                             }else{
259                                 //vibrator.vibrate(400);
260                                 //habilitaProducto(true);
261                                 mensaje = Result.getString("error_message");
262                                 showToast(mensaje);
263                             }
264                         } catch (JSONException e) {
265                             //El servio no trajo nada
266                             //vibrator.vibrate(400);
267                             //habilitaProducto(true);
268                             e.printStackTrace();
269                             Toast.makeText(getContext(),"El servicio de subida de imagenes retorno un error inesperado",Toast.LENGTH_LO
270                         }
271                     }
272                 },
273                 new Response.ErrorListener() {
274                     @Override
275                     public void onErrorResponse(VolleyError error) {
276                         //vibrator.vibrate(400);
277                         //habilitaProducto(true);
278                         Toast.makeText(getContext(),"El servicio no está disponible",Toast.LENGTH_LONG).show();
279                         error.printStackTrace();
280                     }
281                 }));
282         queue.add(postRequest);
283
284     }
285     public String convertirUriToBase64(Bitmap bitmap){
286         ByteArrayOutputStream baos = new ByteArrayOutputStream();
287         bitmap.compress(Bitmap.CompressFormat.JPEG,100,baos);
288         byte[] bytes = baos.toByteArray();
289         String encode = Base64.encodeToString(bytes,Base64.DEFAULT);
290
291         return encode;
292     }
293     private void pickImage(){
294         Intent intent = new Intent(Intent.ACTION_PICK);
295         intent.setType("image/*");
296         intent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE,true);
297         galleryActivityResultLauncher.launch(intent);
298     }
299
300     private ActivityResultLauncher<Intent> galleryActivityResultLauncher = registerForActivityResult(
301             new ActivityResultContracts.StartActivityForResult(),
302             new ActivityResultCallback<ActivityResult>() {
303                 @Override
304                 public void onActivityResult(ActivityResult result) {
305                     try {
306                         if(result.getResultCode() == getActivity().RESULT_OK){
307
308                             Intent data = result.getData();
309                             ClipData clipData = data.getClipData();
310                             if(clipData == null){
311                                 // Para cuando solo elige una imagen
312                                 imagenUri = data.getData();
313                                 listaImagenes.add(imagenUri);
314                             }else{
315                                 //mas de una imagen
316                                 for (int i = 0;i < clipData.getItemCount(); i++){
317                                     listaImagenes.add(clipData.getItemAt(i).getUri());
318                                 }
319                             }
320                             baseAdapter = new GridViewAdapter(getContext(),listaImagenes);
321                             gvImagenes.setAdapter(baseAdapter);
322
323                         }else{
324                             //cancelado
325                             showToast("Selección Cancelada");
326                         }
327                     }catch (Exception ex){
328                         ex.getMessage();
329                     }
330
331                 }
332             }
```

```
    });

    private void showToast(String message){
        Toast.makeText(getContext(),message,Toast.LENGTH_SHORT).show();
    }

    private void habilitaProducto(boolean abo){
        if (abo){
            pb_guardar.setVisibility(View.INVISIBLE);
            et_producto.setEnabled(abo);
            et_modelo.setEnabled(abo);
            et_observacion.setEnabled(abo);
        }else{
            pb_guardar.setVisibility(View.VISIBLE);
            et_producto.setEnabled(abo);
            et_modelo.setEnabled(abo);
            et_observacion.setEnabled(abo);

        }
    }

    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
        String item ;
        item = adapterView.getItemAtPosition(i).toString();
        switch (adapterView.getId()){
            case R.id.sp_categoria:
                ls_categoria = item;
                apiconexion.setSpinners("subcategoria",sp_subcategoria,"&categoria="+item);
                break;
            case R.id.sp_subcategoria:
                ls_subcategoria = item;
                break;
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {

    }

    // TODO: Rename method, update argument and hook method into UI event
    public void onButtonPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnFragmentInteractionListener) {
            mListener = (OnFragmentInteractionListener) context;
        } else {
            throw new RuntimeException(context.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }

    public interface OnFragmentInteractionListener {
        // TODO: Update argument type and name
        void onFragmentInteraction(Uri uri);
    }

}
```

Fuente: Elaboración propia

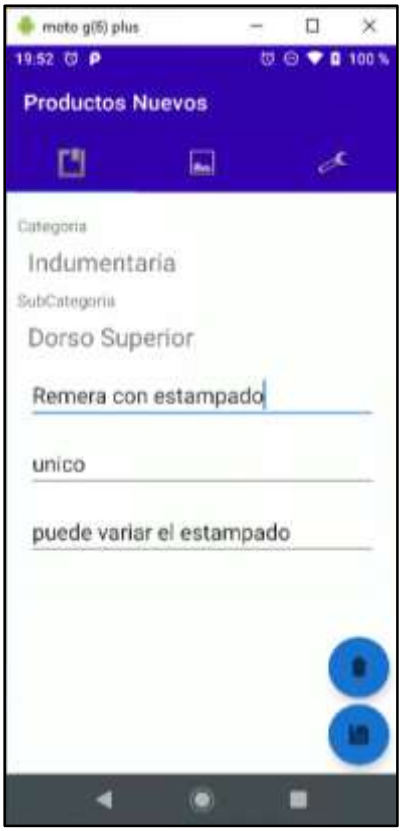
Figura 40: Código fuente de ingresar producto nuevo.

9.4. Tarea T22

Crear pantalla de modificación o eliminar producto nuevo con el listado.

9.4.1. Diseño

Se necesita crear una pantalla que liste los productos nuevos que ha registrado el usuario logueado al aplicativo móvil.



Fuente: Elaboración propia

Figura 41: Pantalla de modificar o eliminar producto nuevo.

9.4.2. Código

```
1 package com.sebasoft.tienda11.ui.fragment;
2
3 import android.content.DialogInterface;
4 import android.os.Bundle;
5
6 import androidx.appcompat.app.AlertDialog;
7 import androidx.fragment.app.Fragment;
8
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.AdapterView;
13 import android.widget.Button;
14 import android.widget.EditText;
15 import android.widget.GridView;
16 import android.widget.ProgressBar;
17 import android.widget.Spinner;
18 import android.widget.TabHost;
19 import android.widget.TextView;
20 import android.widget.Toast;
21
22 import com.android.volley.Request;
23 import com.android.volley.RequestQueue;
24 import com.android.volley.Response;
25 import com.android.volley.VolleyError;
26 import com.android.volley.toolbox.JsonObjectRequest;
27 import com.android.volley.toolbox.Volley;
28 import com.google.android.material.floatingactionbutton.FloatingActionButton;
29 import com.google.android.material.snackbar.Snackbar;
30 import com.sebasoft.tienda11.R;
31 import com.sebasoft.tienda11.api.apiConexion;
32 import com.sebasoft.tienda11.esquema.Aplicacion;
33 import com.sebasoft.tienda11.esquema.ProductoNuevo;
34 import com.sebasoft.tienda11.ui.controller.adapter_gvImagen;
35
36 import org.json.JSONArray;
37 import org.json.JSONException;
38 import org.json.JSONObject;
39
40 import java.util.ArrayList;
```

```java
import java.util.HashMap;
import java.util.Map;

public class fragment_ProductoNuevo_modificar extends Fragment {

    private String Servidor,userid,tiendaId,token;
    private Aplicacion app;
    private JSONArray jimagen;
    private TextView tv_cat,tv_subcat;
    private EditText et_producto,et_modelo,et_observacion;
    private ProgressBar pb_guardar;
    private GridView gvImagenes;
    private ProductoNuevo productoNuevo;
    Fragment fragment;

    public fragment_ProductoNuevo_modificar(ProductoNuevo productoNuevo){
        this.productoNuevo = productoNuevo;
    }

    public static fragment_ProductoNuevo_modificar newInstance(ProductoNuevo productoNuevo) {
        fragment_ProductoNuevo_modificar fragment = new fragment_ProductoNuevo_modificar(productoNuevo);
        Bundle args = new Bundle();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment__producto_nuevo_modificar, container, false);
        String url_imagen,title_imagen;
        fragment = this;

        app = (Aplicacion) view.getContext().getApplicationContext();
        Servidor = app.getServidor();
        userid = Integer.toString(app.getIduser());
        tiendaId = Integer.toString(app.getIdtienda());
        token = app.getToken();
        jimagen = productoNuevo.getJimagenes();

        TabHost tabs = (TabHost) view.findViewById(R.id.tabhostnuevo_fragment);
        tabs.setup();
        setHasOptionsMenu(true);
        TabHost.TabSpec spec = tabs.newTabSpec("tab1?");
        spec.setContent(R.id.tab1);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_input_get));
        tabs.addTab(spec);
        spec = tabs.newTabSpec("tab2?");
        spec.setContent(R.id.tab2);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_gallery));
        tabs.addTab(spec);
        spec = tabs.newTabSpec("tab3?");
        spec.setContent(R.id.tab3);
        spec.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_manage));
        tabs.addTab(spec);
        tabs.setCurrentTab(0);

        tv_cat = (TextView) view.findViewById(R.id.tv_cat2);
        tv_subcat = (TextView) view.findViewById(R.id.tv_subcat2);
        et_producto = (EditText) view.findViewById(R.id.et_producto2);
        et_modelo = (EditText) view.findViewById(R.id.et_modelo2);
        et_observacion = (EditText) view.findViewById(R.id.et_observacion2);
        pb_guardar = (ProgressBar) view.findViewById(R.id.pb_guardar2);
        gvImagenes = (GridView) view.findViewById(R.id.gvImagenes2);

        title_imagen = productoNuevo.getProducto();
        et_producto.setText(title_imagen);
        et_modelo.setText(productoNuevo.getModelo());
        et_observacion.setText(productoNuevo.getObservacion());
        tv_cat.setText(productoNuevo.getCategoria());
        tv_subcat.setText(productoNuevo.getSubcategoria());

        ArrayList<String> listaImagen = new ArrayList<String>();
        for (int i = 0; i < jimagen.length(); i++) {
            try {

                url_imagen = Servidor+jimagen.getJSONObject(i).getString("dsc_dir");
                listaImagen.add(url_imagen);

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        adapter_gvImagen aimagen = new adapter_gvImagen(getContext(),listaImagen);
        gvImagenes.setAdapter(aimagen);

        /*Eventos Click*/
        FloatingActionButton fab_guardar = (FloatingActionButton) view.findViewById(R.id.fab_prod_guardar2);
        fab_guardar.setOnClickListener(new View.OnClickListener() {
```

```
136             @Override
137             public void onClick(View view) {
138                 //Modificar
139                 AlertDialog.Builder dialogo = new AlertDialog.Builder(view.getContext());
140                 dialogo.setTitle("Modificar Producto Nuevo");
141                 dialogo.setMessage("¿Desea Modificar "+productoNuevo.getProducto()+"?");
142                 dialogo.setCancelable(false);
143 
144                 dialogo.setPositiveButton("Guardar",new DialogInterface.OnClickListener(){
145                     public void onClick(DialogInterface dialogo, int id){
146                         pb_guardar.setVisibility(getView().VISIBLE);
147                         try {
148                             onActualizarProducto();
149                         } catch (JSONException e) {
150                             e.printStackTrace();
151                             pb_guardar.setVisibility(getView().INVISIBLE);
152                             Toast.makeText(getContext(),"No se pudo Actualizar el producto",Toast.LENGTH_SHORT).show();
153                         }
154                     }
155 
156                 });
157                 dialogo.setNegativeButton("Cancelar",new DialogInterface.OnClickListener(){
158                     public void onClick(DialogInterface dialogo, int id){
159                         // TODO por ahora no hacemos nada...
160                     }
161 
162                 });
163                 dialogo.show();
164             }
165         });
166         FloatingActionButton fab_eliminar = (FloatingActionButton) view.findViewById(R.id.fab_prod_eliminar2);
167         fab_eliminar.setOnClickListener(new View.OnClickListener() {
168             @Override
169             public void onClick(View view) {
170                 // Eliminar
171                 AlertDialog.Builder dialogo = new AlertDialog.Builder(view.getContext());
172                 dialogo.setTitle("Eliminar Producto Nuevo");
173                 dialogo.setMessage("¿Desea Eliminar "+productoNuevo.getProducto()+"?");
174                 dialogo.setCancelable(false);
175 
176                 dialogo.setPositiveButton("Eliminar",new DialogInterface.OnClickListener(){
177                     public void onClick(DialogInterface dialogo, int id){
178                         pb_guardar.setVisibility(getView().VISIBLE);
179                         onEliminarProducto();
180 
181                     }
182 
183                 });
184                 dialogo.setNegativeButton("Cancelar",new DialogInterface.OnClickListener(){
185                     public void onClick(DialogInterface dialogo, int id){
186                         // TODO por ahora no hacemos nada...
187                     }
188 
189                 });
190                 dialogo.show();
191             }
192         });
193 
194         return view;
195     }
196     private void onActualizarProducto() throws JSONException {
197         /*Armamos el Json*/
198         Map<String, String> data = new HashMap<>();
199 
200         data.put("idtienda", tiendaId);
201         data.put("idusuario", userid);
202         data.put("idproducto", productoNuevo.getProductoId());
203         data.put("token", token);
204 
205         data.put("idcategoria", productoNuevo.getCategoriaId());
206         data.put("idsubcategoria", productoNuevo.getSubcategoriaId());
207 
208         data.put("producto", et_producto.getText().toString());
209         data.put("modelo", et_modelo.getText().toString());
210         data.put("observacion", et_observacion.getText().toString());
211 
212         JSONObject jdata2 = new JSONObject(data);
213         //TODO esto pesado de codigo nos sirve para armar json anidado
214         /*
215         JSONObject jexample = new JSONObject(data);
216         jexample.put("imagenes",productoNuevo.getImagenes());
217         System.out.println(jexample);
218         */
219 
220         String servicio = Servidor+"/productos";
221         RequestQueue queue = Volley.newRequestQueue(getContext());
222         JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.PUT, servicio,
223                 jdata2,
224                 new Response.Listener<JSONObject>() {
225                     @Override
226                     public void onResponse(JSONObject response) {
227                         JSONObject Obj,Result;
228                         Obj = response;
229                         String  status,mensaje;
230                         try {
```

```
                        status = Obj.getString("status");
                        Result = Obj.getJSONObject("result");
                        if (status.equals("ok")){
                            Toast.makeText(getContext(),"Producto Modificado",Toast.LENGTH_LONG).show();
                            pb_guardar.setVisibility(getView().INVISIBLE);
                            getActivity().onBackPressed();

                        }else{
                            //vibrator.vibrate(400);
                            mensaje = Result.getString("error_message");
                            Toast.makeText(getContext(),mensaje,Toast.LENGTH_LONG).show();
                            //getFragmentManager().beginTransaction().remove(fragment).commit();
                        }
                    } catch (JSONException e) {
                        //El servio no trajo nada
                        e.printStackTrace();
                        Toast.makeText(getContext(),"El servicio retorno un error inesperado",Toast.LENGTH_LONG).show();
                    }
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(getContext(),"El servicio no está disponible",Toast.LENGTH_LONG).show();
                    error.printStackTrace();
                }
            });
        queue.add(postRequest);
    }

    private void onEliminarProducto(){
        /*Armamos el Json*/
        Map<String, String> data = new HashMap<>();

        data.put("idtienda", tiendaid);
        data.put("idusuario", userid);
        data.put("idproducto", productoNuevo.getProductoId());
        data.put("idcategoria", productoNuevo.getCategoriaId());
        data.put("idsubcategoria", productoNuevo.getSubcategoriaId());
        data.put("token", token);



        JSONObject jdata2 = new JSONObject(data);
        String servicio = Servidor+"/producto_eliminar";

        //System.out.println(jdata2);
        RequestQueue queue = Volley.newRequestQueue(getContext());
        JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.POST, servicio,
                jdata2,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {
                        JSONObject Obj,Result;
                        Obj = response;
                        String  status,mensaje;
                        try {
                            status = Obj.getString("status");
                            Result = Obj.getJSONObject("result");
                            if (status.equals("ok")){
                                Toast.makeText(getContext(),"Producto Eliminado.",Toast.LENGTH_LONG).show();
                                pb_guardar.setVisibility(getView().INVISIBLE);
                                getActivity().onBackPressed();

                            }else{
                                //vibrator.vibrate(400);
                                mensaje = Result.getString("error_message");
                                Toast.makeText(getContext(),mensaje,Toast.LENGTH_LONG).show();
                                //getFragmentManager().beginTransaction().remove(fragment).commit();
                            }
                        } catch (JSONException e) {
                            //El servio no trajo nada
                            e.printStackTrace();
                            Toast.makeText(getContext(),"El servicio retorno un error inesperado",Toast.LENGTH_LONG).show();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(getContext(),"El servicio no está disponible",Toast.LENGTH_LONG).show();
                        error.printStackTrace();
                    }
                });
        queue.add(postRequest);
    }
}
```

Fuente: Elaboración propia

Figura 42: Código fuente de la pantalla de modificación o eliminar producto nuevo.

9.5. Tarea T23

Crear proceso de aprobación y asociación del producto nuevo con el listado de productos.

9.5.1. Diseño

Crear el proceso api y base de datos que permita pasar los productos e imágenes temporales a las tablas del sistema.

9.5.2. Código

upimagenes.php

```php
<?php

require_once 'clases/respuestas.class.php';
$_respuestas = new respuestas;


header("Content-Type: application/json");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $datos = json_decode(file_get_contents('php://input'),true);
    if(!isset($datos['imagenes']) && !isset($datos['nom'])){
      echo json_encode($_respuestas->error_400());
    }else{
      $base64 = $datos['imagenes'];
      $nom_archivo = $datos['nom'];

      //$base_to_php = explode( ',', $base64);
      //$data = base64_decode($base_to_php[1]);
      $data = base64_decode($base64);
      $filepath = "Imagenes/".$nom_archivo.".jpeg";
      file_put_contents($filepath, $data);
      chmod ($filepath, 0644);

      $result = $_respuestas->response;
      $result['status'] = "ok";
      $result['result'] = array(
          "mensaje" => "Datos guardados correctamente"
      );
      echo json_encode($result);
    }
}else{
    echo json_encode($_respuestas->error_405());
}
```

Fuente: Elaboración propia

Figura 43: Código fuente del proceso api que permite pasar los productos nuevos.

```sql
CREATE DEFINER=`vadmin`@`%` PROCEDURE `sp_crear_producto`(
    in an_idtienda int,
    in an_idcat int,
    in an_idsubcat int,
    in an_id int
)
BEGIN
    declare coderror,imgid,imguserid int;
    declare msjerror varchar(1000);
    declare imgdir varchar(4000);
    declare imgfecha datetime;
    DECLARE finished INTEGER DEFAULT 0;
    DEClARE curImagen CURSOR FOR
        select  a.id,a.dsc_dir,a.userid,a.fecha
        from    imagenes_temp a
        where   a.catid = an_idcat
                and a.subcatid = an_idsubcat
                and a.prodid = an_id;

```

```
    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET finished = 1;

    declare exit handler for sqlexception
    begin
        GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
        @errno = MYSQL_ERRNO, @text = MESSAGE_TEXT;
        rollback;
        set coderror = ifnull(ifnull(coderror,@errno),'-1');
        set msjerror = ifnull(ifnull(msjerror,@text),'Error Inesperado');
        insert into log_errores(objeto,cod_error,mensaje,fecha)
        values('sp_crear_producto',coderror,msjerror,now());
        commit;
        select 'error' status,msjerror message from dual;
    end;


    if (select exists(select 1
                        from producto_temp a where  a.tiendaid = an_idtienda
                                                    and a.catid = an_idcat
                                                    and a.subcatid = an_idsubcat
                                                    and a.id = an_id
                                                    and estadoid = 1)) then
        /*seteamos las variables*/
        select  a.dsc_prod,if(length(trim(a.dsc_obs))=0,null,a.dsc_obs),
        if(length(trim(a.dsc_modelo))=0,null,a.dsc_modelo),a.userid,a.fecha
        into    @producto,@observacion,@modelo,@usuario,@fecha
        from    producto_temp a
        where   a.tiendaid = an_idtienda
                and a.catid = an_idcat
                and a.subcatid = an_idsubcat
                and a.id = an_id
                and estadoid = 1;

        select  ifnull(max(codigo),0) + 1
        into    @codigo
        from    productos
        where   tienda_codigo = an_idtienda
                and categorias_codigo = an_idcat
                and subcategoria_codigo = an_idsubcat;

        set @dsccodigo = concat(cast(an_idcat as char),cast(an_idsubcat as char),
        cast(@codigo as char));

        /*Iniciamos la transacción, para insertar el producto nuevo*/
        start transaction;
        insert into producto_vinculo_nuevo(tienda_codigo,categorias_codigo,
        subcategoria_codigo,producto_codigo,producto_temp_codigo,usuario_codigo,fecha_temp)
        values(an_idtienda,an_idcat,an_idsubcat,@codigo,an_id,@usuario,@fecha);

        insert into productos(tienda_codigo,categorias_codigo,subcategoria_codigo,
        codigo,nombre,observacion,modelo,estado_codigo)
        values(an_idtienda,an_idcat,an_idsubcat,@codigo,@producto,@observacion,@modelo,1);

        /*Iniciamos el traslado de las imagenes*/
        OPEN curImagen;

        getImagen: LOOP
            FETCH curImagen INTO imgid,imgdir,imguserid,imgfecha;
            IF finished = 1 THEN
                LEAVE getImagen;
            END IF;

            /*Generamos el codigo de la imagen*/
            select  ifnull(max(codigo),0) + 1
            into    @imgcodigo
            from    imagenes
            where   categorias_codigo = an_idcat
                    and subcategoria_codigo = an_idsubcat
                    and productos_codigo = @codigo;

            /*Insertamos las Imagenes*/
            insert into imagenes(categorias_codigo,subcategoria_codigo,productos_codigo,
            codigo,direccion,estado_codigo)
            values(an_idcat,an_idsubcat,@codigo,@imgcodigo,imgdir,1);

            insert into imagenes_vinculo_nuevo(categorias_codigo,subcategoria_codigo,
            producto_codigo,imagen_codigo,
                                                imagen_temp_codigo,usuario_codigo,fecha_temp)
            values(an_idcat,an_idsubcat,@codigo,@imgcodigo,imgid,imguserid,now());

```

```
101
102         END LOOP getImagen;
103         CLOSE curImagen;
104
105         /*Limpiamos las Tablas temporales*/
106         delete from producto_temp
107         where   tiendaid = an_idtienda
108                 and catid = an_idcat
109                 and subcatid = an_idsubcat
110                 and id = an_id;
111
112         delete from imagenes_temp
113         where   catid = an_idcat
114                 and subcatid = an_idsubcat
115                 and prodid = an_id;
116
117         commit;
118         select 'ok' status,concat('Se Generó el código: ',@dsccodigo) message from dual;
119     else
120         set coderror = 45001;
121         set msjerror = 'Producto temporal inválido, asegurase de que existe este producto.';
122         signal sqlstate '45001' set message_text = msjerror;
123     end if;
124 END
```

Fuente: Elaboración propia

Figura 44: Código fuente de procedimiento almacenado de aprobar nuevos producto.

10. Desarrollo de Sprint 5

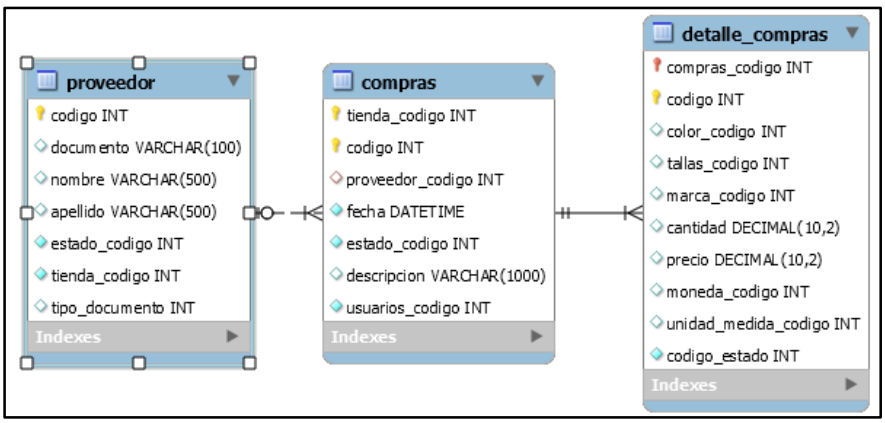
10.1. Tarea T24

Crear esquema de orden de compra.

10.1.1. Diseño

Se necesita crear el esquema de la orden de compra, esto quiere decir los nuevos campos que permiten manejar los estados desde el api.

10.1.2. Código



Fuente: Elaboración propia

Figura 45: Esquema de orden de compra.

10.2. Tarea T25

Crear función de verificación del token.

10.2.1. Diseño

Se necesita crear la función de verificación del token que recibe del aplicativo móvil cuando desea hacer modificaciones a las órdenes de compra.

10.2.2. Código

```
297     private function getToken($token){
298         $query = "  select  id,userid,estado
299                     from    usuario_token
300                     where token = :token
301                     and estado = 1
302                     and now() between fecha and date_add(fecha, interval 4 hour)";
303         $params = [
304             'token' => $token
305         ];
306         $resp = parent::getParamDatos($query,$params);
307         if ($resp){
308             return $resp;
309         }else{
310             return 0;
311         }
312     }
313
314     private function extenderToken($tokenId){
315         $query = "update usuario_token
316                     set fecha = now()
317                     where id = :id";
318         $params = ['id' => $tokenId];
319         $resp = parent::noResultQuery($query,$params);
320         if ($resp) {
321             return $resp;
322         }else{
323             return 0;
324         }
325     }
326
327 }
```

Fuente: Elaboración propia

Figura 46: Código fuente del api de órdenes de compra, extracción de la función de validación.

10.3. Tarea T26

Crear api para listar las órdenes de compra asignadas.

10.3.1. Diseño

Se necesita crear el api que liste las órdenes de compra, estos datos tienen que estar asociados al usuario de logueo del aplicativo móvil.

10.3.2. Código

orden_compra.php

```php
<?php
require_once 'clases/respuestas.class.php';
require_once 'clases/orden_compra.class.php';

$_respuesta = new respuestas;
$_orden_compra = new orden_compra;


header("Content-Type: application/json");
switch($_SERVER['REQUEST_METHOD']){
    /*=============================================================================================
    case 'POST':
        $postBody = file_get_contents('php://input');
        $resp = $_orden_compra->sendOrdenCompra($postBody);
        echo json_encode($resp);
        break;
    /*=============================================================================================
    case 'GET':
        if (isset($_GET['tienda']) && isset($_GET['usuario'])){
            $resultado = $_orden_compra->getOrdenCompra($_GET['tienda'],$_GET['usuario']);
            echo json_encode($resultado);

        }else{
            echo json_encode($_respuesta->error_400());
        }

        break;
}

?>
```

Elaboración propia

Figura 47: Código fuente del api que recibe peticiones para las órdenes de compra.

```php
    public function getOrdenCompra($tienda,$usuario){

        $query = "select
                        a.codigo,b.proveedor,date_format(a.fecha,'%d/%m/%Y') fecha,a.descripcion
                    from
                        compras a left join
                        vproveedor b on b.codigo = ifnull(a.proveedor_codigo,0)
                    where
                        a.tienda_codigo = :tienda
                        and a.usuarios_codigo = :usuario
                        and a.estado_codigo = 1
                        and exists (select 1 from detalle_compras x
                                    where x.compras_codigo = a.codigo
                                    and x.codigo_estado = 1)
                    order by a.fecha desc";
        $params = [
            'tienda' => $tienda,
            'usuario' => $usuario
        ];
        $sql = parent::getParamDatos($query,$params);

        $cab = array();
        foreach ($sql as $columna => $valor) {
            $det = array();
            foreach($valor as $c => $v){
                $det += [$c => $v];
                if($c == "codigo"){
```

```
                    $compra = $v;
                }
            }
            $det += ["marcas" => $this->getDetalleOrdenCompra_producto($compra)];
            $cab += [$columna => $det];
        }

        return $cab;
    }
```

Elaboración propia

Figura 48: Código fuente del api con sus procesos para el listado de órdenes de compra.

10.4. Tarea T27

Crear api para modificar las cantidades de los productos según su marca, color y talla de las órdenes de compra.

10.4.1. Diseño

Se necesita crear el api que permite modificar las cantidades que trae las órdenes de compra según el producto, la marca y la talla, esto con la finalidad de que la orden sea fácil de manejar cuando se está comprando.

10.4.2. Código

clases > orden_compra_editar.class.php

```php
<?php
require_once 'conexion/conexion.php';-
require_once 'respuestas.class.php';

class orden_compra_editar extends conexion{
    private $compra = "";
    private $detalle = "";
    private $cantidad = "";

    public function onModificar($json){
        $_respuesta = new respuestas;
        $datos = json_decode($json,true);
        //============================================================================INI-TOKEN
        if(!isset($datos['token'])){
            return $_respuesta->error_401();
        }else {
            $tokenarray = $this->getToken($datos['token']);
            if(!$tokenarray){
                return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
            }
        }
        //============================================================================FIN-TOKEN
        if(isset($datos['tipo'])){
            /** Elegimos que tipo de modificación hacemos*/

            switch ($datos['tipo']) {
                case '1':
                    return $this->onActualizarCantidad($datos);
                    break;
            }
        }else{
```

```
            return $_respuesta->error_401("Falta el tipo de formulario a cambiar");
        }
    }

    public function onActualizarCantidad($datos){
        $_respuesta = new respuestas;
        if (!isset($datos['compra']) || !isset($datos['detalle']) || !isset($datos['cantidad'])){
            return $_respuesta-> error_400();
        }else{
            $this->compra       = $datos['compra'];
            $this->detalle      = $datos['detalle'];
            $this->cantidad     = $datos['cantidad'];

            //Actualizamos la Orden de Compra como anulado / eliminado
            $query = "  update  detalle_compras
                        set     cantidad = :cantidad
                        where   compras_codigo = :compra
                                and codigo = :detalle";
            $params = [
                'compra'        => $this->compra,
                'detalle'       => $this->detalle,
                'cantidad'      => $this->cantidad
            ];
            $resulset = parent::noResultQuery($query,$params);
            if ($resulset){
                $result = $_respuesta->response;
                $result['status'] = "ok";
                $result['result'] = array(
                    "mensaje" => "Datos guardados correctamente"
                );
                return $result;
            }else{
                return $_respuesta->error500("Ocurrió un problema enviando datos al servicio.");
            }
        }
    }

    private function getToken($token){
        $query = "  select  id,userid,estado
                    from    usuario_token
                    where token = :token
                    and estado = 1
                    and now() between fecha and date_add(fecha, interval 4 hour)";
        $params = [
            'token' => $token
        ];
        $resp = parent::getParamDatos($query,$params);
        if ($resp){
            return $resp;
        }else{
            return 0;
        }
    }

    private function extenderToken($tokenid){
        $query = "update usuario_token
                    set fecha = now()
                    where id = :id";
        $params = ['id' => $tokenid];
        $resp = parent::noResultQuery($query,$params);
        if ($resp) {
            return $resp;
        }else{
            return 0;
        }
    }
}
```

Fuente: Elaboración propia

Figura 49: Código fuente del api de los procesos de órdenes de compra.

10.5. Tarea T28

Crear api que permita anular las órdenes de compra.

10.5.1. Diseño

Se necesita crear el api que permita anular la orden de compra esto, con la finalidad de que, si queda obsoleta la orden de compra, el usuario logueado en el aplicativo móvil pueda liberar esa compra.

10.5.2. Código

```php
public function sendOrdenCompra($json){
    $_respuesta = new respuestas;
    $datos = json_decode($json,true);
    //============================================================================INI-TOKEN
    if(!isset($datos['token'])){
        return $_respuesta->error_401();
    }else {
        $tokenarray = $this->getToken($datos['token']);
        if(!$tokenarray){
            return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
        }
    }
    //============================================================================FIN-TOKEN

    if (!isset($datos['compra']) || !isset($datos['categoria']) || !isset($datos['subcategoria'])
    || !isset($datos['producto']) || !isset($datos['marca']) || !isset($datos['moneda'])
    || !isset($datos['estado'])){
        return $_respuesta-> error_400();
    }else{
        $this->compra       = $datos['compra'];
        $this->categoria    = $datos['categoria'];
        $this->subcategoria = $datos['subcategoria'];
        $this->producto     = $datos['producto'];
        $this->marca        = $datos['marca'];
        $this->moneda       = $datos['moneda'];
        $this->estado       = $datos['estado'];

        //Actualizamos la Orden de Compra como anulado / eliminado
        $query = "  update  detalle_compras
                    set     codigo_estado = :estado
                    where   compras_codigo = :compra
                            and categorias_codigo = :categoria
                            and subcategoria_codigo = :subcategoria
                            and productos_codigo = :producto
                            and ifnull(marca_codigo,0) = :marca
                            and moneda_codigo = :moneda";
        $params = [
            'compra'        => $this->compra,
            'categoria'     => $this->categoria,
            'subcategoria'  => $this->subcategoria,
            'producto'      => $this->producto,
            'marca'         => $this->marca,
            'moneda'        => $this->moneda,
            'estado'        => $this->estado
        ];
        $resulset = parent::noResultQuery($query,$params);
        if ($resulset){
        $result = $_respuesta->response;
        $result['status'] = "ok";
        $result['result'] = array(
            "mensaje" => "Datos guardados correctamente"
        );
```

```
67              return $result;
68              }else{
69              return $_respuesta->error500("Ocurrió un problema enviando datos al servicio.");
70              }
71          }
72      }
```

Fuente: Elaboración propia

Figura 50: Código fuente del api con sus procesos de anular para las órdenes de compra.

10.6. Tarea T29

Crear api que permite confirmar la compra de los productos en la orden de compra.

10.6.1. Diseño

Se necesita crear el api que permita hacer la confirmación de la compra, esto es muy importante para cerrar el circuito de orden de compra.

10.6.2. Código

```
15      public function sendOrdenCompra($json){
16          $_respuesta = new respuestas;
17          $datos = json_decode($json,true);
18          //=============================================================================INI-TOKEN
19          if(!isset($datos['token'])){
20              return $_respuesta->error_401();
21          }else {
22              $tokenarray = $this->getToken($datos['token']);
23              if(!$tokenarray){
24                  return $_respuesta->error_401("El token que ha enviado es inválido o ha caducado");
25              }
26          }
27          //=============================================================================FIN-TOKEN
28
29          if (!isset($datos['compra']) || !isset($datos['categoria']) || !isset($datos['subcategoria'])
30          || !isset($datos['producto']) || !isset($datos['marca']) || !isset($datos['moneda'])
31          || !isset($datos['estado'])){
32              return $_respuesta-> error_400();
33          }else{
34              $this->compra       = $datos['compra'];
35              $this->categoria    = $datos['categoria'];
36              $this->subcategoria = $datos['subcategoria'];
37              $this->producto     = $datos['producto'];
38              $this->marca        = $datos['marca'];
39              $this->moneda       = $datos['moneda'];
40              $this->estado       = $datos['estado'];
41
42              //Actualizamos la Orden de Compra como enviado
43              $query = "  update  detalle_compras
44                          set     codigo_estado = :estado
45                          where   compras_codigo = :compra
46                                  and categorias_codigo = :categoria
47                                  and subcategoria_codigo = :subcategoria
48                                  and productos_codigo = :producto
49                                  and ifnull(marca_codigo,0) = :marca
50                                  and moneda_codigo = :moneda";
```

```php
            $params = [
                'compra'        => $this->compra,
                'categoria'     => $this->categoria,
                'subcategoria'  => $this->subcategoria,
                'producto'      => $this->producto,
                'marca'         => $this->marca,
                'moneda'        => $this->moneda,
                'estado'        => $this->estado
            ];
            $resulset = parent::noResultQuery($query,$params);
            if ($resulset){
            $result = $_respuesta->response;
            $result['status'] = "ok";
            $result['result'] = array(
                "mensaje" => "Datos guardados correctamente"
            );
            return $result;
            }else{
            return $_respuesta->error500("Ocurrió un problema enviando datos al servicio.");
            }
        }
    }
```

Fuente: Elaboración propia

Figura 51: Código fuente de confirmación de órdenes de compra.

10.7. Tarea T30

Crear modelo de orden de compra y productos en el aplicativo móvil.

10.7.1. Diseño

Se necesita crear el modelo de orden de compra y productos en el aplicativo móvil, esto con la finalidad de poder manejar los datos.

10.7.2. Código

```java
package com.sebasoft.tienda11.esquema;

import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

public class OrdenCompra {
    private int id;
    private String codigo,proveedor,fecha,descripcion;
    private JSONArray jdetalle;
    private ArrayList<ordencompra_marca> listamarca;
    public OrdenCompra(){

    }

    public OrdenCompra(int id,String codigo,String proveedor,String fecha,String descripcion,
                       JSONArray jdetalle){
        this.id = id;
        this.codigo = codigo;
        this.proveedor = proveedor;
        this.fecha = fecha;
        this.descripcion = descripcion;
        this.jdetalle = jdetalle;

        onDetalleOC(jdetalle);
    }

    private void onDetalleOC(JSONArray jdata){
```

```
30
31          listamarca = new ArrayList<ordencompra_marca>();
32          ordencompra_marca detalle;
33          try {
34              for (int i = 0; i < jdata.length(); i++) {
35                  detalle = new ordencompra_marca(
36                          i,
37                          jdata.getJSONObject(i).getString("categoria"),
38                          jdata.getJSONObject(i).getString("subcategoria"),
39                          jdata.getJSONObject(i).getString("producto"),
40                          jdata.getJSONObject(i).getString("marca"),
41                          jdata.getJSONObject(i).getString("moneda"),
42                          jdata.getJSONObject(i).getString("subtotal"),
43                          codigo,
44                          jdata.getJSONObject(i).getString("categorias_codigo"),
45                          jdata.getJSONObject(i).getString("subcategoria_codigo"),
46                          jdata.getJSONObject(i).getString("productos_codigo"),
47                          jdata.getJSONObject(i).getString("marca_codigo"),
48                          jdata.getJSONObject(i).getString("moneda_codigo"),
49                          jdata.getJSONObject(i).getJSONArray("colores"));
50                  listamarca.add(detalle);
51              }
52          }catch (JSONException e){
53              e.getMessage();
54          }
55
56
57      }
58
59      public ArrayList<ordencompra_marca> getlistamarca() {
60          return listamarca;
61      }
62
63      public void setListaoc(ArrayList<ordencompra_marca> listamarca) {
64          this.listamarca = listamarca;
65      }
66
67
68      public int getId() {
69          return id;
70      }
71
72      public void setId(int id) {
73          this.id = id;
74      }
75
76      public ArrayList<ordencompra_marca> getListamarca() {
77          return listamarca;
78      }
79
80      public void setListamarca(ArrayList<ordencompra_marca> listamarca) {
81          this.listamarca = listamarca;
82      }
83
84
85      public JSONArray getJdetalle() {
86          return jdetalle;
87      }
88
89      public String getCodigo() {
90          return codigo;
91      }
92
93      public String getDescripcion() {
94          return descripcion;
95      }
96
97      public String getFecha() {
98          return fecha;
99      }
100
101     public String getProveedor() {
102         return proveedor;
```

```
103     }
104
105     public void setCodigo(String codigo) {
106         this.codigo = codigo;
107     }
108
109     public void setDescripcion(String descripcion) {
110         this.descripcion = descripcion;
111     }
112
113     public void setFecha(String fecha) {
114         this.fecha = fecha;
115     }
116
117     public void setJdetalle(JSONArray jdetalle) {
118         this.jdetalle = jdetalle;
119     }
120
121     public void setProveedor(String proveedor) {
122         this.proveedor = proveedor;
123     }
124
125 }
```

Fuente: Elaboración propia

Figura 52: Código fuente del modelo de orden de compra.

```
1  package com.sebasoft.tienda11.esquema;
2
3  public class Producto {
4      private String categoria,subcategoria,producto,observacion,unidad_medida,estado,url_imagen,stock;
5      private int idtienda,idcategoria,idsubcategoria,id,idUnidadMedida,idusuario,idProducto;
6
7      public Producto(){
8          //Contructor vacio
9      }
10
11     public Producto(int id,int idtienda,int idcategoria,int idsubcategoria,int idProducto,int idusuario,int idUnidadMedida
12             ,String categoria,String subcategoria,String producto,String observacion,String unidad_medida
13             ,String estado,String url_imagen,String stock){
14
15         this.idtienda = idtienda;
16         this.idcategoria = idcategoria;
17         this.idsubcategoria = idsubcategoria;
18         this.id = id;
19         this.idusuario = idusuario;
20         this.idUnidadMedida = idUnidadMedida;
21         this.categoria = categoria;
22         this.subcategoria = subcategoria;
23         this.producto = producto;
24         this.observacion = observacion;
25         this.unidad_medida = unidad_medida;
26         this.estado = estado;
27         this.url_imagen = url_imagen;
28         this.stock = stock;
29         this.idProducto = idProducto;
30     }
31
32     public int getIdProducto() {
33         return idProducto;
34     }
35
36     public void setIdProducto(int idProducto) {
37         this.idProducto = idProducto;
38     }
39
40     public String getStock() {
41         return stock;
42     }
43
44     public void setStock(String stock) {
45         this.stock = stock;
46     }
47
48     public String getUrl_imagen() {
49         return url_imagen;
50     }
51
52     public void setUrl_imagen(String url_imagen) {
53         this.url_imagen = url_imagen;
54     }
55
56     public int getId() {
57         return id;
58     }
```

```
60      public int getIdcategoria() {
61          return idcategoria;
62      }
63
64      public int getIdsubcategoria() {
65          return idsubcategoria;
66      }
67
68      public int getIdtienda() {
69          return idtienda;
70      }
71
72      public int getIdUnidadMedida() {
73          return idUnidadMedida;
74      }
75
76      public int getIdusuario() {
77          return idusuario;
78      }
79
80      public String getCategoria() {
81          return categoria;
82      }
83
84      public String getEstado() {
85          return estado;
86      }
87
88      public String getProducto() {
89          return producto;
90      }
91
92      public String getObservacion() {
93          return observacion;
94      }
95
96      public String getSubcategoria() {
97          return subcategoria;
98      }
99
100     public String getUnidad_medida() {
101         return unidad_medida;
102     }
103
104     public void setCategoria(String categoria) {
105         this.categoria = categoria;
106     }
107
108     public void setEstado(String estado) {
109         this.estado = estado;
110     }
111
112     public void setId(int id) {
113         this.id = id;
114     }
115
116     public void setIdcategoria(int idcategoria) {
117         this.idcategoria = idcategoria;
118     }
119
120     public void setIdsubcategoria(int idsubcategoria) {
121         this.idsubcategoria = idsubcategoria;
122     }
123
124     public void setIdtienda(int idtienda) {
125         this.idtienda = idtienda;
126     }
127
128     public void setIdUnidadMedida(int idUnidadMedida) {
129         this.idUnidadMedida = idUnidadMedida;
130     }
131
132     public void setProducto(String producto) {
133         this.producto = producto;
134     }
135
136     public void setIdusuario(int idusuario) {
137         this.idusuario = idusuario;
138     }
139
140     public void setObservacion(String observacion) {
141         this.observacion = observacion;
142     }
143
144     public void setSubcategoria(String subcategoria) {
145         this.subcategoria = subcategoria;
146     }
147
148     public void setUnidad_medida(String unidad_medida) {
149         this.unidad_medida = unidad_medida;
150     }
151 }
```

Fuente: Elaboración propia

Figura 53: Código fuente del modelo de producto para las órdenes de compra.

11. Desarrollo de Sprint 6

11.1. Tarea T31

Crear pantalla de listado de orden de compra en el aplicativo móvil asignado al usuario logueado.

11.1.1. Diseño

Se necesita crear la pantalla que va a listar las órdenes de compra asociadas al usuario logueado en el aplicativo móvil.

Fuente: Elaboración propia

Figura 54: Pantalla del listado de órdenes de compra.

11.1.2. Código

```java
package com.sebasoft.tienda11.ui.fragment;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.esquema.Aplicacion;
import com.sebasoft.tienda11.esquema.OrdenCompra;
import com.sebasoft.tienda11.ui.controller.adapter_OC_rv;

import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

public class fragment_ListarOC extends Fragment {
    private fragment_CrearArticulo.OnFragmentInteractionListener mListener;
    private ListView lv_oc;
    private RecyclerView rv_listaoc;
    private Aplicacion app;
    private String Servidor;
    private int idtienda,iduser;
    private ArrayList<OrdenCompra> listaOC;
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_listaroc, container, false);
        //lv_oc = (ListView) rootView.findViewById(R.id.lv_ordencompra);
        rv_listaoc = (RecyclerView) rootView.findViewById(R.id.rv_oc_lista);
        rv_listaoc.setLayoutManager(new LinearLayoutManager(this.getContext()));

        app = (Aplicacion) rootView.getContext().getApplicationContext();

        Servidor = app.getServidor();
        idtienda = app.getIdtienda();
        iduser = app.getIduser();

        onListarOrdenCompra();
        return rootView;
    }

    public void onListarOrdenCompra(){
        String tienda =Integer.toString(idtienda);
        String usuario =Integer.toString(iduser);
        String servicio = Servidor+"/"+"orden_compra?tienda="+tienda+"&usuario="+usuario;

        RequestQueue queue = Volley.newRequestQueue(getContext());
        StringRequest postRequest = new StringRequest( Request.Method.GET, servicio,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        onCargarLista(new String(response));
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(getContext(),"Api sin respuestas",Toast.LENGTH_SHORT).show();
                    }
                });

```

```
        queue.add(postRequest);
    }

    public void onCargarLista(String response){
        OrdenCompra ordencompra;
        listaOC = new ArrayList<OrdenCompra>();
        try {
            JSONArray jsonArray = new JSONArray(response);
            for(int i=0;i<jsonArray.length();i++){
                ordencompra = new OrdenCompra(
                        i,
                        jsonArray.getJSONObject(i).getString("codigo"),
                        jsonArray.getJSONObject(i).getString("proveedor"),
                        jsonArray.getJSONObject(i).getString("fecha"),
                        jsonArray.getJSONObject(i).getString("descripcion"),
                        jsonArray.getJSONObject(i).getJSONArray("marcas")
                );

                listaOC.add(ordencompra);
            }
            rv_listaoc.setAdapter(new adapter_OC_rv(listaOC, new adapter_OC_rv.OnItemClickListener() {
                @Override
                public void onItemClick(OrdenCompra item) {

                    Fragment frag = new fragment_oc_detalle(item);
                    FragmentManager fmanager = getFragmentManager();
                    FragmentTransaction fragmentTransaction = fmanager.beginTransaction();
                    fragmentTransaction.replace(R.id.app_bar_main, frag);
                    fragmentTransaction.commit();
                    fragmentTransaction.addToBackStack(null);
                    app.setFragmenadd();


                }
            }));

        }catch (JSONException e){
            e.getMessage();
        }

    }

    // TODO: Rename method, update argument and hook method into UI event
    public void onButtonPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof fragment_CrearArticulo.OnFragmentInteractionListener) {
            mListener = (fragment_CrearArticulo.OnFragmentInteractionListener) context;
        } else {
            throw new RuntimeException(context.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }
    public interface OnFragmentInteractionListener {
        // TODO: Update argument type and name
        void onFragmentInteraction(Uri uri);
    }
}
```

Fuente: Elaboración propia

Figura 55:Código fuente del listado de órdenes de compra en el aplicativo móvil.

11.2. Tarea T32

Crear pantalla para modificar, anular o confirmar la orden de compra.

11.2.1. Diseño

Crear pantalla para modificar, anular o confirmar la orden de compra.

Fuente: Elaboración propia

Figura 56: Pantalla del listado de órdenes de compra.

11.2.2. Código

```
1 package com.sebasoft.tienda11.ui.controller;
2
3 import android.annotation.SuppressLint;
4 import android.app.Activity;
5 import android.content.Context;
6 import android.content.DialogInterface;
7 import android.os.Vibrator;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11 import android.widget.ImageButton;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import androidx.annotation.NonNull;
16 import androidx.appcompat.app.AlertDialog;
17 import androidx.fragment.app.FragmentManager;
18 import androidx.recyclerview.widget.LinearLayoutManager;
19 import androidx.recyclerview.widget.RecyclerView;
20
21 import com.android.volley.Request;
22 import com.android.volley.RequestQueue;
23 import com.android.volley.Response;
24 import com.android.volley.VolleyError;
25 import com.android.volley.toolbox.JsonObjectRequest;
26 import com.android.volley.toolbox.Volley;
```

```java
import com.sebasoft.tienda11.R;
import com.sebasoft.tienda11.api.apiConexion;
import com.sebasoft.tienda11.esquema.ordencompra_marca;


import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class adapter_oc_marca extends RecyclerView.Adapter<adapter_oc_marca.marca_vh> {

    ArrayList<ordencompra_marca> listamarca;
    private ordencompra_marca item;
    final adapter_oc_marca.OnItemClickListener listener;
    String Servidor;
    apiConexion apiconexion;
    Vibrator vibrator;
    Context context;
    private Activity activity;
    private FragmentManager frmanager;

    public interface OnItemClickListener{
        void onItemClick(ordencompra_marca item);

    }

    public adapter_oc_marca(ArrayList<ordencompra_marca> listamarca, OnItemClickListener listener, Activity ac
        this.listamarca = listamarca;
        this.listener = listener;
        this.activity = activity;
        this.frmanager = frmanager;
    }

    @NonNull
    @Override
    public marca_vh onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        apiconexion = new apiConexion(parent.getContext());
        Servidor = apiconexion.getServidor();
        vibrator = (Vibrator) parent.getContext().getSystemService(parent.getContext().VIBRATOR_SERVICE);
        context = parent.getContext();
        return new marca_vh(LayoutInflater.from(parent.getContext()).inflate(R.layout.fragment_oc_detalle_marc
    }
    public void removeItem(int position) {
        listamarca.remove(position);
        notifyItemRemoved(position);
    }

    @Override
    public void onBindViewHolder(@NonNull marca_vh holder, @SuppressLint("RecyclerView") int position) {
        item = listamarca.get(position);

        holder.tv_marca.setText(item.getMarca());
        holder.tv_producto.setText(item.getProducto());
        holder.tv_precio.setText(item.getSubtotal() +" "+item.getMoneda());
        holder.rv_color.setAdapter(new adapter_oc_marca_color(item.getListacolor()));
        holder.rv_color.setLayoutManager(new LinearLayoutManager(holder.rv_color.getContext()));
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               // listener.onItemClick(item);
            }
        });
        holder.ib_modificar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ordencompra_marca product = listamarca.get(position);
                df_modificar_oc modificar_oc = new df_modificar_oc(product,activity,context);
                modificar_oc.show(frmanager,"modificar_oc");
                android.app.Fragment frag = activity.getFragmentManager().findFragmentByTag("modificar_oc");
                if (frag!=null){
                    activity.getFragmentManager().beginTransaction().remove(frag).commit();
                }
            }
        });
        holder.ib_eliminar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ordencompra_marca product = listamarca.get(position);
                AlertDialog.Builder dialogo = new AlertDialog.Builder(v.getContext());
                dialogo.setTitle("Quitar Producto de la Orden de Compra");
                dialogo.setMessage("¿Quitar el total de "+product.getProducto()+" "+product.getMarca()+"?");
                dialogo.setCancelable(false);

                dialogo.setPositiveButton("Confirmar", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialogo, int id) {
                        // pb_guardar.setVisibility(getView().VISIBLE);
                        onEnviarCompra(product.getOcompra_id(), product.getCat_codigo(), product.getSubcat_codigo(),
                                product.getPro_codigo(), product.getMarca_codigo(), product.getMoneda_codigo(), "2");
                        removeItem(position);
                        if (getItemCount() == 0) {
                            activity.onBackPressed();
```

```
                    }
                }

            }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialogo, int id) {
                    // TODO por ahora no hacemos nada...
                }
            });
            dialogo.show();
        }
    });
    holder.ib_enviar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ordencompra_marca product = listamarca.get(position);
            AlertDialog.Builder dialogo = new AlertDialog.Builder(v.getContext());
            dialogo.setTitle("Enviar Producto de la Orden de Compra");
            dialogo.setMessage("¿Confirmas el total de "+product.getProducto()+" "+product.getMarca()+"?");
            dialogo.setCancelable(false);

            dialogo.setPositiveButton("Confirmar", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialogo, int id) {
                    // pb_guardar.setVisibility(getView().VISIBLE);
                    onEnviarCompra(product.getOcompra_id(), product.getCat_codigo(), product.getSubcat_codigo(),
                            product.getPro_codigo(), product.getMarca_codigo(), product.getMoneda_codigo(), "2");
                    removeItem(position);
                    if (getItemCount() == 0) {
                        activity.onBackPressed();
                    }
                }

            }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialogo, int id) {
                    // TODO por ahora no hacemos nada...
                }
            });
            dialogo.show();
        }
    });
}

public void onEnviarCompra(String compra,String categoria,String subcategoria,
                           String producto,String marca,String moneda,String estado){
    String servicio;
    Map<String, String> data = new HashMap<>();

    servicio = Servidor+"/orden_compra";
    data.put("compra", compra);
    data.put("categoria", categoria);
    data.put("subcategoria", subcategoria);
    data.put("producto",producto);
    data.put("marca", marca);
    data.put("moneda", moneda);
    data.put("estado",estado);
    data.put("token", apiconexion.getToken());
    JSONObject jsonData = new JSONObject(data);

    RequestQueue queue = Volley.newRequestQueue(context);
    JsonObjectRequest postRequest = new JsonObjectRequest( Request.Method.POST, servicio,
            jsonData,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    JSONObject Obj,Result;
                    Obj = response;
                    String  status,mensaje,sidcat,sidsubcat,sidpro;
                    try {
                        status = Obj.getString("status");
                        Result = Obj.getJSONObject("result");
                        if (status.equals("ok")){
                            //-----------------------------------------OK-------------------------------

                            //Toast.makeText(context,"OK",Toast.LENGTH_LONG).show();
                        }else{
                            vibrator.vibrate(400);
                            mensaje = Result.getString("error_message");
                            Toast.makeText(context,mensaje,Toast.LENGTH_LONG).show();
                        }
                    } catch (JSONException e) {
                        //El servio no trajo nada
                        vibrator.vibrate(400);
                        e.printStackTrace();
                        Toast.makeText(context,"El servicio retorno un error inesperado",Toast.LENGTH_LONG).show();
                    }
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    vibrator.vibrate(400);
                    Toast.makeText(context,"El servicio no está disponible",Toast.LENGTH_LONG).show();
                    error.printStackTrace();
                }
```

```
214                 });
215         queue.add(postRequest);
216
217     }
218
219     @Override
220     public int getItemCount() {
221         return listamarca.size();
222     }
223
224     static class marca_vh extends RecyclerView.ViewHolder{
225
226         TextView tv_marca,tv_producto,tv_precio;
227         ImageButton ib_modificar,ib_eliminar,ib_enviar;
228         RecyclerView rv_color;
229
230         public marca_vh(@NonNull View itemView) {
231             super(itemView);
232
233             tv_marca = itemView.findViewById(R.id.tv_oc_marca);
234             tv_producto = itemView.findViewById(R.id.tv_oc_marca_producto);
235             tv_precio = itemView.findViewById(R.id.tv_oc_marca_precio);
236             rv_color = itemView.findViewById(R.id.rv_oc_marca_color);
237             ib_modificar = itemView.findViewById(R.id.ib_oc_modificar);
238             ib_eliminar = itemView.findViewById(R.id.ib_oc_eliminar);
239             ib_enviar = itemView.findViewById(R.id.ib_oc_enviar);
240
241         }
242     }
243 }
```

Fuente: Elaboración propia

Figura 57: Código fuente del confirmar, anular y modificar de órdenes de compra.

11.3. Tarea T33

Crear proceso de aprobación de la orden de compra para ingreso de productos al stock.

11.3.1. Diseño

Crear un trigger que permita realizar el ingreso al stock de los productos ingresados con las órdenes de compra.

11.3.2. Código

```
1  use ventas;
2  drop trigger if exists itrg_compras_iniciar_transaccion;
3
4  delimiter $$
5  create definer = current_user trigger itrg_compras_iniciar_transaccion
6  after update on compras for each row
7  begin
8      DECLARE finished INTEGER DEFAULT 0;
9      DECLARE cat,subcat,pro,color,talla,marca,moneda,um,transaccion int;
10     DECLARE detalle_transaccion int default 0;
11     DECLARE cantidad,precio decimal(10,2);
12     DECLARE cur_det_oc
13         CURSOR FOR
14             select  a.categorias_codigo,
15                     a.subcategoria_codigo,
16                     a.productos_codigo,
17                     a.color_codigo,
18                     a.tallas_codigo,
19                     a.marca_codigo,
20                     a.cantidad,
21                     a.precio,
22                     a.moneda_codigo,
23                     a.unidad_medida_codigo
24             from    detalle_compras a
25             where   a.compras_codigo = NEW.codigo
26                     and codigo_estado = 7
27             order by codigo;
28     DECLARE CONTINUE HANDLER
29         FOR NOT FOUND SET finished = 1;
30
31     set @tienda = NEW.tienda_codigo;
32     set @codigo = NEW.codigo;
```

```
33
34      if NEW.estado_codigo = 5 then
35
36          insert into transaccion(tipo_transaccion_codigo,fecha,estado_codigo,
37          venta_compra_codigo,tienda_codigo,descripcion,tipo_documento)
38          value(3,now(),1,@codigo,@tienda,'Aprobación de Orden de compra',1);
39          set transaccion = LAST_INSERT_ID();
40
41          insert into detalle_transaccion(transaccion_codigo,transaccion_tipo_transaccion_codigo,
42          codigo,categorias_codigo,subcategoria_codigo,productos_codigo,codigo_estado,
43                                          marca_codigo,color_codigo,tallas_codigo,cantidad,
44                                          precio,moneda_codigo,unidad_medida_codigo)
45          select  transaccion,
46                  3,
47                  a.codigo,
48                  a.categorias_codigo,
49                      a.subcategoria_codigo,
50                      a.productos_codigo,
51                      1,
52                      a.marca_codigo,
53                      a.color_codigo,
54                      a.tallas_codigo,
55                      a.cantidad,
56                      a.precio,
57                      a.moneda_codigo,
58                      a.unidad_medida_codigo
59              from    detalle_compras a
60              where   a.compras_codigo = NEW.codigo
61                      and codigo_estado = 7;
62      end if;
63  end$$
64  delimiter ;
```

Fuente: Elaboración propia

Figura 58: Código fuente del trigger para aprobar la orden de compra y se ingresa al stock.

12. Referencias del código Fuente (GitHub)

Para efectos prácticos se está compartiendo el url donde se encuentra los proyectos:

12.1. Api-Rest

https://github.com/towo10/ApiTienda

12.2. Aplicativo Android

https://github.com/towo10/ProyectoTienda

UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Declaratoria de Originalidad del Autor

Yo, TORRES NUÑEZ WILDER estudiante de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA ESTE, declaro bajo juramento que todos los datos e información que acompañan la Tesis titulada: "APLICACIÓN MÓVIL CON ARQUITECTURA API-REST PARA MYPES", es de mi autoría, por lo tanto, declaro que la Tesis:

1. No ha sido plagiada ni total, ni parcialmente.
2. He mencionado todas las fuentes empleadas, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes.
3. No ha sido publicada, ni presentada anteriormente para la obtención de otro grado académico o título profesional.
4. Los datos presentados en los resultados no han sido falseados, ni duplicados, ni copiados.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Nombres y Apellidos	**Firma**
TORRES NUÑEZ WILDER **DNI:** 43621579 **ORCID** 0000-0002-0504-4569	Firmado digitalmente por: TTORRESNU el 22-12-2021 12:47:04

Código documento Trilce: INV - 0583514