DESARROLLO DE UN SOFTWARE DE GESTIÓN DE INVENTARIOS PARA LA DROGUERÍA ONASSIS

Iván Felipe Rodríguez García

11161711563

Universidad Antonio Nariño

Programa Ingeniería de Sistemas y Computación

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2021

DESARROLLO DE UN SOFTWARE DE GESTIÓN DE INVENTARIOS PARA LA DROGUERÍA ONASSIS

Iván Felipe Rodríguez García

Proyecto de grado presentado como requisito parcial para optar al título de:

Ingeniero de sistemas y Computación

Director:

David Alberto Herrera Álvarez

Doctor en ingeniería de sistemas

Rosalba Cruz Cepeda

Asesor metodológico

Desarrollo de software

Universidad Antonio Nariño

Programa Ingeniería de Sistemas y Computación

Facultad de Ingeniería de sistemas

Bogotá, Colombia

2021

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____,

Cumple con los requisitos para optar

Al título de _____.

_____

Firma del Tutor

_____

Firma Jurado

_____

Firma Jurado

Contenido

Pág.

Preliminares

(Dedicatoria)

Mi trabajo de grado está dedicado a mis padres Alberto y Aurora, quienes con su apoyo constante e incondicional trabajaron conmigo de la mano en este proceso que me deja como satisfacción el poder cumplir una de mis metas.

También agradezco a cada una de las personas que aportaron en mi proceso de formación, sus consejos, correcciones y palabras de aliento que hicieron de mi una mejor persona.

Agradecimientos

Inicialmente dedico los agradecimientos de mi trabajo de grado a Dios, por ser mi guia, por darme sabiduríay constancia en el proceso de formación que me permitió culminar con éxito mi meta propuesta.

Finalizando agradezco también a la Universidad Antonio Nariño, a la facultad de Ingeniería y profesores quienes con sus enseñanzas, conocimientos, apoyo incondicional y dedicación permitieron que pueda crecer día a día como profesional y persona.

Resumen

En este documento se podrá evidenciar de cómo se abordá desde la ingeniería de sistemas problemas cotidianos y como con nuestros conocimientos se pudo brindar la mejor solución posible. Es acá donde entra en juego la droguería Onassis una pequeña empresa que lleva en el mercado más de 30 años prestando sus servicios, con el tiempo el paso de la tecnología la ha dejado rezagada. La droguería Onassis identifica las siguientes problemáticas en su empresa:

- Mala rotación de sus productos
- Errores en los inventarios
- Monotonía al realizar los inventarios
- Pérdida de tiempo al hacer reprocesos
- Falta de información de sus clientes.

Debido a esto, se implementa un software de inventarios que controla las problemáticas evidenciadas por ellos; el software tiene las siguientes funcionalidades:

- Entrada y salida de productos
- Modificación de registros,
- Eliminación de productos
- Registro de clientes y facturación.

Cabe resaltar que no se incluirán informes o cálculos contables. Para obtener el mejor resultado se implementó la metodología ágil Scrum, la cual brindó una flexibilidad a los cambios y un mejor rendimiento del proyecto.

Adicional a esto, por medio de este documento se captura las evidencias del desarrollo del proyecto de acuerdo con los requerimientos identificados por los administradores de la Droguería Onassis; esto se logró gracias a la implementación de la metodología ágil Scrum que controló el desarrollo del proyecto.

También se puede evidenciar la utilización de las herramientas tecnológicas para culminar el proyecto; herramientas como: lenguaje de programación Java mediante framework NetBeans, el diseño de la interfaz mediante JSP y Bootstrap y un repositorio de información en lenguaje de MYSQL gestionado por phpMyAdmin.

Lo anterior, permite mostrar en el documento los resultados obtenidos durante el periodo de desarrollo del proyecto, dejando como conclusión, la satisfacción por parte del usuario con cada uno de los aspectos que conforma el Software; además, se identifica que se logra contribuir en la mejora constante de las funciones que se desempeñan en el día a día de la Droguería Onassis.

Cabe resaltar, que como todo Software o tecnología esta sujeto a mejoras o actualización para el beneficio del usuario.

Abstract

In this document it will be possible to show how everyday problems are addressed from systems engineering and how with our knowledge the best possible solution could be provided. This is where the Onassis drugstore comes into play, a small company that has been in the market for more than 30 years providing its services, over time the passage of technology has left it behind. The Onassis drugstore identifies the following problems in its company, poor rotation of its products, errors in its inventories, monotony when making them, loss of time when doing reprocessing and lack of information from its customers. Because this is implemented an inventory software that controls the problems evidenced by them; The software has the following functionalities: entry and exit of products, modification of records, elimination of products, customer registration and billing. It should be noted that reports or accounting calculations will not be included. To obtain the best result, the agile Scrum methodology is implemented, which provides flexibility to changes and better project performance.

In addition to this, this document captures the evidence of the development of the project in accordance with the requirements identified by the administrators of the Onassis drugstore; This was achieved thanks to the implementation of the agile Scrum methodology which controlled the development of this project. You can also show the use of technológical tools to complete the project, tools while as a Java programming language using NetBeans framework, the interface design using JSP and Bootstrap and a repository of information in MYSQL language managed by phpMyAdmin.

Thanks to this, this document shows the results obtained during the development period of the project, leaving as a conclusión the satisfaction on the part of the user with each of the aspects that make up the software, in addition it is identified that it is possible to contribute to constant improvement of the functions that are performed in the day to day of the onasís drugstore.

It should be noted that like all software or technology it is subject to improvements or updates for the benefit of the user.

1. Planteamiento Del Problema

1.1 Descripción Del Problema

La droguería Onassis surge en la ciudad de Bogotá en agosto de 1989, a partir de la necesidad de brindar un servicio confiable al sector con respecto a sus necesidades en la rama de la farmacia y cuidado personal; esto con el apoyo de sus administradores que cuentan con un conocimiento amplio en el sector farmacéutico gracias a sus experiencias laborales y estudios.

La droguería Onassis tiene como representantes legales a Ángel Alberto Rodríguez y a Campo Elías Arellano, quienes se encargan de la comercialización de productos farmacéuticos, el cuidado personal y la atención al cliente mediante su punto de venta.

Con base en la experiencia, han identificado que la tecnología avanza de manera exponencial en el mercado farmacéutico, llegando a la conclusión de que ellos deben ir de la mano de la tecnología para poder seguir creciendo como empresa, ya que en la droguería llevan todo de manera manual y con registros físicos, lo que es muy tedioso y en ocasiones poco confiable; es por eso que requieren mejorar la manera en que se realizan los inventarios, puesto que es algo primordial en la rama de negocio en la que trabajan.

Algunos de los aspectos en los que se ven afectados son: los productos que se agotan y la poca rotación de otros, las libretas de los inventarios no son legibles o en ocasiones se extravían ocasíonando un reproceso del inventario; otra problemática hace referencia a que no se conoce un control de los clientes que frecuentan la droguería y esta información es primordial para el negocio. Por esta razón se plantea a la droguería Onassis implementar un

sistema de gestión de inventario, que le permita controlar la entrada y salida de los mismos productos.

1.2 Formulación Del Problema

La droguería Onassis al llevar el registro de sus productos de manera manual en físico, no tiene un control de inventarios, desconoce el volumen de venta de sus productos y la rotación de los mismos, lo que afecta tener un registro actualizado y real de existencias. Por lo anterior, desde la ingeniería de sistemas se planteó la siguiente hipótesis:

Con el desarrollo e implantación de un software de gestión de inventarios que facilite el registro de los productos en una base de datos, la droguería Onassis tendrá un mayor orden y control en los productos que ofrece.

1.3 Justificación

A medida que avanza el tiempo, la tecnología es parte fundamental de los negocios, sin importar su tamaño puede aportar demasiados beneficios, como impulsar el crecimiento de los mismos. Por lo anterior, este proyecto es importante para la droguería Onassis ya que proporcionará solución a las dificultades de rotación de productos, reproceso de registro de inventarios y el control de sus clientes.

A nivel económico, este proyecto ofrecerá gran beneficio a la droguería Onassis, brindando una solución acorde a las problemáticas evidenciadas en el establecimiento, específicamente, para reducir el costo de pago de horas extras a su trabajador, puesto que

evitaría realizar un reproceso de verificación o registro en los inventarios; a demás, facilitará en tiempo real la solicitud de productos, evitando demasiado stock y poca rotación.

Desde lo social, este proyecto permitirá que la droguería Onassis siga creciendo como empresa y no desaparezca, lo cual para la comunidad del barrio Onassis es beneficioso porque la droguería podrá seguir con su vocación social de ayudar a las personas; otro beneficio social es para las familias de los tres trabajadores de la droguería, ya que si esta sigue creciendo, podrán seguir teniendo una estabilidad económica para sostener a sus familias; y por último, pero no menos importante, se beneficiarán a los proveedores porque al seguir la droguería, se esta generando trabajo de manera indirecta mediante los proveedores.

A nivel tecnológico, se busca agilizar los procesos contables y contar con la información actualizada en el momento que se requiera, esto con el fin de familiarizar a la droguería Onassis con los avances tecnológicos disponibles en la actualidad.

A nivel profesional, ofrecerá la opción de desarrollar un proyecto completo, aplicando todos los conocimientos adquiridos en los últimos años en la Universidad Antonio Nariño, dando una mirada de cómo será el futuro laboral; por otra parte; permitirá mejorar las habilidades como desarrollador y comunicación con los clientes, para brindar una solución óptima para ellos; por último, como ingeniero permitirá adquirir experiencia para solucionar los problemas de la comunidad o de empresas relacionados con software para inventarios.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un Software de gestión de inventarios que facilite el registro y control de la rotación de los productos de la droguería Onassis para tener un mayor orden y control en los productos que ofrece, apoyados en la metodología Scrum.

1.4.2 Objetivos Específicos

- Analizar los requerimientos propuestos por la droguería Onassis mediante las historias de usuario, para el desarrollo del software de gestión de inventario con el fin de solucionar las problemáticas identificadas en la droguería.
- Diseñar la estructura de la base de datos, interfaz gráfica y arquitectura del software que se implementarán en la droguería Onassis utilizando herramientas tecnológicas como lenguaje Java mediante el framework de NetBeans, jsp y su estructura de datos mediante Mysql.
- Implementar la metodología ágil de Scrum durante el desarrollo del proyecto para garantizar su entrega y optimizar el resultado final del mismo.
- Ejecutar pruebas funcionales para comprobar que el sistema cumpla con los requerimientos propuestos por la droguería Onassis y que tenga una funcionalidad correcta, mediante pruebas de escritorio.

1.5 Alcance y Limitaciones Del Proyecto

1.5.1 Alcance

Según lo solicitado por droguería Onassis se implementará las siguientes funcionalidades para gestionar mejor sus inventarios:

- Módulo de venta: en este módulo se encuentran las funcionalidades de salida de productos, es la encargada de descontar los productos del inventario y la funcionalidad de facturación sin parte contable.
- Módulo de registro: en este módulo se encontrarán todas las funcionalidades de registro de productos; por otro lado, estarán las funcionalidades de modificar o eliminar algún producto.
- Módulo de reporte: en este módulo se podrá visualizar de manera sencilla los productos más vendidos por la droguería Onassis.
- Módulo de stock: este módulo incluye funcionalidades, mostrar el stock en tiempo real, buscar productos en específico, incremento de alguna existencia de un producto.
- Tecnologías: se utilizará el IDE de NetBeans con una base de datos de MySQL mediante phpMyAdmin para realizar todo el proyecto.

1.5.2 Limitaciones

Teniendo en cuenta los alcances anteriores, este proyecto contará con las siguientes limitaciones que no serán implementadas en este tiempo de trabajo, pero no se descartan para un futuro.

- No se actualizarán registros antiguos, estos se obtendrán del actual stock.
- No se implementara alcance contable en el Software establecido, los reportes de venta que se emiten no son relevantes en la administracion contable de la Droguería Onassis.
- No se manejarán actualizaciónes o soportes después de entregado el software.
- No se implementara factura de manera legal por el momento, no se descarta implementar a futuro dicha factura acorde al marco legal estipulado.
- No se contolara los aspectos de backup y seguridad.

2. Marco De Referencia

2.1 Marco Teórico

Con el transcurrir del tiempo las droguerías se han convertido en un comercio de vital importancia; el término de droguería tiene dos definiciones: empresa en la que se pueden adquirir medicamentos prescritos por algún médico o especialista. Otra definición es: establecimiento que brinda un comercio de medicamentos, productos de aseo, cosméticos entre otros productos (Concepto en definición ABC" n.d.).

Teniendo en cuenta estas definiciones se puede conocer el funcionamiento interno, para tener una idea más clara, ya que en este tipo de negocios se manejan términos que no son muy comunes para la mayoría de las personas. Uno de los términos en el que podemos profundizar es la rotación de productos, se refiere a las veces que un producto pasa por cualquiera de los siguientes procesos: ingreso, salida y venta, haciendo que las droguerías tengan una idea de su crecimiento como empresas.

Además, les da una idea de cómo pueden manejar la solicitud de pedidos a los distribuidores para que puedan priorizar los productos que son más solicitados y cuales no, para mejorar sus ganancias. Este proceso se puede realizar mediante una gestión de inventarios que tuvo sus orígenes en la cultura egipcia, donde realizaban la costumbre de almacenar grandes cantidades de alimentos para utilizar en tiempos de sequía o algún improvisto ("Duran Yosmary," 2012), pero con el transcurrir del tiempo, estos inventarios han mejorado a través de la tecnología.

Se han utilizado programas informáticos entendidos como un conjunto de pasos lógicos que facilitan ciertos procesos (CILSA, 2017), esto se puede realizar mediante un

lenguaje de programación que proporcionan la creación de estos programas; su propósito es el de facilitar la comunicación entre los humanos y las máquinas como computadores a través de algoritmos, instrucciones escritas en un lenguaje que puedan interpretar y procesar (López Mendoza, 2020).

Pero esto no es tan sencillo como parece, para que esta tarea sea más amena se pueden utilizar entorno de desarrollo integrado (IDE) que combina varias herramientas útiles de desarrollo representadas mediante una interfaz gráfica; para el caso en particular, se puede utilizar lenguaje de programación Java, es una plataforma informática cuya primera versión comercial fue en 1995 (Evan,2015); estos lenguajes se pueden combinar con otras herramientas como un framework, ejemplo NetBeans, programas como Java se pueden utilizar para la creación de aplicaciones web (Pahino, 2020).

Para la creación de una aplicación completamente funcional, se requiere de un motor de base de datos, cuya función es servir de intermediario entre la base de datos y las aplicaciones. Una base de datos es repositorio de información que permite guardar grandes cantidades de información de una manera muy ordenada (Valdés, 2007); con esto, se consigue un manejo óptimo de la información y realizar operaciones como créate, read, update y delete. Un ejemplo de base de datos es el de My SQL,es un motor de base de datos muy completo que brinda demasiadas ventajas, como: fácil de usar, una base de datos rápida, varias capas de seguridad, pocos requerimientos, eficiencia de memoria y es compatible con windows linux.

Existen varios tipos de bases de datos, pero la más relevante para el proyecto es la base de datos relacional, cuyo funcionamiento es proporcionar acceso a los datos relacionados entre sí a través de tablas que tienen un ID único llamado clave; las columnas

de estas tablas contienen atributos que facilitan la relación entre las tablas (ORACLE, 2020). Ya con esto se tendrían soportadas las herramientas necesarias para realizar un programa funcional, como por ejemplo un software de gestión de inventarios para una droguería.

Es importante tener en cuenta la estructura DAO (Data Access Object) ya que ella permite el acceso a la base de datos para así implementar los metodos de consultar, eliminar, insertar y actualizar; esto facilitará el uso de los anteriores metodos.

De acuerdo con lo anterior se concluyó que el sistema de inventarios que requiere la droguería Onassis debe estar conformado principalmente por las siguientes herramientas:

- Lenguaje de programación Java mediante el entorno de desarrollo integrado (IDE) de NetBeans.
- Base de datos en MySQL mediante el phpMyAdmin esto con el fin de manejar administración de los datos a través de páginas web.
- JavaServerPages (JSP) es la tecnología que se emplea para la creación de las páginas web mediante HTML el cual comparte el mismo lenguaje de programación del proyecto.

2.2.Antecedentes

Se realizó una búsqueda de Software y aplicaciones cuyo propósito es la gestión de inventarios para farmacias o droguerías, que permita identificar el valor agregado del software que se desplegará en la droguería Onassis.

Siki: es una compañía que brinda sistemas de información, inventarios y demás Software para farmacias; algunas de las características de los software de gestión de inventarios son los siguientes: soportes de ventas, múltiples cajas, control de inventarios,

recursos humanos, información de proveedores, contabilidad automática; ellos brindan estos servicios por planes que van desde $ 410.000 mensuales hasta $ 1.400.000 anual (SIKI Software ERP | Planes y PRECIOS Colombia, n.d.)

30Management pro: es un sistema que esta diseñado para administrar y controlar droguerías; las características que manejan son: control de productos, venta y servicio a domicilio; este software da un periodo de prueba de 15 días (ManagementPro ®, n.d.).

J4 Pro: es un sistema diseñado para droguerías y farmacias; brinda las siguientes características: facturación pos, control de inventarios, imprime códigos de barra, tienda en línea y múltiples sucursales; este software da un periodo de prueba de 7 días (Software Para Droguerías y Farmacias | J4Pro Administración Inteligente, n.d.).

Farmatic: es un software que esta diseñado para facilitar los procesos en las farmacias; algunas de sus características son facturación, control de inventarios, reportes, elaboración de presupuestos, fidelización, gestión con proveedores y contabilidad automática; este software se debe adquirir con licencia (Consoft - Características de Farmatic, n.d.).

Bitfarma: es un Software cuya principal función es acompañar a las farmacias o droguerías en su crecimiento mediante el control y acompañamiento, sus características son: gestión de inventarios, control de productos, ventas, facturación, atención farmacéutica y formulación magistral; este software es de pago (Programa Informático de Gestión Para Oficina de Farmacia, n.d.). En la tabla 1 se puede apreciar un breve comparativo en cuanto a las aplicaciones similares que se encuentran en el mercado.

Tabla 1

Comparación de aplicaciones similares

Nombre	Registro de productos	Registro de clientes	Salida de productos sin sistema contable	Stock	Gratis
Siki	X	x		x	
Management pro	X	x		x	
J4 pro	X	x	x	x	
Farmatic	X	x		x	
Bitfarma	X	x		x	

Nota. con base enla información de (Programa Informático de Gestión Para Oficina de Farmacia, n.d, Software Para Droguerías y Farmacias | J4Pro Administración Inteligente, n.d, Consoft - Características de Farmatic, n.d, ManagementPro ®, n.d, SIKI Software ERP | Planes y PRECIOS Colombia, n.d.). Fuente: elaboración propia.

Teniendo en cuenta los anteriores Software de inventarios para farmacias o droguerías, se puede visualizar que tienen una labor muy completa, pero en este caso no cumple con dos requisitos diferenciales e importantes para la droguería Onassis, en lo referente a la parte de la salida de los productos de su inventario sin una parte de facturación contable, el otro factor es el económico ya que estos Software solo dan periodos de prueba

para mirar su funcionalidad, después de esto se debe cancelar membresías o pagar el programa. El sistema que se propone se adecua a las necesidades brindadas por la droguería Onassis, en primera instancia, la salida de productos sin necesidad de registros por el área contable y la segunda es un software que se esta otorgando totalmente gratuito para una pequeña empresa, dándoles beneficios económicos y operativos.

2.3.Marco Legal

2.3.1. Ley 23 de 1982 Derechos de autor

Según esta ley que aplica en el territorio colombiano, las ideas o contenido conceptual de obras literarias, artísticas y científicas no son objetivo de apropiación personal, por lo tanto, este proyecto bajo el cumplimiento de esta ley (CECOLDA - Centro Colombiano Del Derecho de Autor - LEY 23 DE 1982 SOBRE DERECHO DE AUTOR, n.d.).

2.3.2. Ley 1581 2015 Protección de datos personales

Según esta ley se establecerá el correcto tratamiento, protección y actualización de los datos personales de los clientes frecuentados por la droguería Onassis; por lo tanto, este proyecto se ciñe a esta ley y trabaja bajo el cumplimiento de la misma (Legal Team Workers, n.d.).

2.3.3. Ley 962 2005 Facturación Obligatoria

Según esta ley se aplica en el territorio colombiano en donde se obliga a expedir factura a las personas o entidades que tengan calidad de comercio, en este proyecto no se implementara ya que con la estructura que maneja en la Droguería onassis no se tiene vigente actualmente, no se descarta la implementación futura en el software.

3. Aspectos Metodológicos

3.1.Descripción de la Metodología Scrum

Scrum es una de las muchas metodologías ágiles que se encuentran en auge en el mercado. Teniendo un enfoque donde se aplica un conjunto de buenas prácticas para lograr un trabajo colaborativo y obtener el mejor resultado posible del proyecto, estas prácticas tienen un apoyo entre sí para lograr másificar el resultado.

Scrum maneja unas entregas parciales y regulares que le permite obtener un beneficio al receptor del proyecto para obtener el mejor resultado final. Scrum está especialmente enfocado para proyectos cuyo entorno es complejo, se solicitan resultados en un periodo de tiempo corto, donde los requisitos son cambiantes o no muy bien definidos, o donde los proyectos innovan con el pasar del tiempo (Albaladejo 2008).

3.1.1. Roles

En Scrum el equipo garantiza construir un software de calidad, se definen las características del proyecto que se desea construir y se conforman los roles así:

Product owner: representa a los accionistas o clientes que darán uso al software; está enfocado en la parte del negocio y es el que le brinda al equipo la visión del proyecto que se desea desarrollar (Clarís. Pau 2015). En el Presente proyecto el producto owner esta representado por Ángel Alberto Rodríguez, uno de los representantes legales de la droguería Onassis.

Scrum máster: es la persona encargada de liderar el equipo de trabajo; es quien vela por que se cumplan las reglas y procesos de la metodología y gestiona la reducción de

impedimentos del proyecto (Clarís. Pau 2015). En el presente proyecto el Scrum máster esta representado por Iván Felipe Rodríguez, estudíante de ingeniería de sistemas y computación.

Equipo de desarrollo: es el grupo de profesionales con los conocimientos necesarios para llevar el proyecto a su culminación, teniendo en cuenta las historias con las que se comprometieron desde el inicio del proyecto (Clarís. Pau 2015). En el presente proyecto el equipo de desarrollo está representado por Iván Felipe Rodríguez estudíante de ingeniería de sistemas y computación.

3.1.2. Buenas prácticas de Scrum

Desarrollo incremental: se basó en desarrollo de los requisitos en periodos o en bloques cortos y fijos para maximizar el rendimiento y cambios(“Fundamentos de Scrum | Proyectos Ágiles” n.d.), en el presente proyecto se implementó esta buena práctica.

Control del Proyecto: en este punto al final de cada iteración se le da a conocer al cliente el resultado obtenido para que pueda revisar y realizar los cambios que sean necesarios para el proyecto (Proyectos Ágiles n.d.). En el presente proyecto se implementó esta buena práctica.

Potenciación del equipo: este ítem se basa en comprometer a los desarrolladores en el cumplimiento de los requerimientos, para ello se les otorga la autoridad y recursos necesarios para el cumplimiento de éste (Proyectos Ágiles n.d.). En el presente proyecto se implementó esta buena práctica.

Sistematización: se garantiza la colaboración y comunicación entre el equipo de trabajo y el cliente para que el proyecto culmine de la mejor manera (Proyectos Ágiles

n.d.).En el presente proyecto se implementó esta buena práctica. Como evidencia para esta práctica se implementaron actas de reunión en donde se plantean avances, inquietudes y soluciones de acuerdo al caso.

Timeboxing: se basa en generar el tiempo de entrega máximo de alguna tarea, así se logra tener la fecha de entrega de la misma y no dejarla a la deriva (Proyectos Ágiles n.d.) En el presente proyecto se implementó esta buena práctica.

3.1.3. Artefactos

Product Backlog: es el inventario que contiene cualquier tipo de trabajo que hay que realizar para el proyecto, como por ejemplo: requerimientos, caso de usos, tareas y dependencias; esta labor debe ser realizada por el producto owner, el cual es el contacto con el cliente (Roche 2020). En el presente proyecto se implementó este artefacto.

Sprint backlog: es un elemento que permite trabajar durante la etapa del sprint, estos elementos se componen de tareas pequeñas, las cuales permiten el incremento del rendimiento del software para su terminación. Permite visualizar qué tareas del sprint aún no se han empezado a desarrollar y aquellas que sí y quienes las estan realizando (Roche 2020). En el presente proyecto se implementó este artefacto.

Incremento: es reducir Scrum a una sola tarea, la cual es entregar una pieza de software terminando cada spring al cliente. Es decir, la suma de todas las tareas que se realizaron en el spring casos de uso, historias de usuario, software y cualquier otra actividad que se haya desarrollado del proyecto en el spring (Roche 2020). En el presente proyecto se implementó este artefacto.

Definition of ready: es el documento que da luz verde para definir si un requerimiento esta totalmente definido para proseguir con su parte de desarrollo y así ser envíado al equipo de desarrollo (Roche 2020). En el presente proyecto se implementó este artefacto.

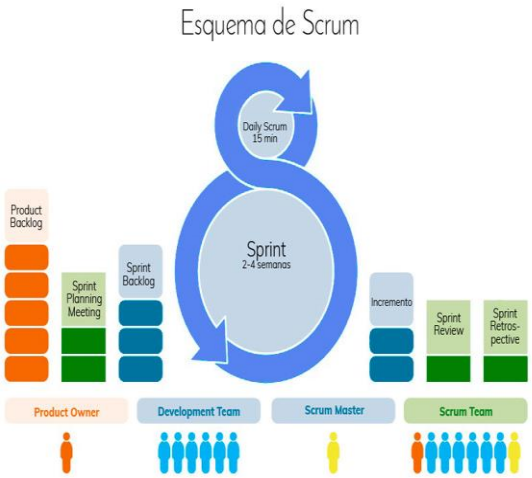
Sprint: es uno de los ciclos o iteraciónes que se realizarán en el proyecto permitiendo tener un ritmo de trabajo ya definido con una fecha prefija; la duración de estos sprint es de 2 semanas (Requena Mesa 2018). Pero, basados en la característica de este proyecto las cuales son que no es un proyecto con cambios frecuentes y un alcance muy amplio, se manejarán solamente tres sprints de 4 semanas cada uno; en el presente proyecto se implementó este artefacto.

3.2. Aplicación de la Metodología Scrum

En la figura 1 se puede observar el esquema de trabajo por el cual se rige Scrum.

Figura 1

Esquema de Scrum



Nota. el anterior gráfico es la representación del esquema de la metodología Scrum(Saiz 2017).

3.2.1. *Actividades*

- Análisis de requerimientos
- Reunión con el cliente
- Estandarización de requerimientos
- Creación de historias de usuario
- Díagramás de caso de uso (Entregable)
- Desarrollo del software
- Revisión de requerimientos o historias de usuario
- Diseño del Diagrama MER
- Definición de herramientas a utilizar
- Desarrollo de seudocódigo del software (Entregable)
- Aplicación de metodologías ágiles
- Investigación de metodologías ágiles que se adecuen al proyecto
- Definición de roles
- Implementación de buenas prácticas
- Documentación de la implementación de la metodología (Entregable)
- Ejecución de pruebas de funcionalidad
- Reunión con el cliente
- Aplicación de pruebas funcionales
- Corrección o retroalimentación del cliente
- Documentación de las pruebas funcionales (Entregable)

3.2.2. Sprint

En este punto se muestran cuáles son las actividades que se realizaron en los tres Sprint mediante Scrum.

- Sprint 1

- Reunión de inicio con el cliente para conocer los requerimientos del establecimiento.
- Identificar las fallas del proceso actual .
- Crear backlog.
- Interacción de desarrollo compuesto por validación de usuarios, registros de clientes y registro de productos.
- Reunión con el product owner: se muestran las funcionalidades del software acordadas (validación de ususuario, registro de usuarios y productos).
- Ajustes del software según recomendaciones del usuario y actualizacion del backlog.

Entregable: en este sprint se entregará las historias de usuario, backlog y las funcionalidades de registro de producto, registro de usuarios, modificación y eliminación.

La Figura 2 se relaciona el módulo de registro de productos.

Figura 2

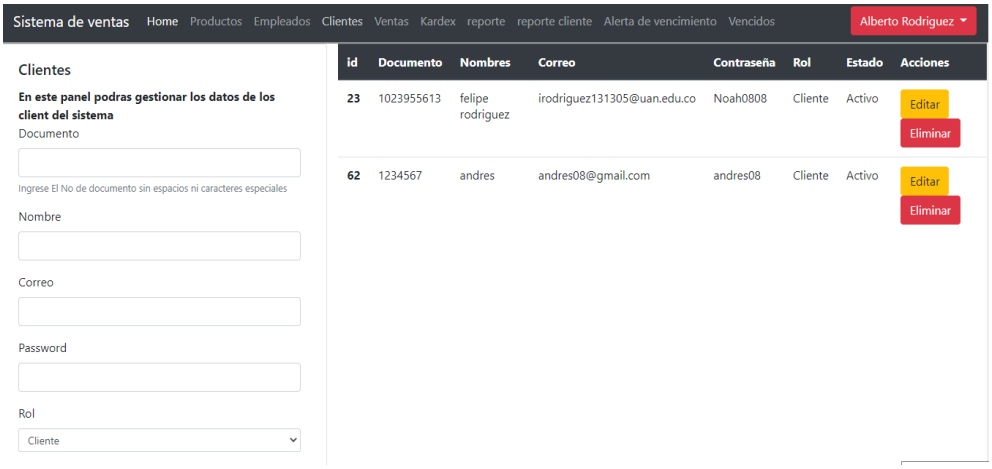
Módulo registro de productos

Fuente: elaboración propia.

La figura 3 muestra el módulo de registro de usuarios

Figura 3

Módulo registro de usuarios



Fuente: elaboración propia.

- Sprint 2

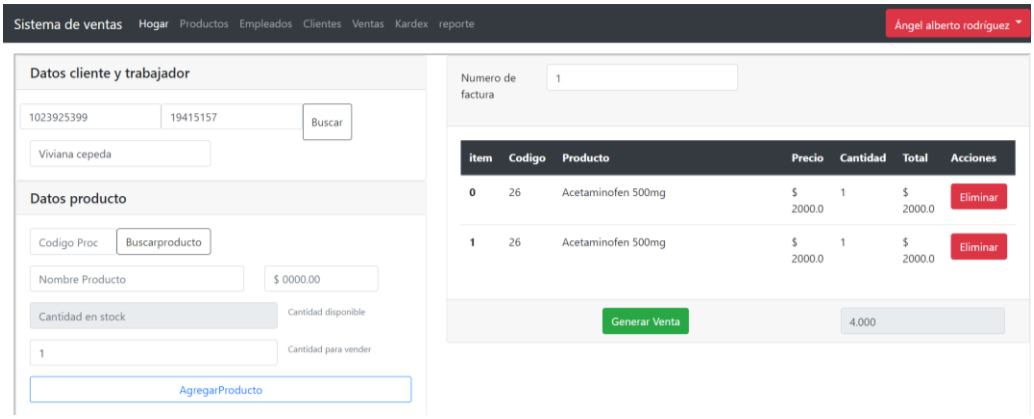
- Interacción de desarrollo compuesto por salida de productos, modificación, eliminación y búsqueda de productos.
- Reunión product owner: se muestran las funcionalidades del software acordadas(salida de productos, modificacion, eliminación y búsqueda de productos).
- Ajustes del software según recomendaciones del usuario.
- Actualización del backlog

Entregable: en este sprint se entregará las historias de usuario, backlog actualizado y las funcionalidades de salida de producto y buscar producto conectada a una base de datos con las anteriores funcionalidades.

En la figura 4 se puede observar el módulo de salida de productos

Figura 4

Módulo de venta



Fuente: elaboración propia.

La Figura 5 se relaciona el módulo de kárdex donde se encuentran los productos registrados y la funcionalidad de buscar un producto en específico.

Figura 5

Módulo registro de kárdex

Fuente: elaboración propia

- Sprint 3

- Interacción de desarrollo compuesto por la funcionalidad de reporte de medicamentos.
- Reunión product owner: se muestran las funcionalidades del software acordadas(reporte de medicamentos).
- Ajustes del software según recomendaciones del usuario.
- Actualización del backlog

Entregable: en este sprint se entregará las historias de usuario, backlog actualizado y la funcionalidad de reporte de los medicamentos según sus ventas.

La Figura 6 se relaciona el módulo de reporte de medicamentos, donde se encuentran: cantidad vendida de cada medicamento y su valor.

Figura 6

Módulo de reporte

Fuente: elaboración propia

4. Desarrollo del Proyecto

A continuación se mostrará la metodología que se utilizó para el desarrollo del proyecto con sus respectivos entregables de acuerdo con sus sprint.

4.1.Levantamiento De Requerimientos

Para lograr el desarrollo de este proyecto se identificaron las necesidades que tiene la droguería Onassis; se realizó una reunión con el administrador donde él reflejó las principales funciones que requiere con el sistema. Con esto se logra identificar los requerimientos funcionales y no funcionales.

4.1.1. Requerimientos Funcionales

En la tabla 2 se puede apreciar los requerimientos solicitados por la droguería Onassis para el desarrollo de este proyecto.

Tabla 2

Requerimientos Funcionales

Requerimiento	Descripción
RQ-01	El sistema debe permitir el registro de usuarios de acuerdo a su rol (Empleado y cliente) con los siguientes datos nombre, documento, correo, password, rol y estado.
RQ-02	El sistema permitirá el acceso de los usuarios permitidos a la aplicación, en este caso el empleado con número de documento y password.
RQ-03	El sistema debe permitir las funcionalidades de actualizar datos de los usuarios y eliminar usuarios.

RQ-04	El sistema permitirá el registro de los productos vigentes en la droguería con los siguientes datos: nombre del producto, descripción, precio y cantidad.
RQ-05	El sistema debe permitir las funcionalidades de actualizar datos de los productos y eliminar productos.
RQ-06	El sistema debe permitir la visualización de los productos con su stock actual y precio. Además, debe permitir la búsqueda de un producto en específico.
RQ-07	El sistema debe permitir la facturación de los productos vendidos.
RQ-08	El sistema debe permitir el reporte de la cantidad de productos vendidos de forma díaria semanal o mensual.

Fuente: elaboración propia

4.1.2. Requerimientos No Funcionales

En la tabla 3 se puede apreciar los requerimientos no funcionales solicitados por la droguería Onassis para el desarrollo de este proyecto.

Tabla 3

Requerimientos No Funcionales

Requerimiento	**Descripción**
RQN-01	El sistema contará con manual de usuario para el uso de la aplicación.
RQN-02	La interfaz con el usuario debe ser de fácil interacción.

Fuente : elaboración propia

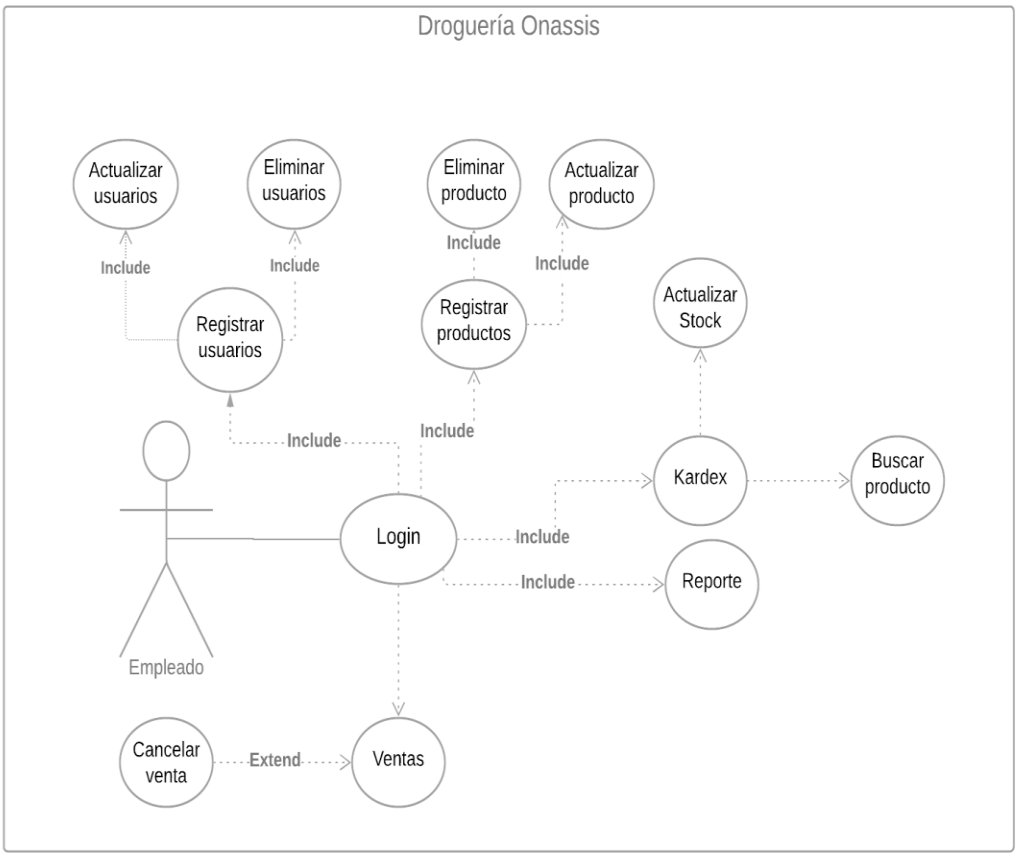
4.2.Modelo conceptual

4.2.1. Diagrama de caso de uso

En la Figura 7 se relaciona el Diagrama de casos de uso del usuario empleado.

Figura 7

Diagrama De Caso De Uso



Fuente: elaboración propia

4.2.2. Casos de uso

De acuerdo con los requerimientos acordados con la droguería Onassis se define el correcto funcionamiento del sistema, por medio de los casos de uso. En la tabla 4 se puede apreciar el caso de uso de inicio de sesión del usuario con rol de empleado.

Tabla 4

Caso de uso inicio de sesión

Indicador: DLP01	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Inicio de sesión	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores.	
Resumen:	Los Trabajadores van a ingresar su número de documento y la contraseña para poder ingresar a las demás funcionalidades de la aplicación.	
Curso Básico Eventos	1. El Trabajador ingresa su número de documento y la contraseña en los campos solicitados y selecciona la opción de ingresar. 2. El sistema verifica las credenciales y lo dirige a la página principal.	
Caminos de Excepción	1. Si los datos ingresados no coinciden con los registrados, el sistema volverá a cargar la página de inicio para el login. 2. Si al tratar de seleccionar la opción de ingresar el trabajador no ingresó usuario y/o contraseña, el sistema volverá a cargar la página de inicio para el login.	
Pre – Condiciones	N/A.	
Post – Condiciones	El sistema lo dirige a la página principal.	
Criterios de Aceptación	El sistema debe validar las credenciales e indicar si son incorrectas, si son correctas debe dirigirlo a la página principal.	

Fuente: elaboración propia

En la tabla 5 se puede apreciar el caso de uso de registro de usuarios.

Tabla 5

Caso de uso registro de usuarios

Indicador: DLP02	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Registro de usuarios	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores ingresarán los datos para registrar un nuevo usuario ya sea con un rol del cliente o trabajador; los datos ingresados son: número de documento, nombre, contraseña, correo, rol y estado	
Curso Básico Eventos	El Trabajador ingresa los datos solicitados por el sistema, los cuales son: nombre, cédula, contraseña, correo, rol y estado Presionará el botón de agregar y el sistema actualizará la tabla de usuarios,mostrando la creación del cliente o empleado en la aplicación .	
Caminos de Excepción	1. Si los datos ingresados no corresponden con el tipo correspondiente, la aplicación no lo dejará registrar; ejemplo :en cédula solo pueden ingresar caracteres numéricos,la aplicación no deja ingresar otro tipo de dato. 2. Si al tratar de seleccionar la opción de agregar no seleccionó todos los datos,la aplicación no dejará agregar al usuario y mostrará un letrero que indica que el campo esta vacío.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de clientes mostrando al nuevo usuario creado.	
Criterios de Aceptación	El sistema debe validar que todos los datos sean diligenciados correctamente para agregar el usuario en la aplicación y actualizarlo en la base de datos.	

Fuente: elaboración propia.

En la tabla 6 se puede apreciar el caso de uso de actualización de datos de usuarios

Tabla 6

Caso de uso Actualización de datos de usuario

Indicador: DLP03	**Indispensable** **Prioridad: Alta**
Nombre de Caso de Uso:	Actualización de datos de usuario
Autor:	Felipe Rodríguez
Fecha:	18-08-2021
Categoría (Visible)	Actores Involucrados: trabajadores
Resumen:	Los trabajadores presionar el botón de editar que se encuentra al lado de cada usuario el cuál cargará la información en los campos requeridos para su actualización.
Curso Básico Eventos	1. El trabajador presionará el botón de editar para cargar los datos del usuario seleccionado. 2. La información se cargará en los campos específicos nombre, documento, correo, contraseña, rol y estado para su modificación. 3. Después del que el trabajador actualice la información presionará el botón de actualizar en donde se volverá a cargar la lista de usuarios con los datos actualizados.
Caminos de Excepción	1. Si los datos ingresados no corresponden con el tipo correspondiente, la aplicación no lo dejará que la actualización sea efectiva; ejemplo: en cédula solo pueden ingresar caracteres numéricos, la aplicación no deja ingresar otro tipo de dato. 2. Si al tratar de seleccionar la opción de actualizar no seleccionó todos los datos, la aplicación no dejará actualizar al usuario y mostrará un letrero que indica que el campo esta vacío.
Pre – Condiciones	N/A
Post – Condiciones	El sistema cargará la tabla de clientes mostrando al usuario actualizadó.
Criterios de Aceptación	El sistema debe validar que todos los datos sean diligenciados correctamente para actualizar el usuario en la aplicación y actualizarlo en la base de datos.

Fuente: construción propia

En la tabla 7 se puede apreciar el caso de uso de eliminar usuarios.

Tabla 7

Caso de uso eliminar usuario

Indicador: DLP04	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Eliminar usuario	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Presionar el botón de eliminar que se encuentra al lado de cada usuario, el cuál eliminará el perfil del usuario de la aplicación 1. El Trabajador presionaráel botón de eliminar el cual borrará al usuario de la aplicación y base de datos. 2. La información se cargará en la página mostrando que el usuario ha sido eliminado de la lista	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción		
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de usuarios mostrando al que el usuario se ha eliminado.	
Criterios de Aceptación	El sistema debe validar que el usuario exista para eliminarlo; en seguida lo eliminará de la aplicación y base de datos; posteriormente carga la lista de usuarios actualizada.	

Fuente: elaboración propia

En la tabla 8 se puede apreciar el caso de uso de registro de productos

Tabla 8

Caso de uso registro de productos

Indicador: DLP05	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Registro de productos	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores ingresarán los datos para registrar el nuevo producto con los datos solicitados.	
Curso Básico Eventos	1. El trabajador ingresa los datos solicitados por el sistema,como: nombre del producto, categoria, cantidad (numerico), lote, fecha de vencimiento y precio (numerico). 2. Presionará el botón de agregar y el sistema actualizará la tabla de productos mostrando la creación del producto.	
Caminos Alternativos	2. Si los datos ingresados no corresponden con el tipo correspondiente, la aplicación no lo dejará registrar; por ejemplo: en precio y cantidad solo pueden ingresar caracteres numéricos, la aplicación no deja ingresar otro tipo de dato.	
Caminos de Excepción	2. Si al tratar de seleccionar la opción de agregar no seleccionó todos los datos, la aplicación no dejará agregar el producto y mostrará un mensaje que indica que el campo esta vacío.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de productos, mostrando al nuevo producto creado.	
Criterios de Aceptación	El sistema debe validar que todos los datos sean diligenciados correctamente para agregar el producto en la aplicación y actualizarlo en la base de datos	

Fuente: elaboración propia

En la tabla 9 se puede apreciar el caso de uso de actualización de productos

Tabla 9

Caso de uso actualización de productos

Indicador: DLP06	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Actualización de productos	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores presionan el botón de cargar que se encuentra al lado de cada producto, el cuál cargará la información en los campos requeridos para su actualización	
Curso Básico Eventos	1. El Trabajador presionará el botón de cargar para importar los datos del producto seleccionado. 2. La información se cargará en los campos específicos: nombre, descripción, precio y cantidad para su modificación. 3. Después del que el trabajador actualice la información, presionará el botón de actualizar, en donde se volverá a cargar la lista de productos con los datos actualizados.	
Caminos Alternativos		
Caminos de Excepción	1. Si los datos ingresados no corresponden con el tipo correspondiente, la aplicación no lo dejará que la actualización sea efectiva; por ejemplo: en precio y cantidad solo pueden ingresar caracteres numéricos, la aplicación no deja ingresar otro tipo de dato 2. Si al tratar de seleccionar la opción de actualizar no seleccionó todos los datos, la aplicación no dejará actualizar al usuario y mostrará un mensaje que indica que el campo esta vacío.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de productos mostrando el producto actualizado.	
Criterios de Aceptación	El sistema debe validar que todos los datos sean diligenciados correctamente para actualizar el producto en la aplicación y actualizarlo en la base de datos	

Fuente: elaboración propia

En la tabla 10 se puede apreciar el caso de uso de eliminación de productos

Tabla 10

Caso de uso eliminación de productos

Indicador: DLP07	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Eliminar Producto	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores presionan el botón de eliminar que se encuentra al lado de cada producto el cuál eliminará el artículo de la aplicación, al realizar este proceso le mostrara un mensaje de alerta solicitando la confirmacion del proceso que va realizar 1. El Trabajador presionara el botón de eliminar el cual borrara el producto de la aplicación y base de datos. 2. La información se cargará en la página mostrando que el producto ha sido eliminado de la lista	
Curso Básico Eventos		
Caminos de Excepción		
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de productos mostrando que el producto ya no esta registrado.	
Criterios de Aceptación	El sistema debe validar que el producto exista para eliminarlo en seguida; lo eliminará de la aplicación y base de datos para cargar la lista de productos actualizada.	

Fuente: elaboración propia

En la tabla 11 se puede apreciar el caso de uso de buscar producto

Tabla 11

Caso de uso buscar producto

Indicador: DLP08	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Buscar Producto	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores ingresarán en la opción de Kárdex, en donde se encuentran listados todos los productos registrados y detalles en donde podrán buscar algún producto en específico 1. El Trabajador ingresará a la opción de Kárdex donde se listarán todos los productos registrados. 2. En el buscador escribir el nombre o código del producto que desea buscar en específico. 3. Presionará el botón de buscar, en donde enseguida se actualizará la lista con el producto que se buscó.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción	En caso que el usuario ingrese el nombre o código de un producto no registrado, el sistema mostrará un mensaje indicando que ese producto no se encuentra registrado y lo redireccionará nuevamente a realizar la búsqueda.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará la tabla de productos mostrando el producto o productos que coinciden con la búsqueda.	
Criterios de Aceptación	El sistema debe validar que el producto exista para así listarlo y el usuario lo pueda observar.	

Fuente: elaboración propia

En la tabla 12 se puede apreciar el caso de uso de buscar cliente

Tabla 12

Caso de uso buscar cliente

Indicador: DLP09	**Indispensable**	**Prioridad: Media**
Nombre de Caso de Uso:	Buscar cliente venta	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores ingresarán en la opción de ventas, en donde ingresarán el documento del cliente para validar si se encuentra registrado en el sistema.	
Curso Básico Eventos	1. El Trabajador ingresará a la opción de ventas donde se ubicará en documento cliente. 2. En el buscador deberá escribir el documento del cliente sin puntos ni comás. 3. Presionará el botón de buscar, en donde el sistema validará si el cliente se encuentra registrado y cargará su documento y nombre.	
Caminos Alternativos		
Caminos de Excepción	En caso que el usuario ingrese el documento del cliente y no se encuentre registrado, la aplicación lo redireccionará a la opción de registrar clientes en donde lo registrará en el aplicativo.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará los datos del cliente, los como: cédula y nombre del cliente.	
Criterios de Aceptación	El sistema debe validar que el cliente exista para así poder continuar con la venta.	

Fuente: elaboración propia

En la tabla 13 se puede apreciar el caso de uso de buscar empleado

Tabla 13

Caso de uso buscar empleado

Indicador: DLP09	**Indispensable**	**Prioridad: Media**
Nombre de Caso de Uso:	Buscar Empleado	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los Trabajadores ingresarán en la opción de ventas, en donde ingresarán el documento del empleado para validar si se encuentra registrado en el sistema.	
	1. El Trabajador ingresará a la opción de ventas donde se ubicará en documento empleado 2. En el buscador deberá escribir el documento del empleado sin puntos ni comás 3. Presionará el botón de buscar en donde el sistema validará si el empleado se encuentra registrado y cargará su documento.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción	En caso que el usuario ingrese el documento del empleado y ingrese un documento no valido el sistema mostrará un mensaje indicando el documento del empleado ingresado no se encuentra registrado	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará los datos del empleado los cuales son cédula.	
Criterios de Aceptación	El sistema debe validar que el empleado exista para así poder continuar con la venta.	

Fuente: construccion propia.

En la tabla 14 se puede apreciar el caso de uso de buscar producto venta

Tabla 14

Caso de uso buscar producto venta

Indicador: DLP09	**Indispensable**	**Prioridad: Medía**
Nombre de Caso de Uso:	Buscar producto venta	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: Trabajadores	
Resumen:	Los trabajadores ingresarán en la opción de ventas, en donde ingresarán el código del producto que se agregara a la venta. 1. El trabajador ingresará a la opción de ventas donde se ubicará en código producto. 2. En el buscador deberá escribir el código del producto sin puntos, ni comás. 3. Presionará el botón de buscar, en donde el sistema validará si el Producto se encuentra registrado y cargará su código, nombre, cantidad disponible y precio.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción	En caso que el usuario ingrese el código del producto e ingrese un código no existente, el sistema le indicará que ese producto no se encuentra registrado.	
Pre – Condiciones	N/A	
Post – Condiciones	El sistema cargará los datos del producto los cuales son: nombre, código, precio, cantidad disponible.	
Criterios de Aceptación	El sistema debe validar que el producto exista para así poder continuar con la venta.	

Fuente: consrucción propia

En la tabla 15 se puede apreciar el caso de uso de agregar producto venta

Tabla 15

Caso de uso agregar producto venta

Indicador: DLP11	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Buscar Agregar producto	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: trabajadores	
Resumen:	Los trabajadores ingresarán en la opción de ventas, en donde se dirigirán a la opción de agregar producto. 1. El trabajador ingresará a la opción de ventas donde se ubicará en la opción de agregar producto. 2. Oprimirá la opción de agregar producto en donde instantáneamente se actualizará la lista de productos.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción	En caso que el usuario oprima la opción de agregar producto sin que ningún campo este diligenciado el sistema le indicará que diligencié los campos faltantes.	
Pre – Condiciones	Antes de realizar esta función se debe realizar la búsqueda de un producto primero.	
Post – Condiciones	El sistema cargará los datos del producto los cuales son: ítem, nombre, código, precio, cantidad de venta y la opción de eliminar producto.	
Criterios de Aceptación	El sistema debe validar que el producto exista para así poder agregarlo a la lista de ventas.	

Fuente: elaboración propia

En la tabla 16 se puede apreciar el caso de uso de generar venta

Tabla 16

Caso de uso generar venta

Indicador: DLP12	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Generar venta	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: trabajadores	
Resumen:	Los trabajadores ingresarán en la opción de ventas, en donde se dirigirán a la opción de Generar Venta. 1. El trabajador ingresará a la opción de ventas donde se ubicará en la opción de Generar venta. 2. Oprimirá la opción de generar venta en donde instantáneamente se generará la venta y se imprima la factura.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción		
Pre – Condiciones	Antes de realizar esta función se debe realizar la función de agregar producto para poder habilitar la función.	
Post – Condiciones	El sistema generará la venta y se imprimirá la factura	
Criterios de Aceptación	El sistema debe generar la venta con los productos agregados y de ser el caso se imprima la factura.	

Fuente: elaboración propia

En la tabla 17 se puede apreciar el caso de uso de reporte de medicamentos vendidos

Tabla 17

Caso de uso reporte de medicamentos vendidos

Indicador: DLP13	**Indispensable**	**Prioridad: Alta**
Nombre de Caso de Uso:	Generar reporte de medicamentos vendidos	
Autor:	Felipe Rodríguez	
Fecha:	18-08-2021	
Categoría (Visible)	Actores Involucrados: trabajadores	
Resumen:	Los trabajadores ingresarán en la opción de reporte, diligenciará las fechas en donde quiere el reporte. 1. El trabajador ingresará a la opción de reporte donde se ubicará en la opción de fechas ingresando el rango de fechas del reporte. 2. Oprimirá la opción de reporte en donde instantáneamente se generará la lista de medicamentos vendidos su cantidad un subtotal por cada medicamento y el total de todas las ventas.	
Curso Básico Eventos		
Caminos Alternativos		
Caminos de Excepción		
Pre – Condiciones	Antes de realizar esta función se debe haber realizado una venta para obtener estos reportes.	
Post – Condiciones	El sistema generará el reporte del rango de fechas escogido.	
Criterios de Aceptación	El sistema debe generar el reporte en el rango escogido mostrando detalladamente los medicamentos y sus cantidades despachadas.	

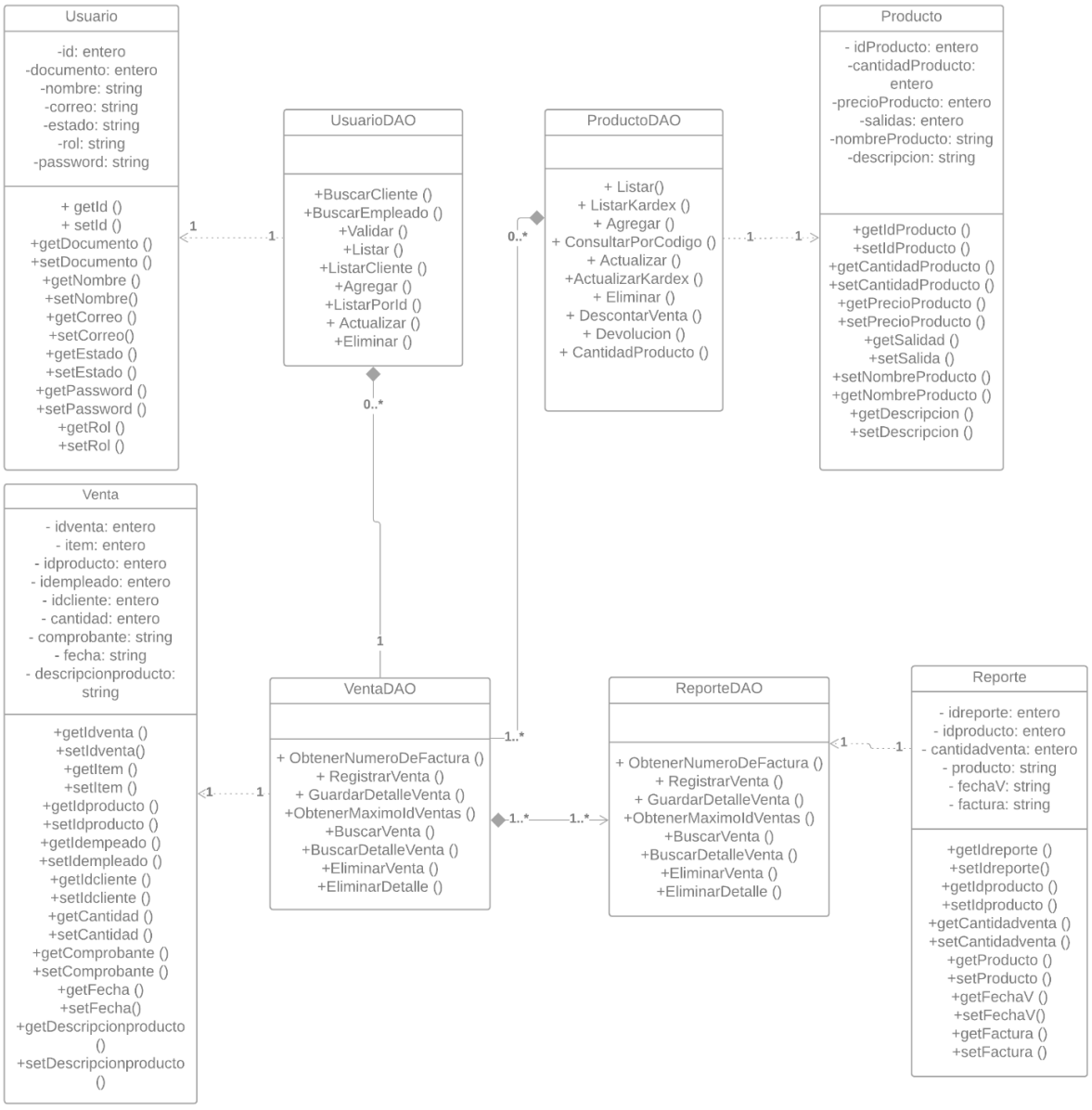
Fuente: elaboración propia

4.2.3. Diagrama de clases

En la Figura 8 se relaciona el Diagrama de clases.

Figura 8

Diagrama de clases



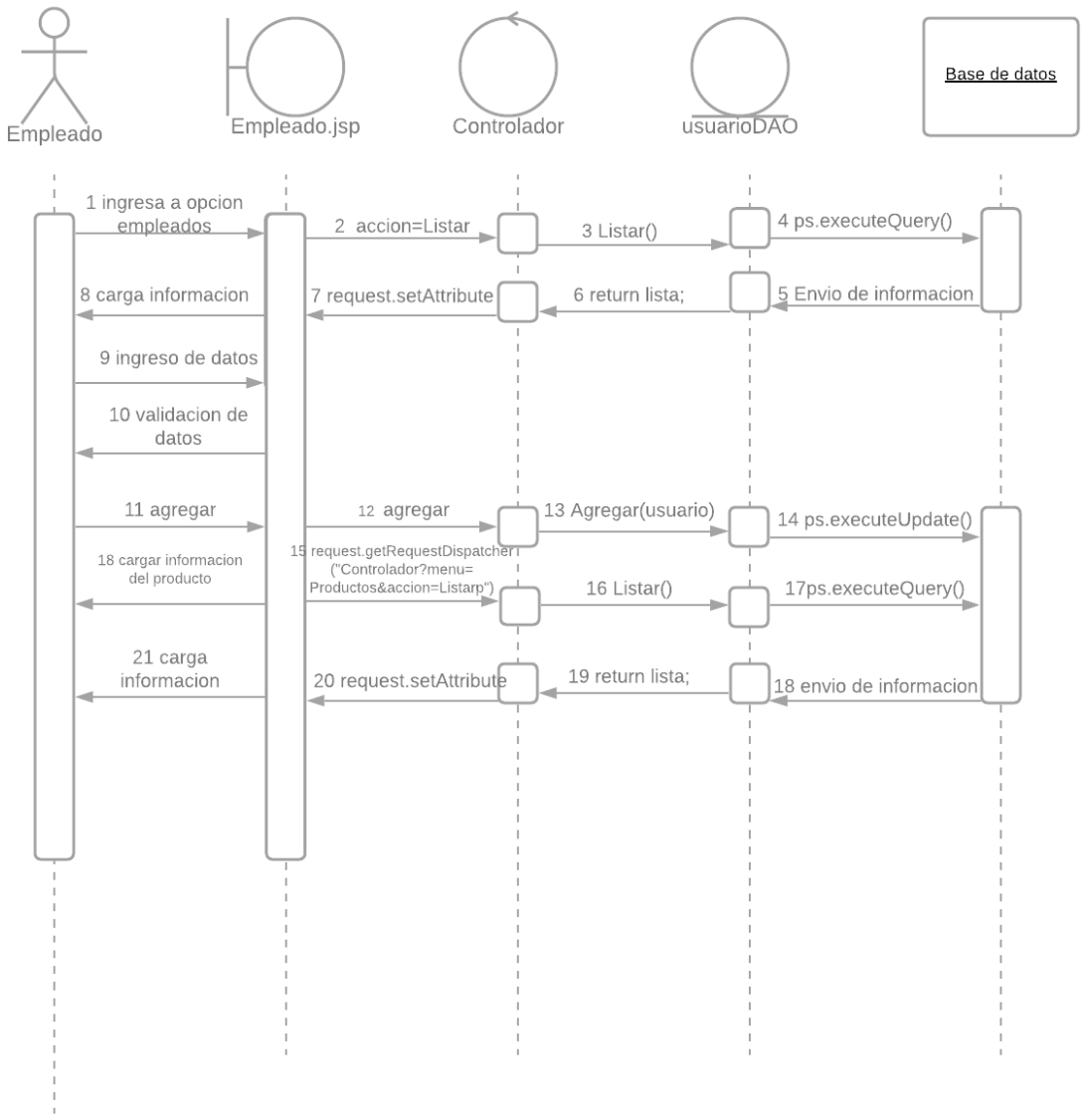
Fuente: elaboración propia

4.2.4. Diagrama de secuencia

En la Figura 9 se relaciona el Diagrama de secuencia de resgistro de empleado.

Figura 9

Diagrama de secuencia registro empleado.

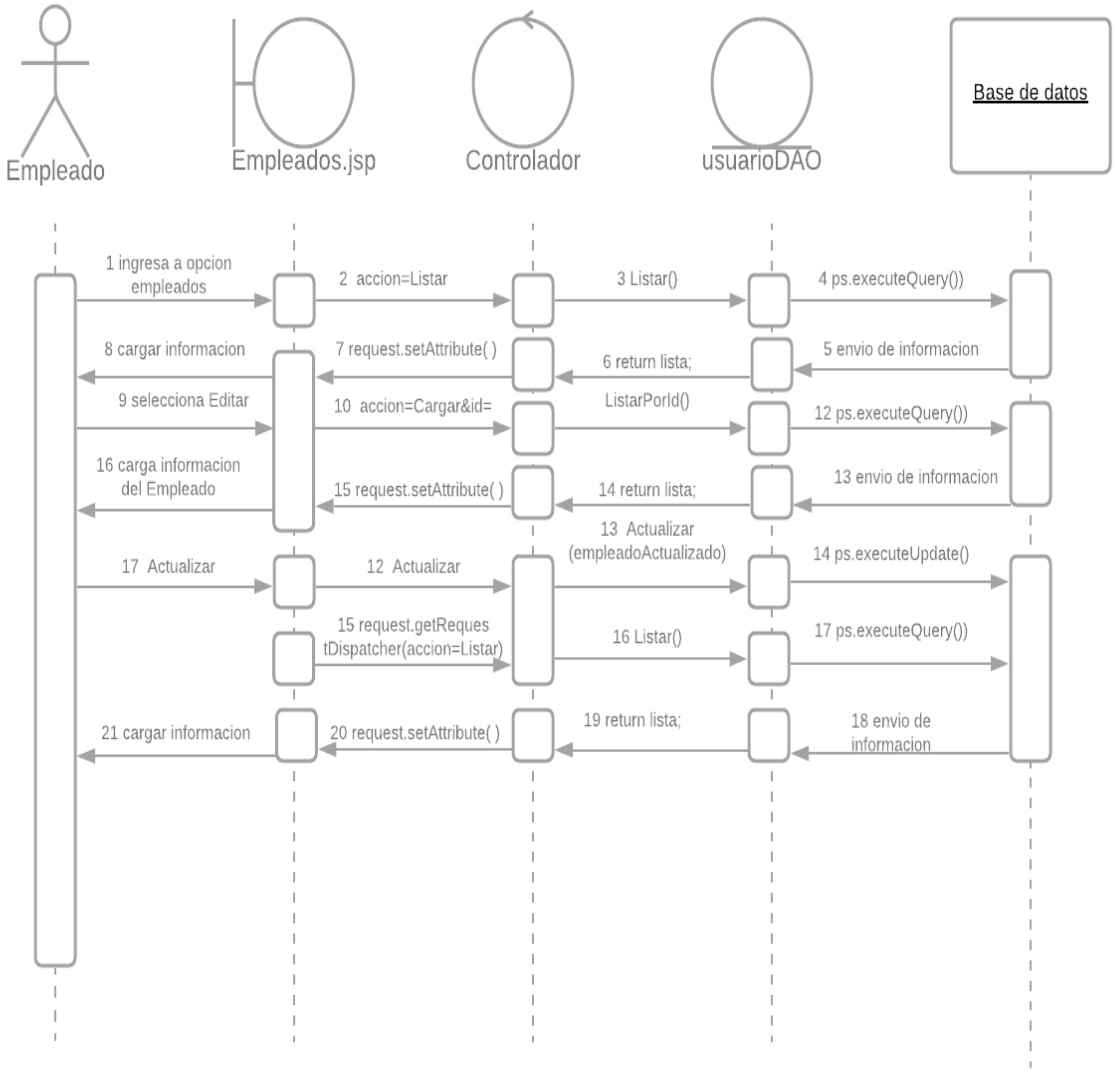


Fuente: elaboración propia

En la Figura 10 se relaciona el Diagrama de secuencia de actualización de empleado.

Figura 10

Diagrama de secuencia actualización empleado

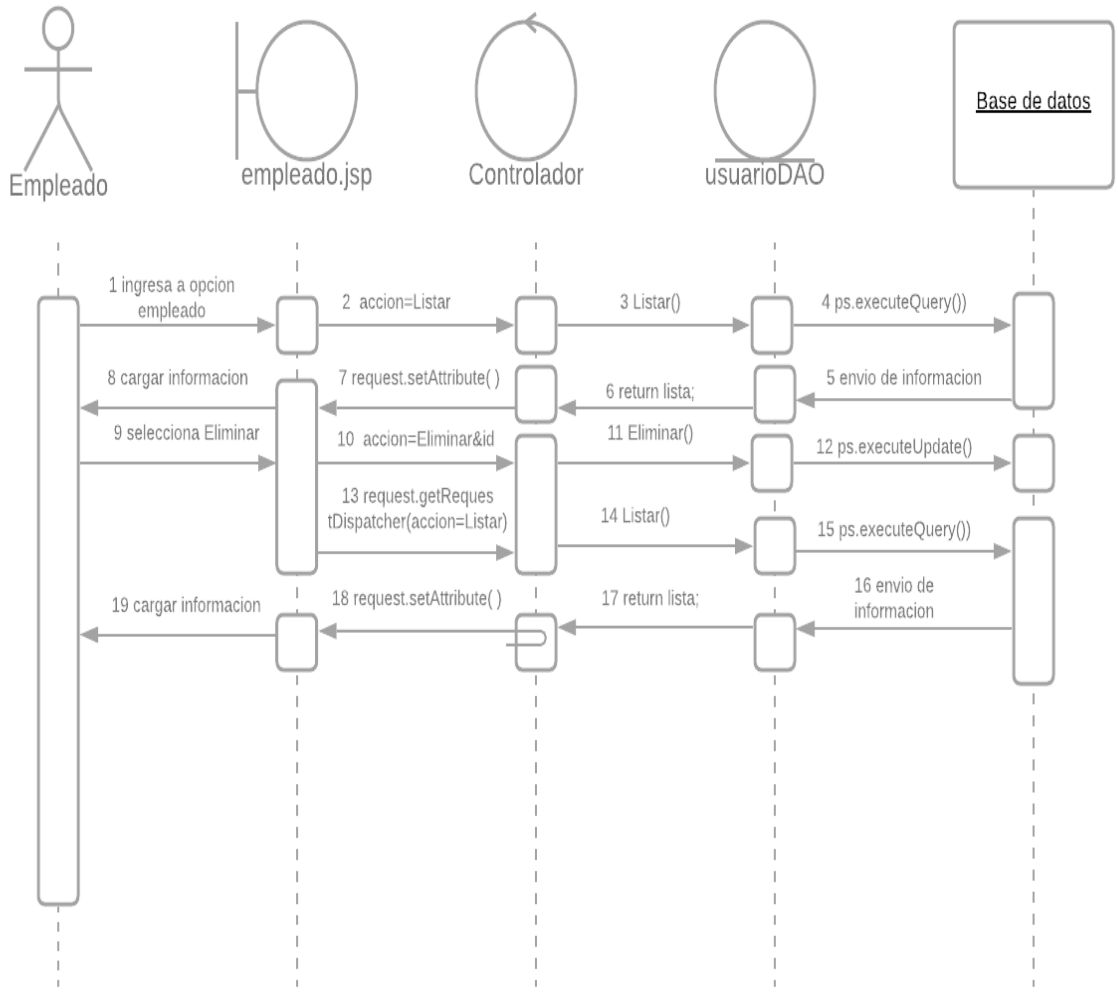


Fuente: elaboración propia

En la Figura 11 se relaciona el Diagrama de secuencia de eliminación de empleado.

Figura 11

Diagrama de secuencia eliminación empleado

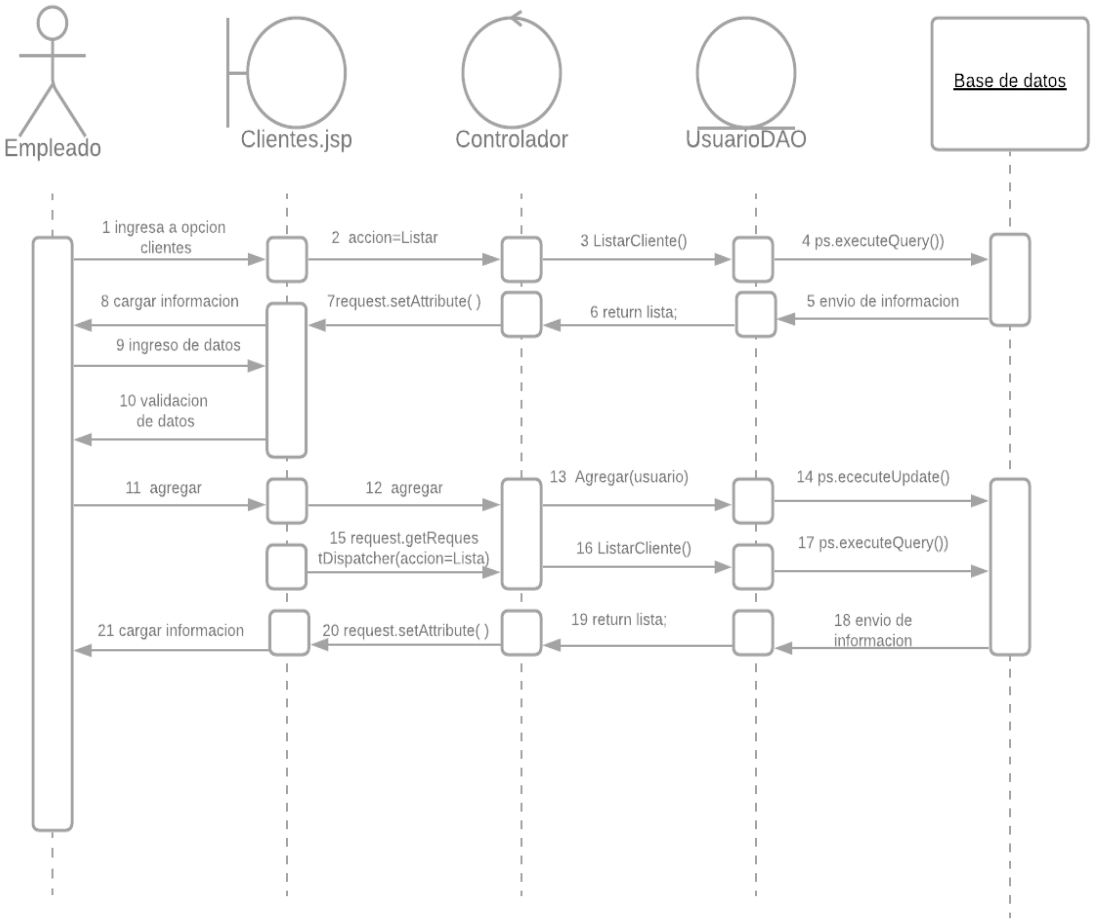


Fuente: elaboración propia.

En la Figura 12 se relaciona el Diagrama de secuencia de resgistro de cliente.

Figura 12

Diagrama de secuencia registro cliente

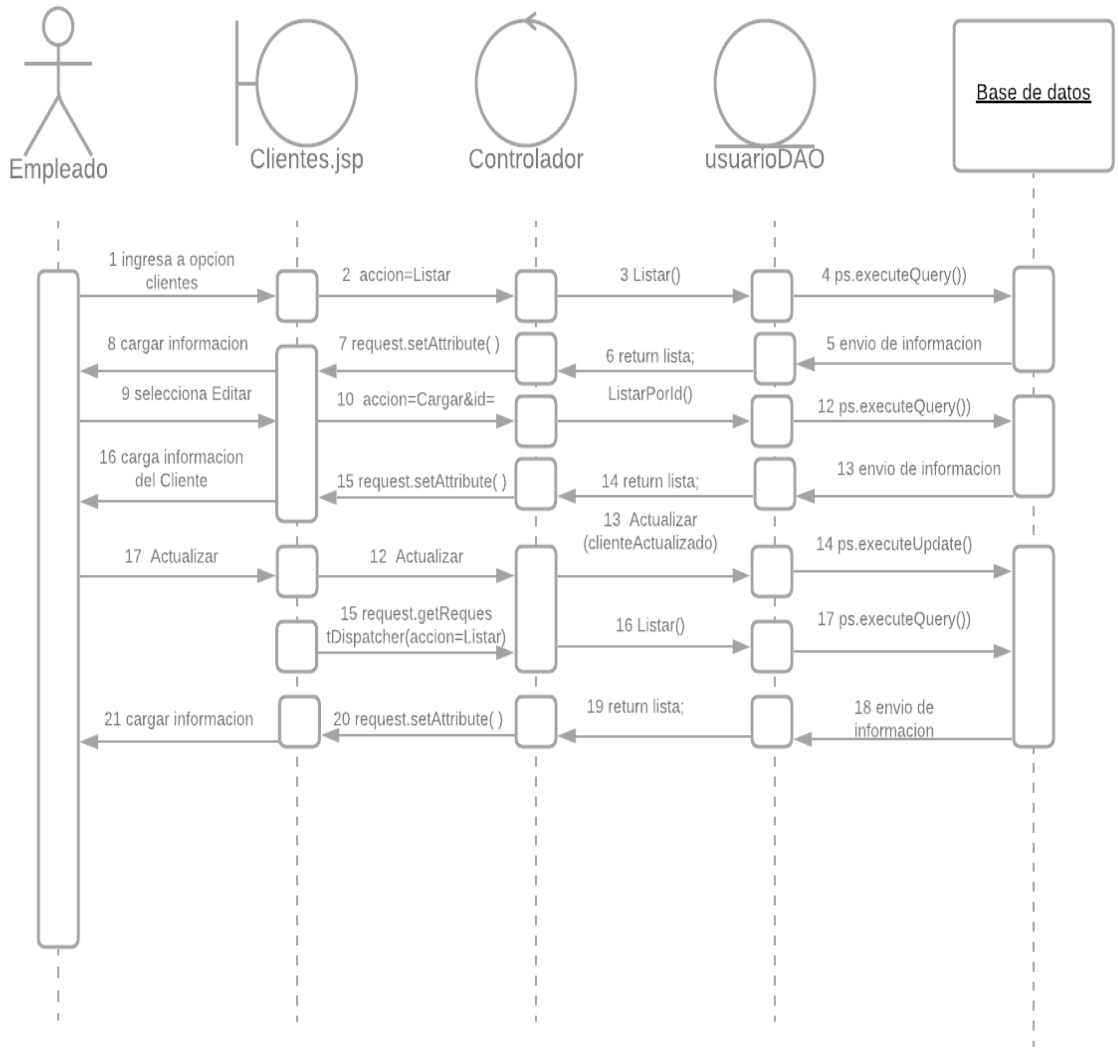


Fuente: elaboración propia.

En la Figura 13 se relaciona el Diagrama de secuencia de actualización de cliente.

Figura 13

Diagrama de secuencia actualización cliente

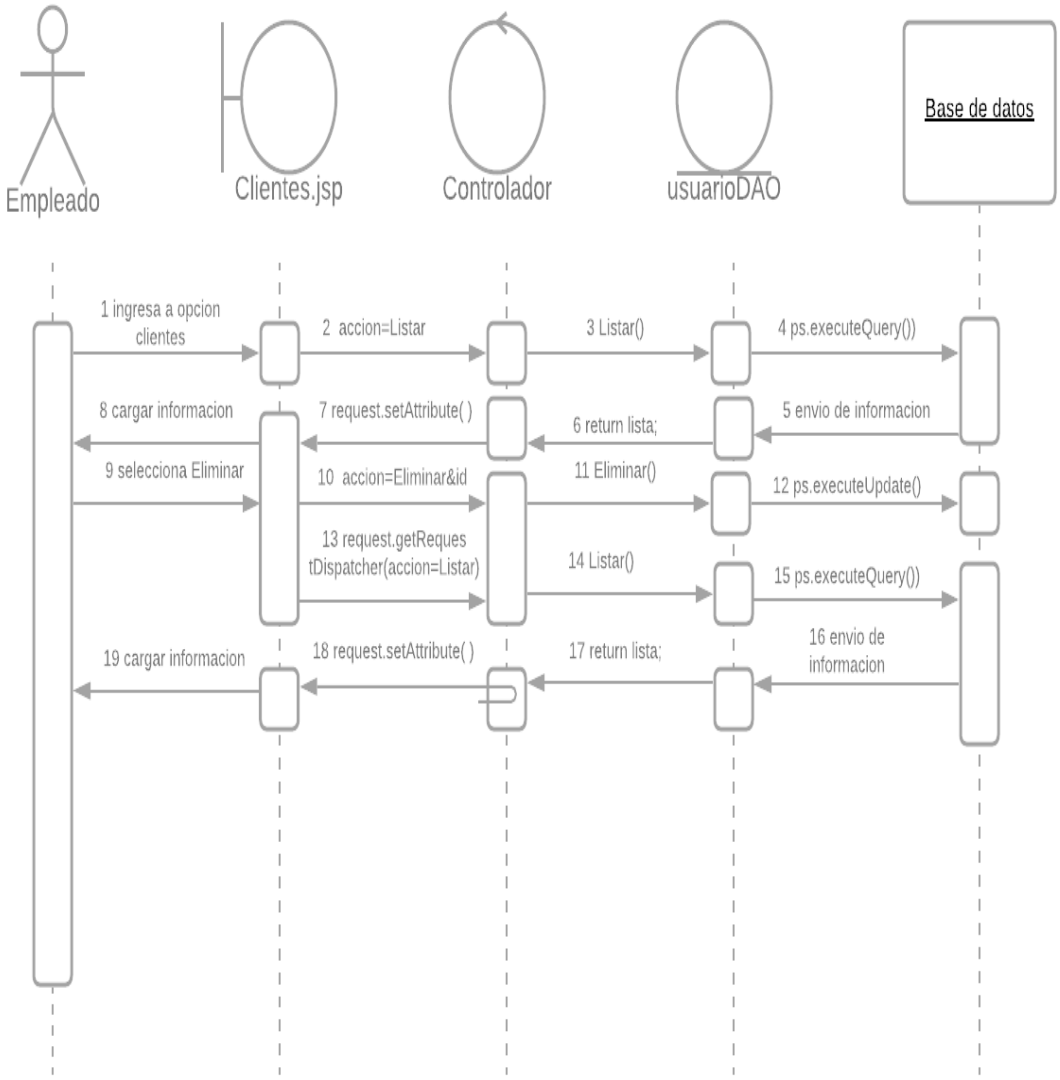


Fuente: elaboración propia.

En la Figura 14 se relaciona el Diagrama de secuencia de eliminación de cliente.

Figura 14

Diagrama de secuencia eliminacion cliente

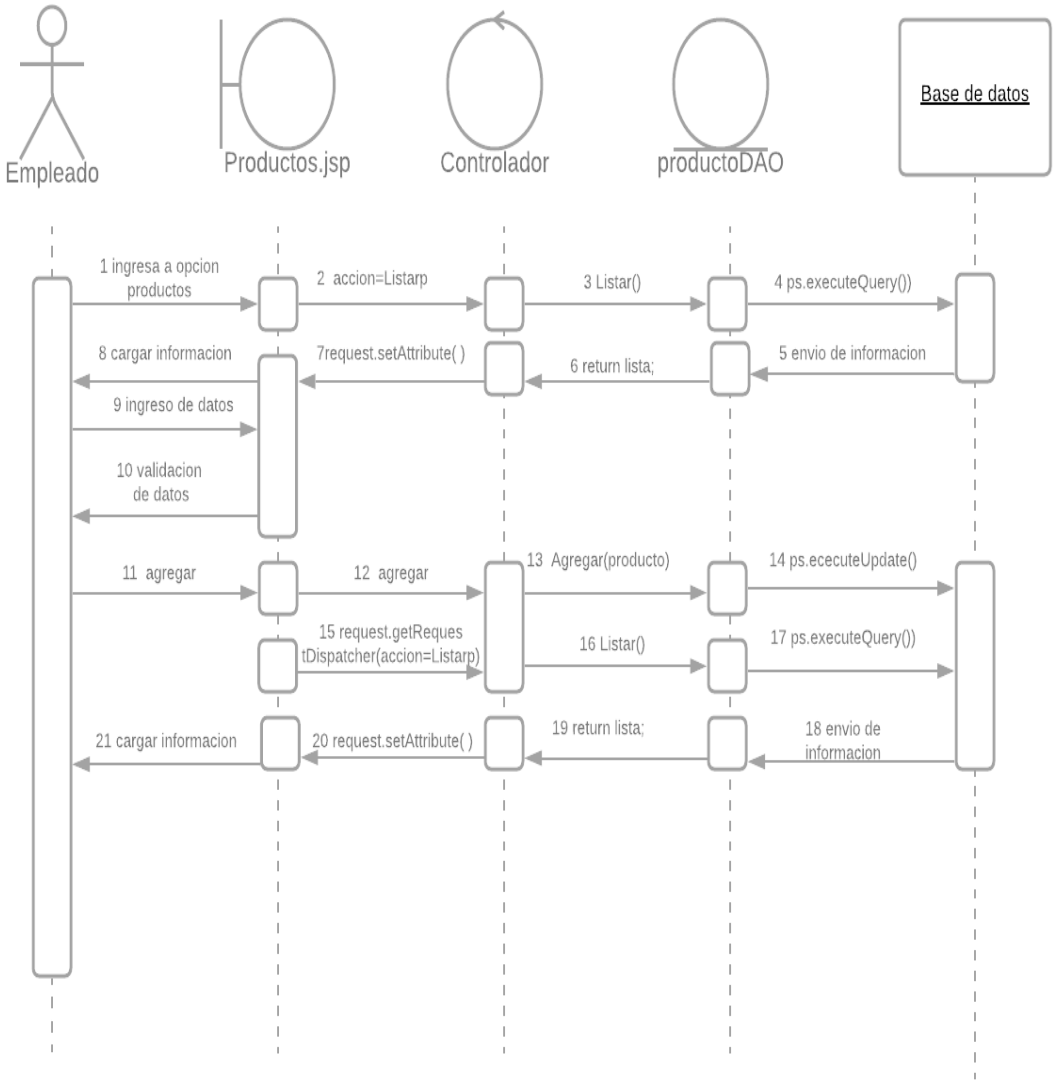


Fuente: elaboración propia.

En la Figura 15 se relaciona el Diagrama de secuencia de registro de producto.

Figura 15

Diagrama de secuencia registro producto

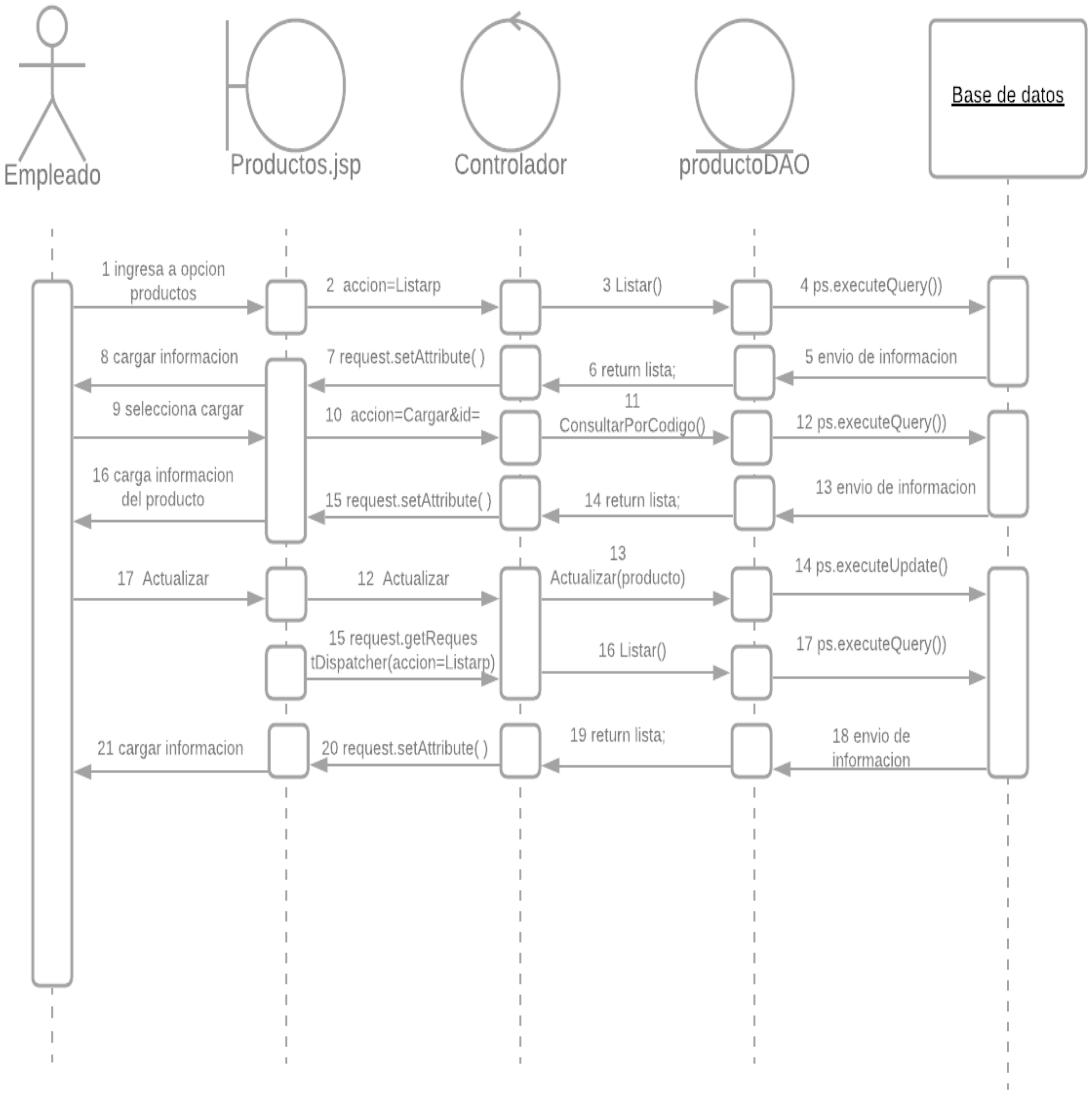


Fuente: elaboración propia

En la Figura 16 se relaciona el Diagrama de secuencia de actualización de producto.

Figura 16

Diagrama de secuencia actualización producto

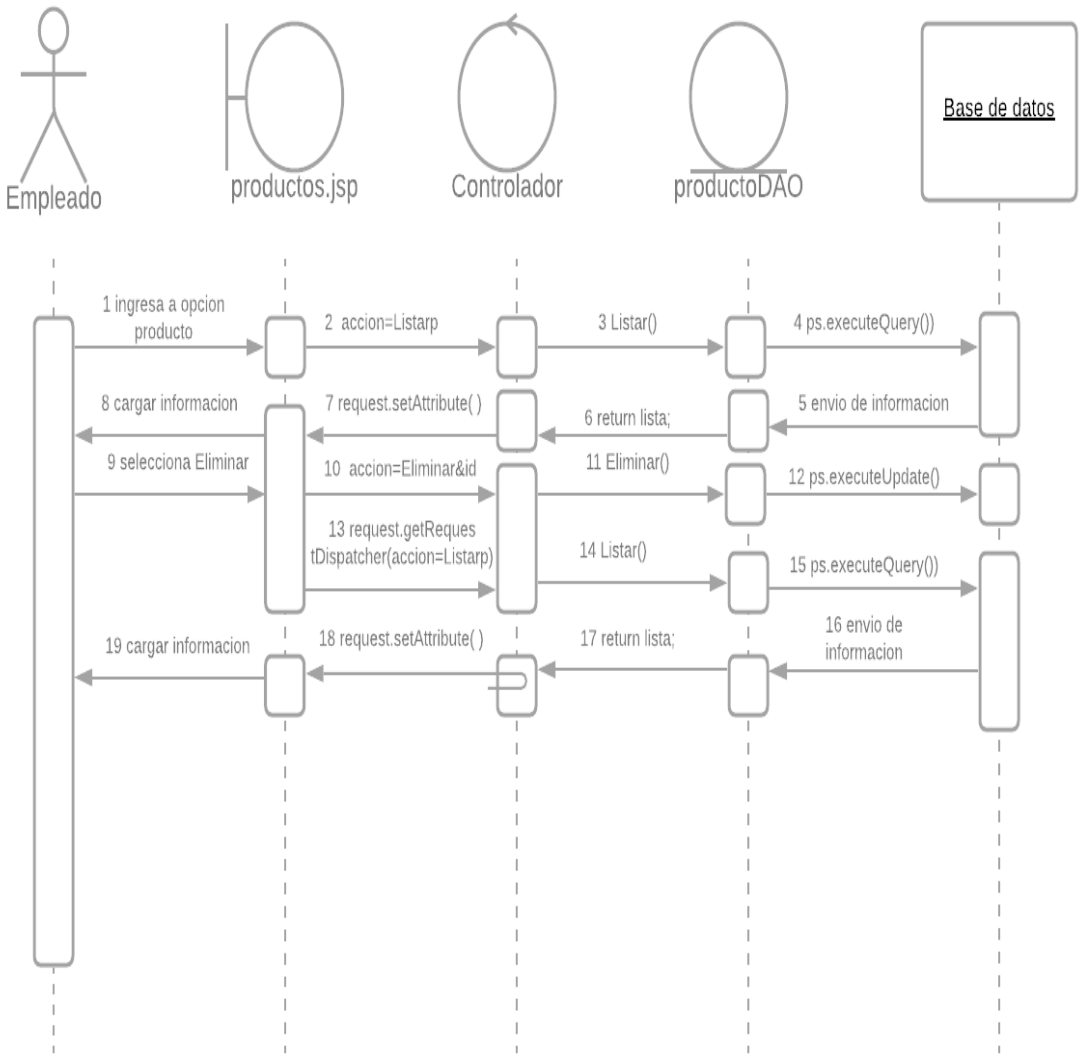


Fuente: elaboración propia

En la Figura 17 se relaciona el Diagrama de secuencia de eliminación de producto.

Figura 17

Diagrama de secuencia eliminación producto

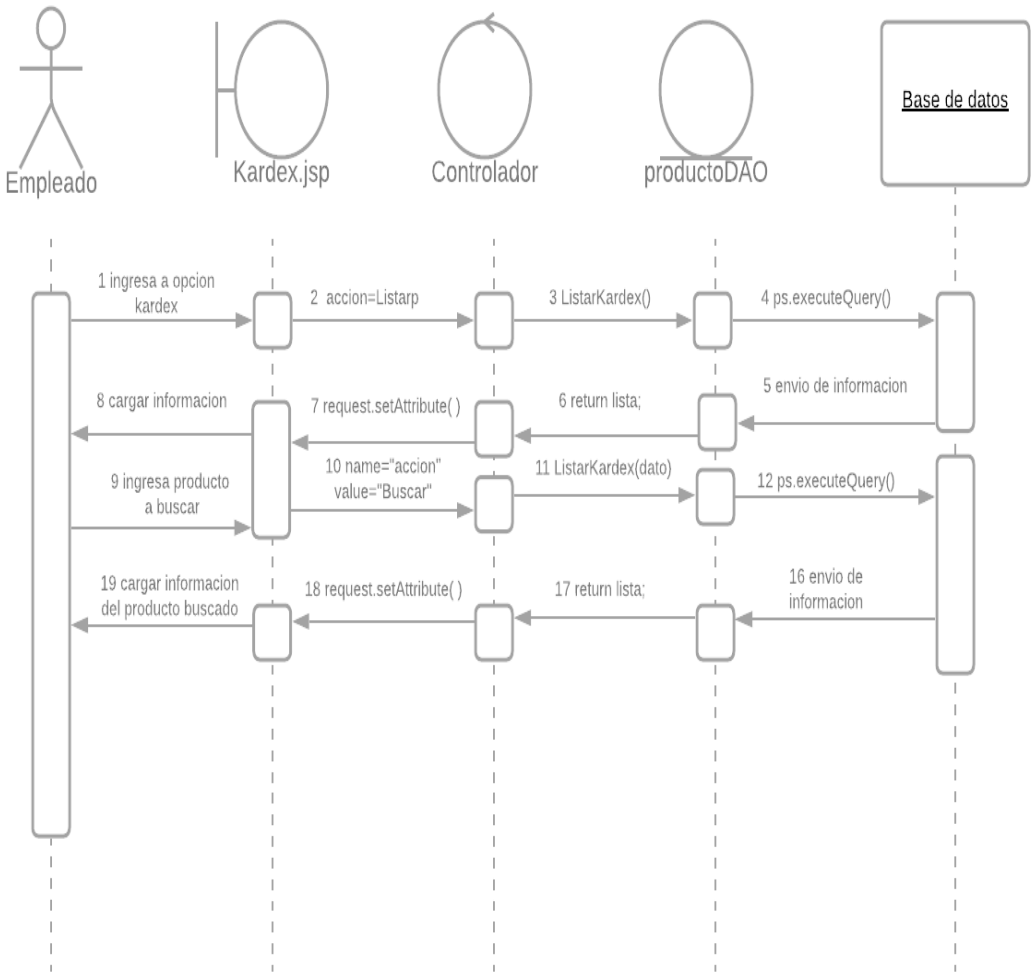


Fuente: elaboración propia

En la Figura 18 se relaciona el diagrama de secuencia de búsqueda de producto en el kárdex.

Figura 18

Diagrama de secuencia busqueda de producto

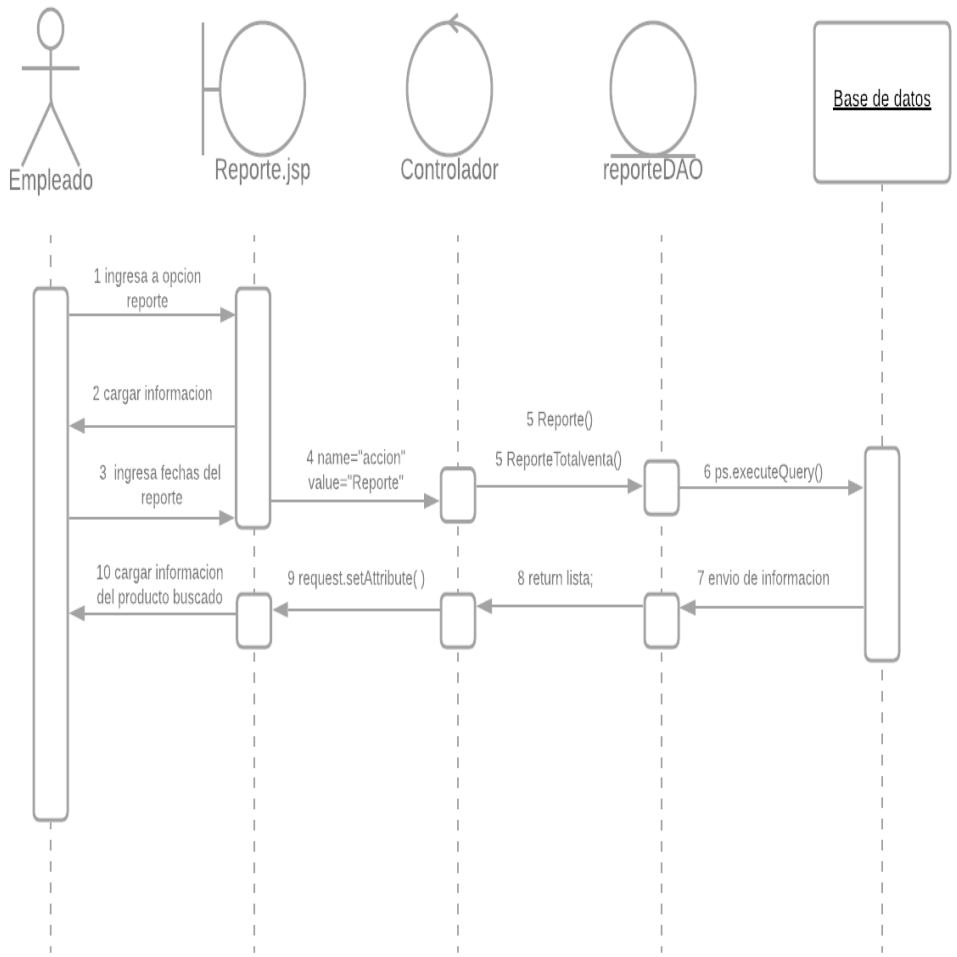


Fuente: elaboración propia

En la Figura 19 se relaciona el Diagrama de secuencia de reporte de venta de medicamentos.

Figura 19

Diagrama de secuencia reporte de venta

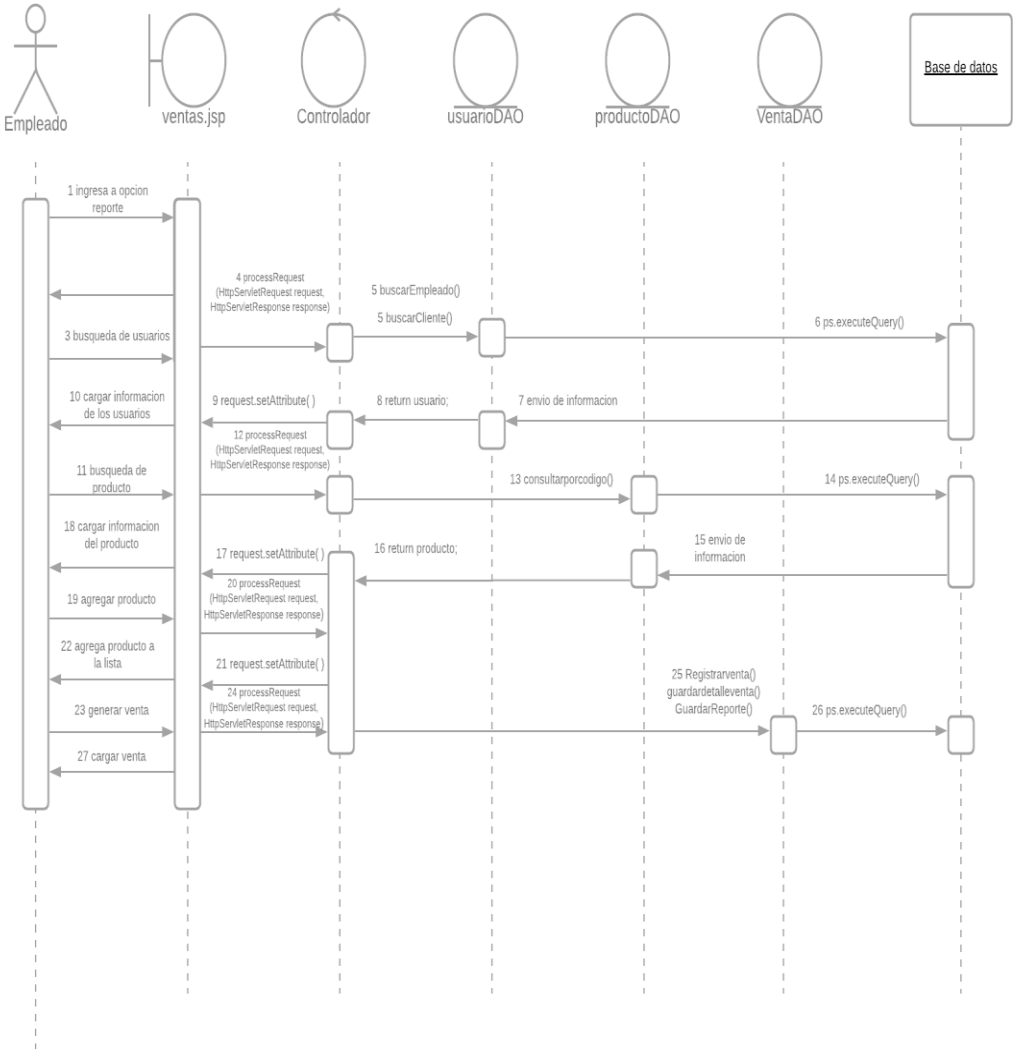


Fuente : elaboración propia

En la Figura 20 se relaciona el Diagrama de secuencia de generar venta.

Figura 20

Diagrama de secuencia venta



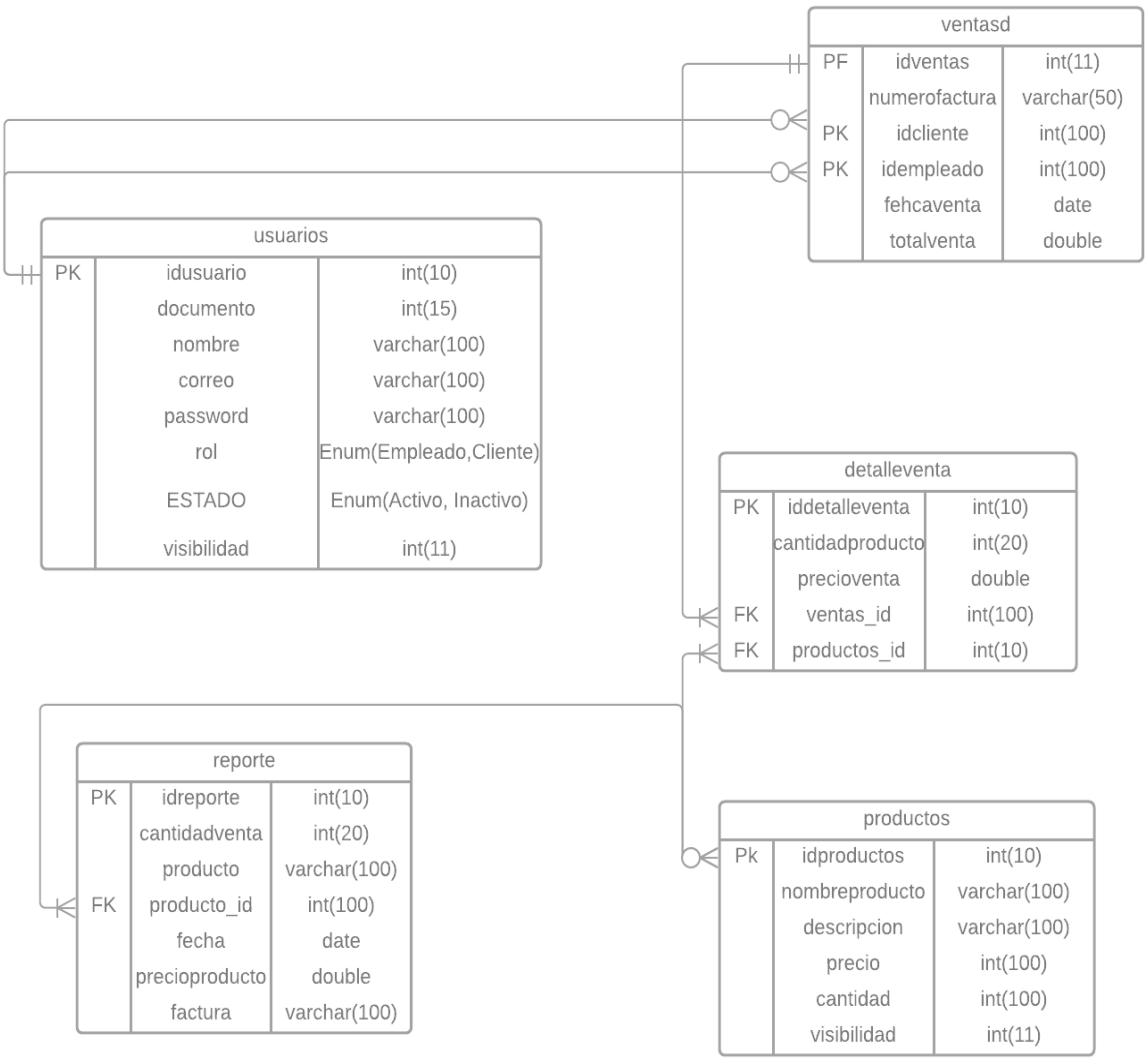
Fuente: elaboración propia

4.2.5. Diagrama MER

En la Figura 21 se relaciona el Diagrama modelo entidad relación (mer) .

Figura 21

Diagrama MER



Fuente: elaboración propia

4.3. Desarrollo Sprint Uno

Para el correcto desarrollo de este proyecto se realizaron 3 sprint, cada uno de 4 semanas, dividiendo en cada uno de los sprints entregas puntuales o avances del software que el cliente en este caso la droguería Onassis, representada por su administrador Ángel Alberto Rodríguez, quien revisó y aprobó el resultado de los entregables del proyecto. El siguiente es el desarrollo del sprint número uno.

4.3.1. Reunión inicial

Se realizó una reunión con el administrador de la droguería Onassis, en la cual se plasmaron los siguientes requerimientos:

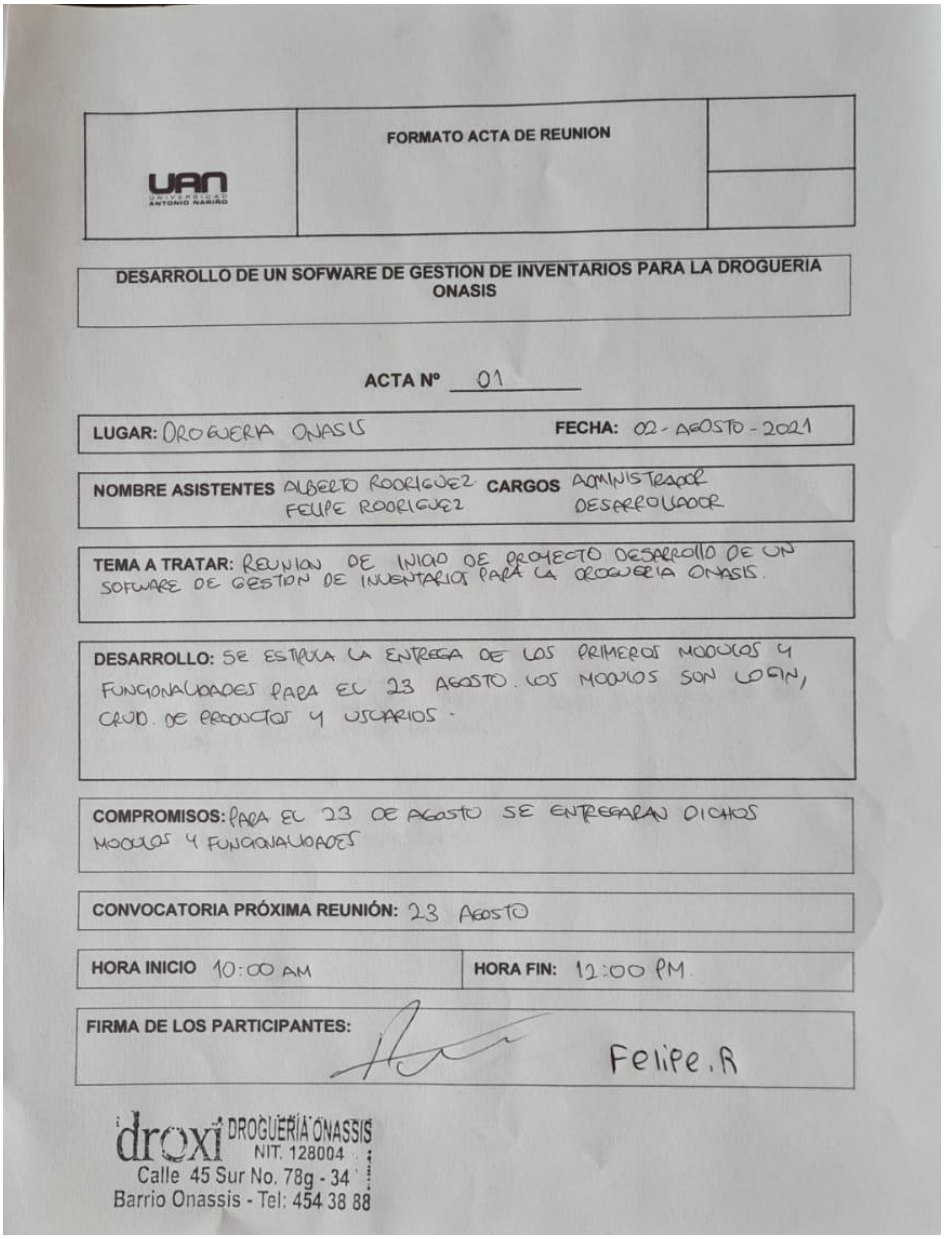
- Registro, actualización y eliminación de usuarios de acuerdo a su rol(empleado y cliente).
- Acceso a los usuarios registrados desde el perfil de trabajador.
- Registro, actualización y eliminación de productos.
- Visualización del stock y precio de los productos.
- Búsqueda de productos registrados.
- Facturación de productos vendidos.
- Reporte de ventas (díaria, semanal o mensual).

De cuerdo con las solicitudes realizadas por el cliente, se deja como compromiso para el 30 de agosto la entrega del módulo de login de admisión de trabajadores y CRUD de productos y usuarios.

En la Figura 22 se relaciona el acta de reunión número 1 .

Figura 22

Acta de reunión número 1



FORMATO ACTA DE REUNION

DESARROLLO DE UN SOFWARE DE GESTION DE INVENTARIOS PARA LA DROGUERIA ONASIS

ACTA Nº 01

LUGAR: DROGUERIA ONASIS FECHA: 02 - AGOSTO - 2021

NOMBRE ASISTENTES ALBERTO RODRIGUEZ CARGOS ADMINISTRADOR
FELIPE RODRIGUEZ DESARROLLADOR

TEMA A TRATAR: REUNION DE INICIO DE PROYECTO DESARROLLO DE UN SOFWARE DE GESTION DE INVENTARIOS PARA LA DROGUERIA ONASIS.

DESARROLLO: SE ESTIPULA LA ENTREGA DE LOS PRIMEROS MODULOS Y FUNCIONALIDADES PARA EL 23 AGOSTO. LOS MODULOS SON LOGIN, CRUD DE PRODUCTOS Y USUARIOS.

COMPROMISOS: PARA EL 23 DE AGOSTO SE ENTREGARAN DICHOS MODULOS Y FUNCIONALIDADES

CONVOCATORIA PRÓXIMA REUNIÓN: 23 AGOSTO

HORA INICIO 10:00 AM HORA FIN: 12:00 PM

FIRMA DE LOS PARTICIPANTES: Felipe.R

droxi DROGUERÍA ONASSIS NIT. 128004
Calle 45 Sur No. 78g - 34
Barrio Onassis - Tel: 454 38 88

Fuente: elaboración propia.

4.3.2. Creación sprint backlog

Se realizó la creación del sprint backlog con el fin de monitorear la planificación táctica del trabajo a realizar en la iteracción actual, permitiendo ver las tareas y el desarrollo de estas mismás; en caso de presentar dificultades se permite tomar decisiones al respecto. Este documento es un anexo complementario a la monografía

En la Figura 23 se relaciona el sprint backlog en el ciclo 1.

Figura 23

Spint Backlog Ciclo 1

ID TAREAS	HISTORIA DE USUARIO		ACTIVIDAD	DIAS	RESPONSABLES	DEPENDENCIA	PRIORIDAD	ESTADO(%)	CONDICIÓN DE APROBACIÓN	HECHO
	CODIGO	NOMBRE								
Ciclo de desarrollo Sprint 1 (4 semanas)										
1	HU01	LOGIN	LOGIN	28				0%		ok
1,1	HU01	LOGIN	Definir características del Login	7	Felipe Rodriguez		Alta	0%	Se definen los valores, los cuales se ingresaran para acceder a la aplicacion	ok
1,2	HU01	LOGIN	Desarrollo	7	Felipe Rodriguez	1 - 1.1	Alta	0%	validar valores de login y redireccionar al aplicativo principal	ok
1,3	HU01	LOGIN	Pruebas	7	Felipe Rodriguez	1,2	Media	0%	Validar que el programa no tenga fallos al ingresar el usuario con rol empleado y los demas roles no tegan acceso	ok
1,4	HU01	LOGIN	Correcciones	7	Felipe Rodriguez	1,3	Alta	0%	Corregir los errores presentados en las pruebas implementadas	ok
2	HU02	REGISTRO DE USUARIOS	REGISTRO DE USUARIOS	28				0%		ok
2,1	HU02	REGISTRO DE USUARIOS	Definir características del registro de usuarios	7	Felipe Rodriguez		Alta	0%	Se definen los valores, los cuales se ingresaran para registrar un usuario ya sea con el rol de empleado o cliente	ok
2,2	HU02	REGISTRO DE USUARIOS	Desarrollo	7	Felipe Rodriguez	1 - 1.1	Alta	0%	validar valores de regsito de usuarios y crear y guardar usuarios en la base de datos	ok
2,3	HU02	REGISTRO DE USUARIOS	Pruebas	7	Felipe Rodriguez	2,2	Media	0%	Validar que el programa no tenga fallos al registrar un usuario en el aplicativo y guardarlo en la base de datos	ok
2,4	HU02	REGISTRO DE USUARIOS	Correcciones	7	Felipe Rodriguez	2,3	Alta	0%	Corregir los errores presentados en las pruebas implementadas	ok
3	HU03	ACTUALIZACION DE USUARIOS	ACTUALIZACION DE USUARIOS	28				0%		ok
3,1	HU03	ACTUALIZACION DE USUARIOS	Definir características que se actualizaran en un usuario ya	7	Felipe Rodriguez		Alta	100%	Se definen los valores, los cuales se ingresaran para la actualizacion de datos del usuario ya sea con el rol de	ok

Fuente: elaboración propia

4.3.3. Iteración de desarrollo

Se realizó la primera iteración de desarrollo donde se construyeron los módulos y funcionalidades de registro, actualización y eliminación de usuarios y productos. Esto se efectuó mediante las siguientes herramientas:

- motor de base de datos realizado en MySQL mediante la herramienta phpMyAdmin.
- XAMPP sistema de gestión de base de datos MySQL e interprete de lenguajes PHP, desde allí gestionando phpMyAdmin.
- IDE NetBeans el cual permite la codificación en lenguaje Java en el que se basa el desarrollo del proyecto.
- GlassFish el cual es un servidor de código abierto para Java y funciona como un web server.

- Desarrollo de login

Se realizó la creación de base de datos con las tablas correspondientes para el desarrollo de este proyecto en phpMyAdmin; para lograr la validación de login se crea un usuario con rol de empleado y otro con rol de cliente.

En la Figura 24 se relaciona la base de datos creada con sus dos primeros usuarios .

Figura 24

Usuarios de la Base de Datos

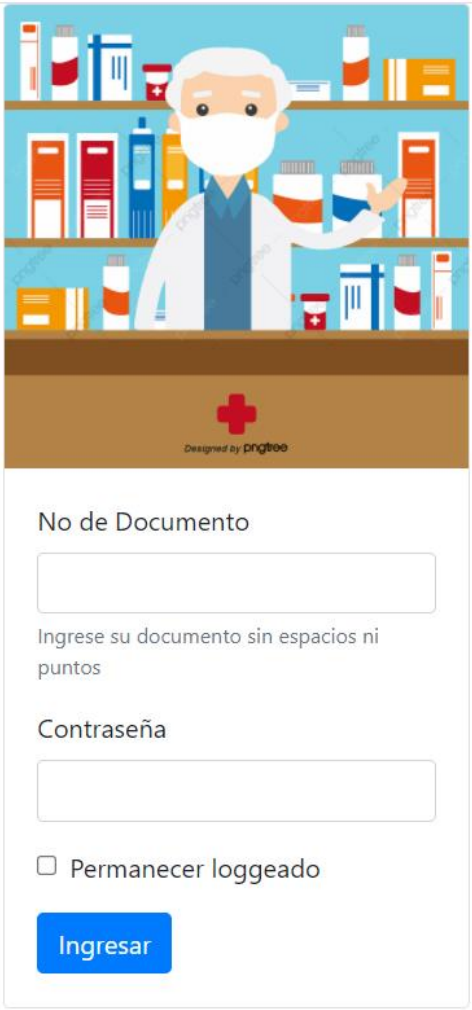
Fuente: elaboración propia

El paso siguiente es la creación de la vista de inicio de sesión, en donde los parámetros a validar para el ingreso a esta son : número de documento y password. Esta vista se realizó mediante jsp y diseños que permitiron la creación de la interfaz de login .

En la Figura 25 se relaciona la vista de login del usuario .

Figura 25

Login de Usuarios



Fuente: elaboración propia

Para continuar con la validación de login, el paso siguiente es la codificación de la parte lógica, encargada de solicitar la validación del usuario a la base de datos; esto se realiza mediante una consulta, en donde se envían los parámetros de documento y password que el usuario ingresó, se capturan estos datos mediante un request.getParameter guardándolos en variables, para luego envíarlos a la clase UsuarioDao mediante la función validar, la cual envía la siguiente consulta a la base de datos:

- select * from usuarios where documento=? and password=?.

Estos datos tienen que ser iguales a los diligenciado por el usuario. Esto se realiza mediante un executeQuery(); esta función retoma un objeto con todas las características del usuario como se puede observar en la figura 26, en donde enseguida se valida que la respuesta no sea null, su rol sea empleado y su estado sea activo, así podrá ingresar a la página principal de la aplicación.

En la Figura 26 se relaciona la función validar de la clase UsuarioDao.

Figura 26

Función Validar

```
public Usuario Validar(int documento, String password) throws SQLException {
    Usuario usuario = new Usuario();
    String consulta = "SELECT * FROM usuarios WHERE documento = ? AND password = ?";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(consulta);
        ps.setInt(1, documento);
        ps.setString(2, password);
        rs = ps.executeQuery();
        rs.next();
        do {
            usuario.setId(rs.getInt("idusuario"));
            usuario.setDocumento(rs.getInt("documento"));
            usuario.setNombre(rs.getString("nombre"));
            usuario.setPassword(rs.getString("password"));
            usuario.setCorreo(rs.getString("correo"));
            usuario.setEstado(rs.getString("ESTADO"));
            usuario.setRol(rs.getString("rol"));
            usuario.setVisibilidad(rs.getInt("visibilidad"));

        } while (rs.next());
    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
```

Fuente: elaboración propia

4.3.4. Desarrollo de pruebas login

Para esta funcionalidad se realizaron las siguientes pruebas, con el fin de identificar fallos y corregirlos y así obtener una aplicación más funcional y óptima; se evaluaron fallos que el usuario final puede cometer en el momento del login .

En la Figura 27 se relaciona la documentación de la prueba de login.

Figura 27

Prueba de Login

Nombre del proyecto: GESTIÓN DE INVENTARIOS DROGUERÍA ONASSIS		**Caso No:** 1	
Nombre de caso de la prueba: Login			
Estado de la prueba: Finalizada		**Módulo de Usuarios**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: Se valida el login de usuarios en la aplicación			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login y haber registrado un usuario en la base de datos para la validación.			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Ingresar los datos solicitados por la aplicación	La aplicación deja ingresar los datos correspondientes al usuario en este caso en documento solo ingresarán números	Exitoso
2	Presionar ingresar	Al presionar el botón ingresar la aplicación validará los datos ingresados del usuario y si son correctos cargara la página principal	Exitoso
Excepciones			
1	Presionar ingresar sin datos diligenciados	la aplicación avisará de los campos no diligenciados	Exitoso
2	Ingreso de usuario empleado inactivo o no registrado	la aplicación mostrará un mensaje indicando que el usuario no está registrado o esta inactivo	Exitoso
3	Ingrese de usuario con rol cliente	la aplicación volverá a cargar la página de login	Exitoso

Fuente : elaboración propia

- Desarrollo CRUD de usuarios

Para este módulo se creó su tabla corresponidente en la base de datos con las características solicitadas por la Droguería Onassis, la creación de esta tabla es con el fin de realizar el módulo de usuarios con las funcionalidades de: crear, actualizar y eliminar usuarios.

En la Figura 28 se relaciona la tabla corrrespondiente a usuarios.

Figura 28

Tabla de Usuarios

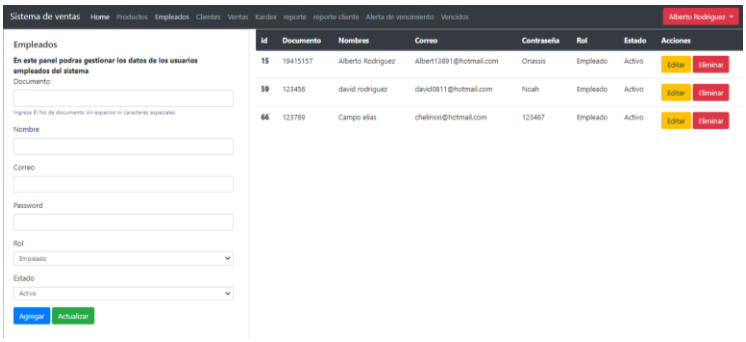
Fuente: elaboración propia.

El paso siguiente fue creación de la vista de cliente y empleado, en donde se gestiona: la creación de nuevos usuarios, su actualización y eliminación de los mismos. Lo anterior se realizó mediante jsp y diseño de bootstrap.

En la Figura 29 se relaciona la vista del crud de empleado.

Figura 29

Vista de Crud de Empleados

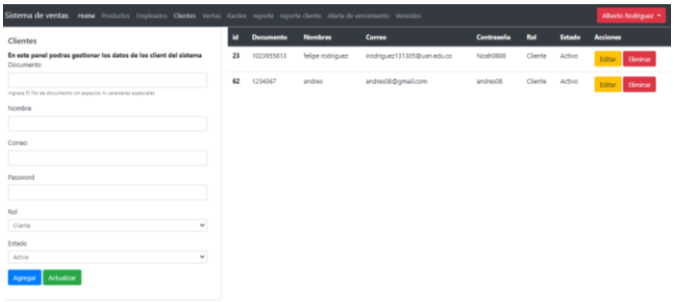


Fuente:elaboración propia

En la Figura 30 se relaciona la vista del crud de cliente.

Figura 30

Vista de Crud de Clientes



Fuente: elaboración propia

Para continuar con el crud de empleados y clientes, el paso siguiente fue la codificación de la parte lógica, encargada de recibir la información diligenciada por el usuario y direccionarla a la base de datos para su actualización. Para la creación o registro de un usuario se capturó la información diligenciada por el usuario mediante un request.getParameter() que recibe como parámetro el nombre del input de la vista de cliente o empleado. Una vez capturada la información es enviada a la clase UsuarioDAO mediante la función de esta misma clase Agregar(), la cual tiene como trabajo envíar la información

a la tabla usuarios de la base de datos para la creación del nuevo usuario; esto lo realiza mediante la siguiente sentencia:

- insert into usuarios (documento,nombre,correo,password,rol,estado,visibilidad) values (?,?,?,?,?,?,?).

Cuyos valores sean los diligenciados por el usuario, esto es realizado mediante un executeUpdate, como se puede observar en la figura 31 la estructura de la función Agregar().

En la Figura 31 se relaciona la funcionalidad de Agregar.

Figura 31

Función Agregar

```
public void Agregar(Usuario usuario) {

    String sentencia = "INSERT INTO usuarios (documento,nombre,correo,password,rol,ESTADO,visibilidad) VALUES (?,?,?,?,?,?,?)";
    try {
        con = cn.Conexion();
        ps = con.prepareStatement(sentencia);
        ps.setInt(1, usuario.getDocumento());
        ps.setString(2, usuario.getNombre());
        ps.setString(3, usuario.getCorreo());
        ps.setString(4, usuario.getPassword());
        ps.setString(5, usuario.getRol());
        ps.setString(6, usuario.getEstado());
        ps.setInt(7, usuario.getVisibilidad());
        ps.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Fuente: elaboración propia

Al tener el registro de cliente y empleados se realizó la funcionalidad de actualización. Esta es la encargada de la correción de datos erroneos o que ya no son válidos. Esto se realizó mediante el botón de editar, el cual esta al lado de cada registro del cliente o empleado. Al presionar el botón editar se captura el ID de registro del usuario en donde se envía

como parámetro a la función ListarPorId() de la clase UsuarioDAO. Esta función realiza la siguiente consulta en la base de datos:

- select * from usuarios where idusuario= ID.

En donde carga el usuario correspondiente al ID y se retorna todos sus atributos mediante un objeto de la clase Usuario. Enseguida, esto es envíado a los input de la vista ya sea de empleado o cliente, mediante un request.setAttribute(). Se cargan los datos de los usuarios correspondiente en la vista, ya sea de empleado o cliente. Se realizan las modificaciones o correcciones correspondientes y enseguida se presiona el botón de actualizar, en donde el controlador captura los datos de los input de la vista mediante un request.getParameter(), cuyo parámetro es el nombre de los input. Posteriormente estos datos son guardados en un objeto de la clase Usuario, en donde se envía como parámetro este objeto a la función Actualizar() de la clase UsuarioDAO, la cual es la encargada de envíar la siguiente sentencia a la base de datos para la actualización del usuario:

- (update usuarios set documento=?,nombre=?,correo=?,password=?,rol=?,estado=? where idusuario=?).

Esto se realiza mediante la función executeUpdate(), recibiendo esta sentencia la base de datos y actualizando el usuario correspondiente. La vista de los datos del usuario son actualizados y listados nuevamente.

En la Figura 32 se relaciona la funcionalidad Actualizar de la clase UsuarioDAO.

Figura 32

Función Actualizar

```
public void Actualizar(Usuario usuario) {
    String sentencia = "UPDATE usuarios set documento=?,nombre=?,correo=?,password=?,Rol=?,ESTADO=? WHERE idusuario=?";
    try {
        con = cn.Conexion();
        ps = con.prepareStatement(sentencia);
        ps.setInt(1, usuario.getDocumento());
        ps.setString(2, usuario.getNombre());
        ps.setString(3, usuario.getCorreo());
        ps.setString(4, usuario.getPassword());
        ps.setString(5, usuario.getRol());
        ps.setString(6, usuario.getEstado());
        ps.setInt(7, usuario.getId());
        ps.executeUpdate();

    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Fuente: elaboración propia

Ya teniendo la funcionalidad de actualización de clientes y empleado se desarrollo el método de Eliminar(). Con este proceso se puede eliminar el cliente o empleado de la vista del usuario más no de la base de datos por la integridad de la misma. Esta funcionalidad es manejada de la siguiente manera: se debe oprimir el botón de eliminar que se encuentra en la derecha del usuario que se desea elimiar. Al presionar este botón se captura el ID del usuario, el controlador captura el dato con un request.getParameter() en donde se envía el ID capturado como parámetro a la función Eliminar() de la clase UsuarioDAO. Esta envía la siguiente sentencia a la base de datos:

- update usuarios set visibilidad=1 where idusuario=" + ID.

En donde se actualiza la visibilidad de cero a uno permitiendo que la aplicación no muestre este usuario en la interfaz del programa ya que para ser visible este atributo debe estar en cero.

En la Figura 33 se relaciona la funcionalidad Eliminar() de la clase UsuarioDAO.

Figura 33

Función Eliminar

```
public void Eliminar(int id) {

    String sql = "UPDATE usuarios set visibilidad=1 WHERE idusuario=" + id;
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(sql);
        ps.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Fuente: elaboración propia

4.3.5. Desarrollo de pruebas crud usuarios

Para esta funcionalidad se realizaron las siguientes pruebas con el fin de identificar fallos y corregirlos para así obtener una aplicación más funcional y óptima. Se evaluaron fallos que el usuario final puede cometer en el momento de realizar el proceso de registro, actualización y eliminación de empleados o clientes.

En la Figura 34 se relaciona la documentación de la prueba de crud de usuarios.

Figura 34

Pruebas de Crud de Uusarios

Nombre del proyecto: GESTIÓN DE INVENTARIOS DROGUERÍA ONASSIS		**Caso No: 2**	
Nombre de caso de la prueba: Crud de usuarios			
Estado de la prueba: Finalizada		**Módulo de Usuarios**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: Se valida el correcto funcionamiento de las funcionalidades de crear, actualizar y eliminar usuarios			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login ingresar y dirigirse a la opción de empelado o cliente			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Registro de usuarios	La aplicación deja ingresar los datos correspondientes al registro de usuario y al presionar el botón de guardar se listarán los usuarios creados	Exitoso
2	Actualización usuarios	Al presionar el botón editar se cargarán los datos del usuario correspondiente permitiendo realizar las modificaciones pertinentes y al presionar actualizar se cargará el usuario en la lista ya actualizado	Exitoso
3	Presionar eliminar	Al presionar el botón eliminar el usuario ya no será visualizado en la lista de usuarios de la aplicación	Exitoso
Excepciones			
1	Presionar guardar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso
2	Presionar actualizar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso

Fuente: elaboración propia

- Desarrollo CRUD de productos

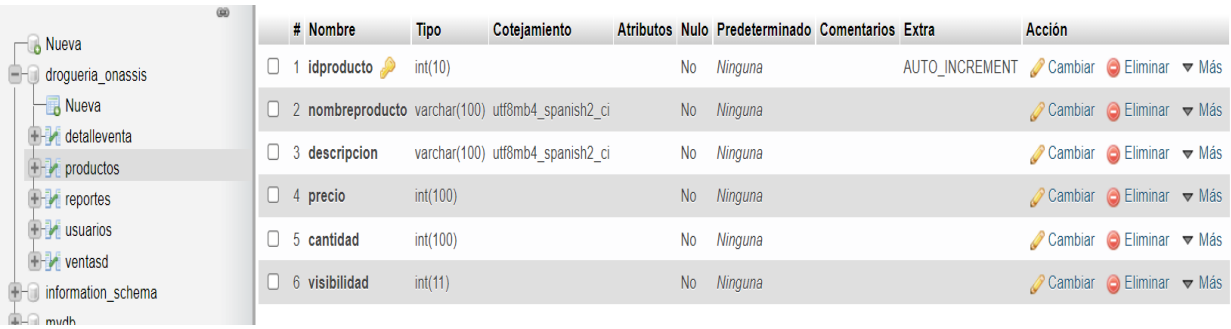
Para este módulo se creó su tabla correspondiente en la base de datos con las características solicitadas por la Droguería Onassis.

La creación de esta tabla es con el fin de realizar el módulo de productos con las funcionalidades de crear, actualizar y eliminar productos.

En la Figura 35 se relaciona la tabla corrrespondiente a productos con sus respectivos atributos y características.

Figura 35

Tabla de Productos



Fuente: elaboración propia.

El paso siguiente fue la creación de la vista de productos, en donde se gestiona la creación de nuevos productos, su actualización y eliminación de los mismos.

Esto se realizó mediante jsp y diseño de bootstrap como se puede observar en la figura 36 el diseño de la vista del crud de productos.

En la Figura 36 se relaciona la vista del crud de producto, muestra la vista en donde se gestiona la creación, actualización y eliminación de productos.

Figura 36

Vista de Crud de Productos

Fuente: elaboración propia.

Para continuar con el crud de productos el paso siguiente fue la codificación de la parte lógica, la cual es la encargada de recibir la información diligenciada por el usuario y direccionarla a la base de datos para su actualización. Para la creación o registro de un nuevo producto se capturó la información registrada por el usuario mediante un request.getParameter() que recibe como parámetro el nombre del input de la vista de producto.

Ya habiendo capturado esta información es envíada a la calse ProductoDAO mediante la función de esta misma clase AgregarProducto(), la cual tiene como trabajo envíar la información a la tabla productos de la base de datos para la creación del nuevo. Esto lo realiza mediante la sentencia:

- (insert into productos (nombreproducto, descripcion, precio, cantidad, visibilidad) values (?,?,?,?,?)).

Que son los diligenciados por el usuario. Esto es realizado mediante un executeUpdate, como se puede observar en la figura 37 la estructura de la función AgregarProducto().

En la Figura 37 se relaciona la funcionalidad de AgregarProducto de la clase ProductoDAO.

Figura 37

Función Agregar Producto

```
public void AgregarProducto(Producto producto) {
int r = 0;
con = cn.Conexion();
String sentencia = "INSERT INTO productos (nombreproducto, descripcion, precio, cantidad, visibilidad) VALUES (?,?,?,?,?)";
try {
    ps = con.prepareStatement(sentencia);
    ps.setString(1, producto.getNombreProducto());
    ps.setString(2, producto.getDescripcion());
    ps.setInt(3, producto.getPrecioProducto());
    ps.setInt(4, producto.getCantidadProducto());
    ps.setInt(5, producto.getVisivilidad());
    ps.executeUpdate();

} catch (Exception e) {

}
```

Fuente: elaboración propia

Al tener el registro de productos se prosiguió a realizar la funcionalidad de actualización, esta es la encargada de la corrección de datos erróneos o que ya no son válidos esto se realizó mediante el botón de cargar, el cual esta al lado de cada registro del producto.

Al presionar el botón cargar capturamos el ID del registro del producto en donde lo enviamos como parámetro a la función ConsultaPorCódigo() de la clase UsuarioDAO. Esta función realiza la siguiente consulta en la base de datos:

- (select * from productos where visibilidad = 0 and idproducto = códigoproducto).

En donde carga el producto correspondiente al ID del producto y se retorna todos sus atributos mediante un objeto de la clase Producto. En seguida esto es enviado a los input de la vista ya de productos mediante un request.setAttribute().

Se cargan los datos del producto correspondiente en la vista de productos. Se realizan las modificaciones o correcciónes correspondiente y en seguida se presiona el botón de actualizar en donde el controlador captura los datos de los input de la vista mediante un request.getParameter() cuyo parámetro es el nombre de los input.

A continuacion estos datos son guardados en un objeto de la clase Producto en donde se envía como parámetro este objeto a la función ActualizarProducto() de la calse ProductoDAO la cual es la encargada de envíar la siguiente sentencia a la base de datos para la actualización del producto:

- (update productos set nombreproducto=?,descripcion=?,precio=?,cantidad=? where idproducto=?).

Esto se realiza mediante la función executeUpdate() recibiendo esta sentencia la base de datos y actualizado el producto correspondiente, a continuacion en la vista los datos del producto son actualizados y listados nuevamente.

En la Figura 38 se relaciona la funcionalidad ActualizarProducto de la clase ProductoDAO.

Figura 38

Función Actualizar Producto

```
public void ActualizarProducto(Producto producto) {
String sentencia = "UPDATE productos set nombreproducto=?,descripcion=?,precio=?,cantidad=? WHERE idproducto=?";
try {
    con = cn.Conexion();
    ps = con.prepareStatement(sentencia);

    ps.setString(1,producto.getNombreProducto());
    ps.setString(2,producto.getDescripcion());
    ps.setInt(3, producto.getPrecioProducto());
    ps.setInt(4, producto.getCantidadProducto());
    ps.setInt(5, producto.getIdProducto());



    ps.executeUpdate();

} catch (SQLException ex) {
    Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
}

}
```

Fuente: elaboración propia

Ya desarrollada la funcionalidad de actualización de productos se desarrollo la funcionalidad de eliminar, para esta funcionalidad se elimina el producto de la vista del usuario más no de la base de datos por la integridad de la misma. Esta funcionalidad es manejada de la siguiente manera: el usuario oprimirá el botón de eliminar que se encuentra en la derecha del producto que desea eliminar.

Al presionar este botón se captura el ID del producto, el controlador captura el dato con un request.getParameter() en donde se envía el ID capturado como parámetro a la función EliminarProducto de la clase ProductoDAO. Esta envía la siguiente sentencia a la base de datos:

- (update productos set visibilidad=1 where idproducto= id),en donde se actualiza la visibilidad de cero a uno permitiendo que la aplicación no muestre este producto en la interfaz de producto ya que para ser visible este atributo debe estar en cero.

En la Figura 39 se relaciona la funcionalidad EliminarProducto de la clase ProductoDAO.

Figura 39

Función Eliminar Producto

```
    public void EliminarProducto(int id) {

    String sql = "UPDATE productos set visibilidad=1 WHERE idproducto=" + id;
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(sql);
        ps.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(ProductoDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Fuente: elaboración propia

4.3.6. Desarrollo de pruebas crud productos

Para esta funcionalidad se realizaron las siguientes pruebas con el fin de identificar fallos y corregirlos para así obtener una aplicación más funcional y óptima, se evaluaron fallos que el usuario final puede cometer en el momento de realizar las funcionalidades de registro, actualización y eliminación de productos.

En la Figura 40 se relaciona la documentación de la prueba de crud de productos.

Figura 40

Prueba de Crud de Productos

Nombre del proyecto: GESTIÓN DE INVENTARIOS DROGUERÍA ONASSIS		**Caso No: 3**	
Nombre de caso de la prueba: Crud de productos			
Estado de la prueba: Finalizada		**Módulo de Productos**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: Se valida el correcto funcionamiento de las funcionalidades de crear, actualizar y eliminar usuarios			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login ingresar a la opción de productos			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Registro de productos	La aplicación deja ingresar los datos correspondientes al registro de un producto y al presionar el botón de guardar se listarán los productos creados	Exitoso
2	Actualización usuarios	Al presionar el botón cargar se cargarán los datos del producto correspondiente permitiendo realizar las modificaciones pertinentes y al presionar actualizar se cargará el producto en la lista ya actualizado	Exitoso
3	Presionar eliminar	Al presionar el botón eliminar el producto ya no será visualizado en la lista de productos de la aplicación	Exitoso
Excepciones			
1	Presionar guardar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso
2	Presionar actualizar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso

Fuente: elaboración propia

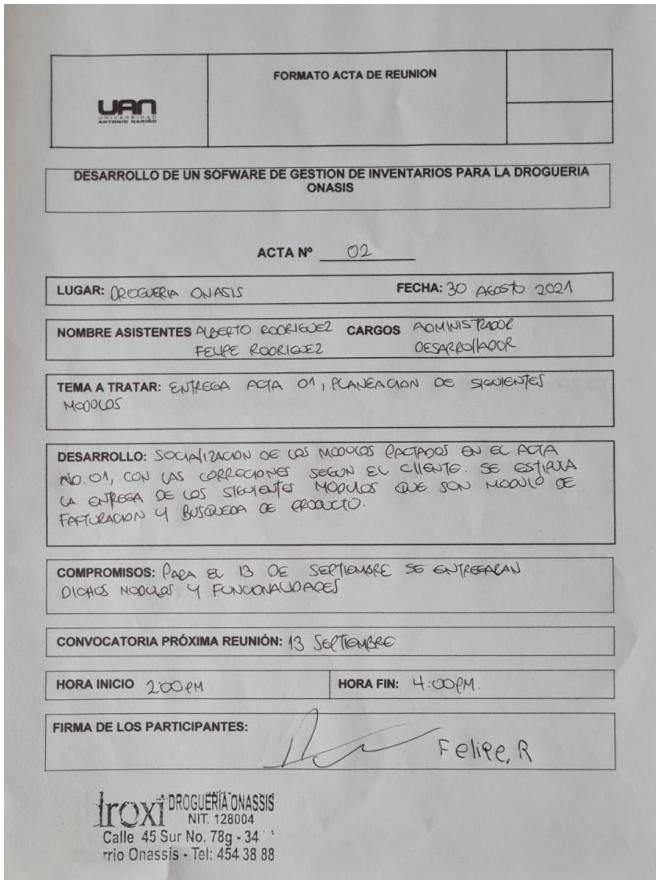
4.3.7. Segunda reunión

Se socializó con el cliente el proceso de ingreso al aplicativo con número de documeto y password establecido, se dió a conocer el paso a paso para el registro, actualización y eliminación de usuarios y productos.Para la reunión del 13 de septiembre se programó entregar el módulo de facturación y búsqueda de producto.

En la Figura 41 se relaciona el acta de reunión No. 2, en donde se puede observar los compromisos de entrega para la próxima reunión y la firma de los participantes.

Figura 41

Acta de Reunión número 2



FORMATO ACTA DE REUNION

DESARROLLO DE UN SOFWARE DE GESTION DE INVENTARIOS PARA LA DROGUERIA ONASIS

ACTA Nº 02

LUGAR: DROGUERIA ONASIS FECHA: 30 AGOSTO 2021

NOMBRE ASISTENTES ALBERTO RODRIGUEZ CARGOS ADMINISTRADOR
FELIPE RODRIGUEZ DESARROLLADOR

TEMA A TRATAR: ENTREGA ACTA 01, PLANEACION DE SIGUIENTES MODULOS

DESARROLLO: SOCIALIZACION DE LOS MODULOS PACTADOS EN EL ACTA NO.01, CON LAS CORRECCIONES SEGUN EL CLIENTE. SE ESTIPULA LA ENTREGA DE LOS SIGUIENTES MODULOS QUE SON MODULO DE FACTURACION Y BUSQUEDA DE PRODUCTO.

COMPROMISOS: PARA EL 13 DE SEPTIEMBRE SE ENTREGARAN DICHOS MODULOS Y FUNCIONALIDADES

CONVOCATORIA PRÓXIMA REUNIÓN: 13 SEPTIEMBRE

HORA INICIO 2:00 PM HORA FIN: 4:00PM

FIRMA DE LOS PARTICIPANTES: Felipe, R

DROGUERÍA ONASSIS
NIT. 128004
Calle 45 Sur No. 78g - 34
rrio Onassis - Tel: 454 38 88

Fuente: elaboración propia

4.4.Desarrollo Sprint Dos

Para el correcto desarrollo de este proyecto se realizaron 3 sprint, cada uno de 4 semanas, dividiendo en cada uno de los sprints entregas puntuales o avances del software que el cliente, en este caso la Droguería Onassis representada por su administrador Ángel Alberto Rodríguez revisó y aprobó el resultado de los entregables del proyecto el siguiente es el desarrollo del sprint número dos.

4.4.1. Actualización sprint backlog

De acuerdo al proceso en el sprint No.1 se realiza la actualización del sprint backlog culminando cada una de las tareas plasmadas. Adicionalmente, se plantean las nuevas tareas las cuales ayudará a la culminacion del sprint No.2.

En la Figura 42 se relaciona la actualización y nuevas tareas del sprint backlog.

Figura 42

Sprint Backlog Cliclo Número 2

6,3	HU06	ACTUALIZACION DE PRODUCTOS	Pruebas	7	Felipe Rodriguez	6,2	Media	100%	Validar que el programa no tenga fallos al actualizar un producto en el aplicativo y al guardarlo en la base de datos	ok
6,4	HU06	ACTUALIZACION DE PRODUCTOS	Correcciones	7	Felipe Rodriguez	6,3	Alta	100%	Corregir los errores presentados en las pruebas implementadas	ok
7	HU07	ELIMINAR PRODUCTOS	ELIMINAR PRODUCTOS	28				100%		ok
7,1	HU07	ELIMINAR PRODUCTOS	Definir caracteristicas que se implementaran para eliminar un producto	7	Felipe Rodriguez		Alta	100%	Se definen los pasos a seguir para poder eliminar un producto del aplicativo	ok
7,2	HU07	ELIMINAR PRODUCTOS	Desarrollo	7	Felipe Rodriguez	5	Alta	100%	validar que la funcionalidad de eliminar productos se vea reflejada tanto en el aplicativo como en la base de datos	ok
7,3	HU07	ELIMINAR PRODUCTOS	Pruebas	7	Felipe Rodriguez	7,2	Media	100%	Validar que el programa no tenga fallos al eliminar un productos en el aplicativo y guardarlo en la base de datos	ok
7,4	HU07	ELIMINAR PRODUCTOS	Correcciones	7	Felipe Rodriguez	7,3	Alta	100%	Corregir los errores presentados en las pruebas implementadas	ok
Ciclo de desarrollo Sprint 2 (4 semanas)										
8	HU08	BUSQUEDA DE PRODUCTOS	BUSQUEDA DE PRODUCTOS	28				0%		ok
8,1	HU08	BUSQUEDA DE PRODUCTOS	Definir caracteristicas que se implementaran para la busqueda de productos	7	Felipe Rodriguez		Alta	0%	Se definen los pasos a seguir para poder realizar la busqueda de productos en el aplicativo	ok
8,2	HU08	BUSQUEDA DE PRODUCTOS	Desarrollo	7	Felipe Rodriguez	5	Alta	0%	validar que la funcionalidad de buscar productos se vea reflejada tanto en el aplicativo como en la base de datos	ok
8,3	HU08	BUSQUEDA DE PRODUCTOS	Pruebas	7	Felipe Rodriguez	8,2	Media	0%	Validar que el programa no tenga fallos al realizar la busqueda de un producto en el aplicativo	ok
8,4	HU08	BUSQUEDA DE PRODUCTOS	Correcciones	7	Felipe Rodriguez	8,3	Alta	0%	Corregir los errores presentados en las pruebas implementadas	ok
9	HU09	BUSQUEDA DE PRODUCTOS VENTA	BUSQUEDA DE PRODUCTOS VENTA	28				0%		ok
9,1	HU09	BUSQUEDA DE PRODUCTOS VENTA	Definir caracteristicas que se implementaran para la busqueda de productos para una venta	7	Felipe Rodriguez		Alta	0%	Se definen los pasos a seguir para poder realizar la busqueda de productos en el aplicativo	ok
9,2	HU09	BUSQUEDA DE PRODUCTOS VENTA	Desarrollo	7	Felipe Rodriguez	5	Alta	0%	validar que la funcionalidad de buscar productos se vea reflejada tanto en el aplicativo como en la base de datos	ok
9,3	HU09	BUSQUEDA DE PRODUCTOS VENTA	Pruebas	7	Felipe Rodriguez	9,2	Media	0%	Validar que el programa no tenga fallos al realizar la queda de un producto en el aplicativo	ok

Fuente: elaboración propia.

4.4.2. Iteración de desarrollo

Se realizó la segunda iteración de desarrollo donde se construyeron los módulos y funcionalidades de facturación y búsqueda de producto. Esto se efectuó mediante las siguientes herramientas:

- motor de base de datos realizado en MySQL mediante la herramienta phpMyAdmin.
- XAMPP sistema de gestión de base de datos MySQL e interprete de lenguajes PHP, desde allí gestionando phpMyAdmin.
- IDE NetBeans el cual permite la codificación en lenguaje Java en el que se basa el desarrollo del proyecto.
- GlassFish es un servidor de código abierto para Java y funciona como un web server.

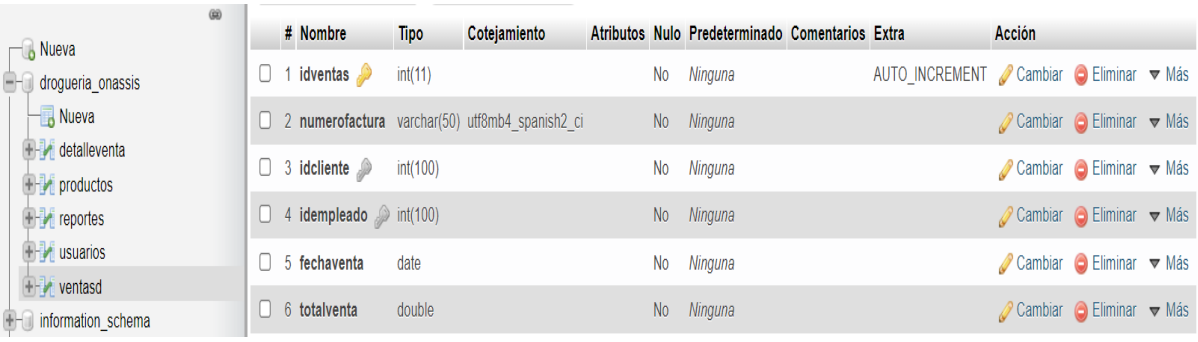
- Desarrollo de facturación

Para este módulo se crearon las tablas correspondiente en la base de datos para cumplir con las características solicitadas por la Droguería Onassis; la creación de estas tablas es con fin de realizar el módulo de facturación.

En la Figura 43 se relaciona la tabla corrrespondiente a ventas con sus respectivos atributos y características y relaciones.

Figura 43

Tabla de Ventas

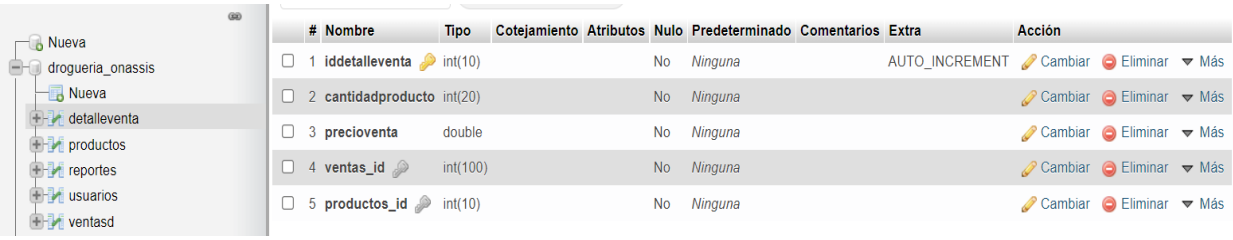


Fuente: elaboración propia.

En la Figura 44 se relaciona la tabla corrrespondiente a detalle venta con sus respectivos atributos, características y relaciones.

Figura 44

Tabla de Detalle de Venta



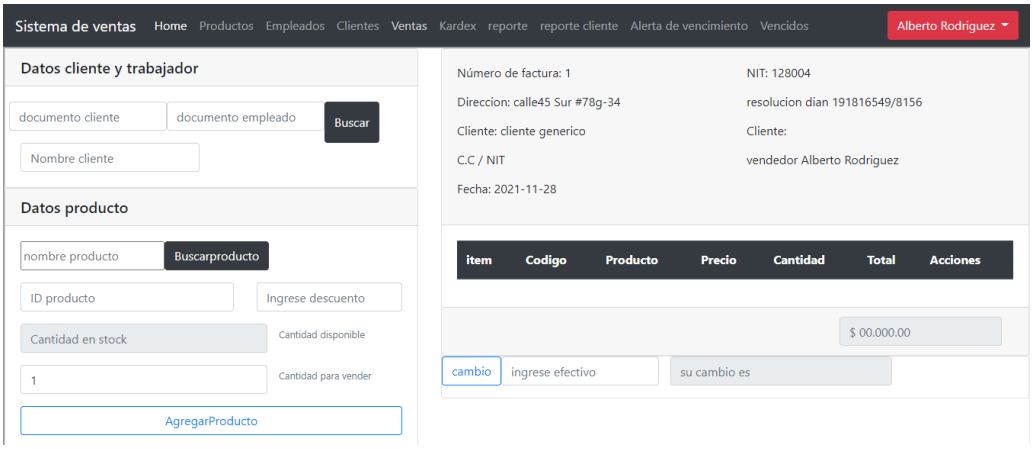
Fuente: elaboración propia.

El paso siguiente es la creación de la vista de facturación, en donde se gestiona la venta de los medicamentos. Esta vista se realizó mediante jsp y diseño de bootstrap.

En la Figura 45 se relaciona la vista de facturación.

Figura 45

Vista de venta



Fuente: elaboración propia.

Para continuar con el módulo de facturación, el paso siguiente fue la codificación de la parte lógica, la cual es la encargada de recibir la información diligenciada por el usuario y direccionarla a la base de datos para su actualización. Para la creación de una factura se captura varia información; la primera de ella son los datos del cliente y trabajador, que en este caso son los números de documento diligenciados por el usuario. Al presionar el botón de buscar el controlador captura esta información mediante un request.getParameter() que recibe como parámetro el nombre del input de la vista de ventas; ya habiendo capturado esta información es enviada a la clase UsuarioDAO mediante la funciones de esta misma clase BuscarEmpleado() y BuscarCliente(), la cual tiene como trabajo envíar la información a la base de datos para realizar una consulta de los usuarios esto se realiza con la sentencia:

- (select * from usuarios where documento = ? and visibilidad=0).

Los datos diligenciados tienen que ser iguales a los registrados por el usuario, esto se realializó mediante un executeQuery(), al obtener esta información se guardó en un objeto de tipo usuario en donde el controlador validará primero que el usuario cliente no sea null en caso de ser null lo redirecciona al módulo de registro de clientes, esto por medio de request.getRequestDispatcher(). En caso contrario, validará que el usuario empleado no sea null en caso de no serlo el controlador envíará la información de cliente y empleado a la vista mediante un request.setAttribute, en donde se volverá a cargar el documeto del cliente y además su nombre; en el caso del empleado solo cargará su documento, si en alguna circunstancia el empleado resulta ser null la aplicación le avisará que el trabajador no se encuentra registrado.

En la Figura 46 se relaciona la función de BuscarCliente.

Figura 46

Funcion Buscar Cliente

```
public Usuario BuscarCliente(int documento){
    Usuario usuario = new Usuario();
    String consulta = "SELECT * FROM usuarios WHERE documento = ? AND visibilidad =0";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(consulta);
        ps.setInt(1, documento);
        rs = ps.executeQuery();
        while(rs.next()){
            usuario.setId(rs.getInt("idusuario"));
            usuario.setDocumento(rs.getInt("documento"));
            usuario.setNombre(rs.getString("nombre"));
            usuario.setCorreo(rs.getString("correo"));
            usuario.setRol(rs.getString("rol"));
            System.err.println(""+usuario.getNombre());
          }
    } catch (Exception e) {
    }
    return usuario;
}
```

Fuente:elaboración propia.

En la Figura 47 se relaciona la función de BuscarEmpleado.

Figura 47

Función Buscar Empleado

```
public Usuario BuscarEmpleado(int documento){
    Usuario usuario = new Usuario();
    String consulta = "SELECT * FROM usuarios WHERE documento = ? AND visibilidad=0";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(consulta);
        ps.setInt(1, documento);
        rs = ps.executeQuery();
        while(rs.next()){
            usuario.setId(rs.getInt("idusuario"));
            usuario.setDocumento(rs.getInt("documento"));
            usuario.setNombre(rs.getString("nombre"));
            usuario.setCorreo(rs.getString("correo"));
            usuario.setRol(rs.getString("rol"));
            System.err.println(""+usuario.getNombre());
          }
    } catch (Exception e) {
    }
    return usuario;
}
```

Fuente: elaboración propia.

En la Figura 48 se relaciona la validación del controlador a cliente y empleado.

Figura 48

Validar Cliente y Empleado

```
int documentoCliente = Integer.parseInt(request.getParameter("documentocliente"));
int documentoEmpleado = Integer.parseInt(request.getParameter("documentoempleado"));
usuarioEmpleado = usuarioDAOE.BuscarEmpleado(documentoEmpleado);
usuario = usuarioDAO.BuscarCliente(documentoCliente);
if (usuario.getNombre() == null) {

    request.getRequestDispatcher("Controlador?menu=Clientes&accion=Listar").forward(request, response);
} else if (usuarioEmpleado.getNombre() == null) {

    request.setAttribute("Validar", validar);
} else {
    request.setAttribute("cliente", usuario);
    request.setAttribute("empleado", usuarioEmpleado);
}
break;
```

Fuente: elaboración propia.

Ya desarrollada la funcionalidad de búsqueda y validación del cliente y empleado, el paso a seguir es realizar la búsqueda del producto para su venta. Esto se realiza mediante la captura de la información que en este caso es el ID del producto que el usuario ingresó. La captura la hace el controlador en el momento en que el usuario presione el botón buscar producto y se captura por medio de un request.getParameter, que recibe como parámetro el nombre del input de la vista de venta; de ahí es envíado a la clase ProductoDAO mediante la función ConsultarPorCódigo. Ella es la encargada de envíar la siguiente sentencia a la base de datos:

- (select * from productos where visibilidad = 0 and idproducto =códigoproducto).

El resultado de esta consulta es guardado en un objeto de tipo usuario que el controlador validará que no sea null y lo envíará a la vista mediante un request.setAttribute(), en donde la vista de venta cargará la siguiente información del producto: ID, nombre, precio y cantidad disponible.

En la Figura 49 se relaciona la función de ConsultarPorCódigo.

Figura 49

Función Consultar po Código

```
public Producto ConsultaPorCodigo(int codigoProducto) {
Producto producto = new Producto();
con = cn.Conexion();
String consulta = "SELECT * FROM productos WHERE  visibilidad = 0 AND idproducto = " + codigoProducto;

try {
    ps = con.prepareStatement(consulta);
    rs = ps.executeQuery();
    while(rs.next()){
        producto.setIdProducto(rs.getInt(1));
        producto.setNombreProducto(rs.getString(2));
        producto.setDescripcion(rs.getString(3));
        producto.setPrecioProducto(rs.getInt(4));
        producto.setCantidadProducto(rs.getInt(5));


    }
} catch (SQLException ex) {
    Logger.getLogger(ProductoDAO.class.getName()).log(Level.SEVERE, null, ex);
}
return producto;
```

Fuente: elaboración propia.

Ya al tener el cliente, empleado y producto validados con su información correspondiente se continua con la funcionalidad de agregar productos, en donde el usuario ingresa la cantidad del producto para su venta, presionará el botón de agregar producto, el controlador captura toda la información del producto de los input de esta sección mediante un request.getParameter() que recibe como parámetro los nombres de los input de venta.

En el controlador esta información es guardada en un objeto de tipo venta y de ahí es guardada en un ArraList de objetos para que el usuario agregue los productos que la venta requiera.

En la Figura 50 se relaciona el procedimiento de captura de datos y agregar productos del controlador.

Figura 50

Código de Agregar Producto

```
case "AgregarProducto":
    totalapagar = 0;
    venta = new Venta();
    codProducto = Integer.parseInt(request.getParameter("codigoproducto"));
    descripcion = request.getParameter("nombreproducto");
    precio = Integer.parseInt(request.getParameter("precioproducto"));
    cantidad = Integer.parseInt(request.getParameter("cantidadproducto"));
    subtotal = precio * cantidad;
    venta.setItem(item);
    venta.setDescripcionproducto(descripcion);
    venta.setCantidad(cantidad);
    venta.setPrecio(precio);
    venta.setSubTotal(subtotal);
    venta.setIdProducto(codProducto);
    listaVentas.add(venta);
```

Fuente: elaboración propia.

Al agregar los productos pertinentes a la venta el paso a seguir es generar la venta. Esto se realiza cuando el usuario presiona el botón de generar venta en donde el controlador envíará la información a la clase VentaDAO mediante las siguientes funciones: RegistrarVenta() la cual envía la siguiente sentencia a la base de datos para su registro:

- insert into ventasd (idcliente,idempleado,númerofactura,fechaventa,totalventa) values (?,?,?,?,?).

Esto por medio de un executeUpdate() guardando la venta en la base de datos.

La siguiente función es guardardettalleventa() la cual envía la siguiente sentencia a la base de datos para registrar los detalles de la venta:

- (insert into detalleventa (ventas_id,productos_id,cantidadproducto,precioventa) values (?,?,?,?)).

Lo anterior a través de un executeUpdate(), guardando los detalles de la venta en la base de datos.

En la Figura 51 se relaciona las funciones de RegistarVenta() y GuardarDetalleVenta().

Figura 51

Funciones Registrar Venta y Guardar Detalle

```
public void RegistrarVenta(Venta venta){
    String sentencia = "INSERT INTO ventasd (idcliente,idempleado,numerofactura,fechaventa,totalventa) VALUES(?,?,?,?,?)";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(sentencia);
        ps.setInt(1,venta.getIdCliente());
        ps.setInt(2, venta.getIdEmpleado());
        ps.setString(3, venta.getComprobante());
        ps.setString(4, venta.getFecha());
        ps.setDouble(5, venta.getMonto());
        ps.executeUpdate();

    } catch (SQLException ex) {
        Logger.getLogger(VentaDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

}

public void GuardarDetalleVenta(Venta venta){
     String sentencia = "INSERT INTO detalleventa (ventas_id,productos_id,cantidadproducto,precioventa) VALUES(?,?,?,?)";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(sentencia);
        ps.setInt(1,venta.getIdventa());
        ps.setInt(2, venta.getIdProducto());
        ps.setInt(3, venta.getCantidad());
        ps.setDouble(4, venta.getPrecio());
        ps.executeUpdate();

    } catch (SQLException ex) {
        Logger.getLogger(VentaDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Fuente: elaboración propia.

4.4.3. Desarrollo de pruebas facturación

Para esta funcionalidad se realizaron las siguientes pruebas con el fin de identificar fallos y corregirlos y obtener una aplicación más funcional y óptima; se evaluaron fallos que el usuario final puede cometer en el momento de realizar una venta.

En la Figura 52 se relaciona la la documentación de la prueba de módulo de ventas

Figura 52

Pruebas de Módulo de Ventas

Nombre del proyecto: GESTIÓN DE INVENTARIOS DROGUERÍA ONASSIS		**Caso No: 4**	
Nombre de caso de la prueba: facturación			
Estado de la prueba: Finalizada		**Módulo de ventas**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: se valida el correcto funcionamiento del modulo de venta y sus funcionalidades			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login ingresar a la opción de ventas			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Buscar cliente y empelado	La aplicación deja ingresar los datos correspondientes para la búsqueda los cuales son el número del documento al presionar buscar validará la información si se encuentran registrados	Exitoso
2	Búsqueda producto	La aplicación deja ingresar los datos correspondientes para la búsqueda el cual es el id del producto al presionar buscarproducto validará la información si se encuentra registrado	Exitoso
3	Agregar producto	Al presionar el botón de agregarproducto se actualizará la lista de venta	Exitoso
4	Generar venta	Al presionar el botón de generar venta se descontarán los productos y brindara la opción de imprimir factura	Exitoso
Excepciones			
1	Presionar buscar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso
2	Presionar buscarproducto sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso
3	Presionar agregar sin datos diligenciados	La aplicación avisará de los campos no diligenciados	Exitoso

Fuente: elaboración propia.

- Desarrollo de búsqueda de producto

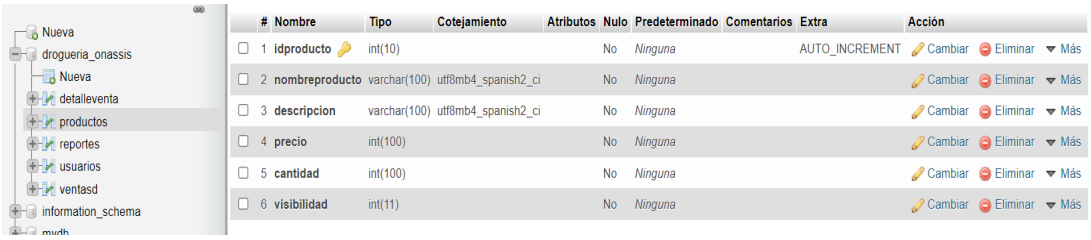
Para este módulo se utilizó la tabla correspondiente en la base de datos para cumplir con las características solicitadas por la Droguería Onassis; en este caso se empleo la tabla

de productos con el fin de realizar el módulo de kárdex con la funcionalidad de buscar producto.

En la Figura 53 se relaciona la tabla corrrespondiente a productos.

Figura 53

Tabla de Productos



Fuente: elaboración propia.

El paso siguiente fue creación de la vista de Kárdex. En donde se gestiona la búsqueda de productos y su actualización. Esto se realizó mediante jsp y diseño de bootstrap.

En la Figura 54 se relaciona la vista corrrespondiente a kárdex.

Figura 54

Vista de Kárdex

Fuente: elaboración propia.

Ya teniendo la vista creada de kárdex se prosiguió con el desarrollo de la parte lógica de la búsqueda de producto, la cual es la encargada de capturar la información diligenciada del producto que el usuario desea buscar. Esto lo realiza el controlador mediante un request.getParameter(), el cual recibe como parámetro el nombre del input de la vista del kárdex donde el usuario esta diligenciando la información del producto. Esta información es enviada a la clase ProductoDao, mediante la función ListarKárdex() de esta misma clase,la cual recibe como parámetro el nombre, ID o estado del producto a buscar. Enseguida esta función envía la siguiente sentencia a la base de datos:

- select * from productos where visibilidad =0 and (nombreproducto like '%" + texto + "%' or idproducto like '%" + texto + "%' or estado like '%" + texto + "%').

En donde la base de datos realiza la búsqueda que concuerde con los parámetros solicitados por el usuario, esto gracias a un executeQuery() la base de datos retorna la información del producto o productos que concuerden con la búsqueda. Esta información se en un objeto producto de la clase Producto, para luego ser agregado a un ArrayList de objetos. Esta lista es retornada en donde el controlador validará que la lista no este vacía, en caso de estarlo la aplicación le avisara al usuario que el producto que esta buscando no se encuentra registrado, de lo contrario se listará el producto o productos que concuerden con la búsqueda del usuario. Esto es envíado gracias a un request.setAttribute() el cual es el encargado de enviar la información a la vista para que sea visualizada por el usuario.

En la Figura 55 se relaciona el método ListarKárdex.

Figura 55

Función Listar Kárdex

```
public List Listarkardex( String texto) {

    String consulta = "SELECT * FROM productos WHERE visibilidad =0 AND (nombreproducto LIKE '%"+texto+"%' or idproducto LIKE '%"+texto+"
    List<Producto> lista = new ArrayList();

    try {
        con = cn.Conexion();
        ps = con.prepareStatement(consulta);
        rs = ps.executeQuery();
        while (rs.next()) {
            Producto producto = new Producto();
            producto.setIdProducto(rs.getInt("idproducto"));
            producto.setNombreProducto(rs.getString("nombreproducto"));
            producto.setDescripcion(rs.getString("descripcion"));
            producto.setPrecioProducto(rs.getInt("precio"));
            producto.setCantidadProducto(rs.getInt("cantidad"));
            lista.add(producto);

        }
    } catch (SQLException ex) {

    }
    return lista;

}
```

Fuente: elaboración propia.

En la Figura 56 se relaciona la validación por parte del controlador.

Figura 56

Validación de controlador

```
if (menu.equals("Kardex")) {

    switch (accion) {
        case "Listarp":
            List lista = productoDAO.Listarkardex(texto);
            request.setAttribute("Producto", lista);
            break;
        case "Buscar":
            String dato = request.getParameter("txtBuscar");
            List lista1 = productoDAO.Listarkardex(dato);
            if (lista1.isEmpty()) {
                int existencia = 1;
                request.setAttribute("Existencia", existencia);
                request.setAttribute("Producto", lista1);
            } else {
                int existencia = 0;
                request.setAttribute("Existencia", existencia);
                request.setAttribute("Producto", lista1);
            }
```

Fuente: elaboración propia

Ya desarrollada la parte lógica de búsqueda de producto se realizó la funcionalidad de actualizar la que ya ha sido utilizada anteriormente y se puede utilizar nuevamente en la vista del kárdex. Para esto, el usuario oprime el botón de editar, en donde se captura el ID del producto seleccionado, esto se realiza mediante un request.getParameter(), el cual recibe como parámetro el valor que se captura al oprimir este botón, que es el ID del producto. Este dato es envíado a la clase ProductoDao mediante la función ConsultarPorCódigo(), ella es la encargada de envíar la siguiente consulta a la base de datos:

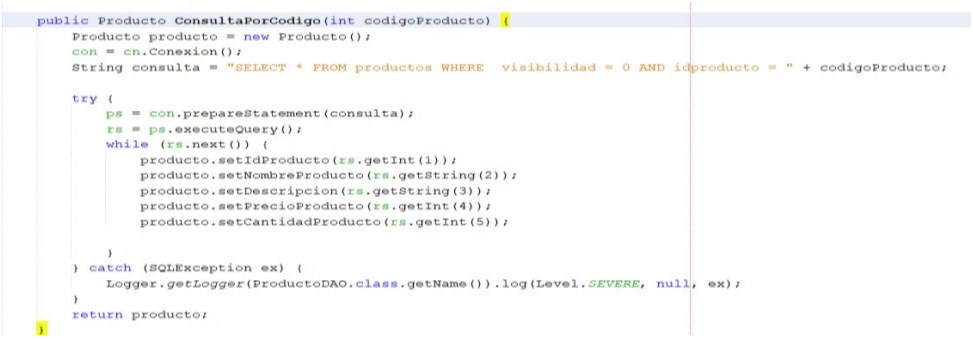
- (select * from productos where visibilidad = 0 and idproducto = " + códigoproducto).

Esta retorna los atributos del ID seleccionado, estos valores se guardan en un objeto de la clase Producto y este objeto es retornado al controlador, en donde el lo envía a la vista de kárdex por medio de request.setAttribute() y la vista los caraga en los inputs correspondientes para la visualización del usuario.

En la Figura 57 se relaciona la funcionalidad de ConsultaPorCódigo.

Figura 57

Función Consulta por Código



```
public Producto ConsultaPorCodigo(int codigoProducto) {
    Producto producto = new Producto();
    con = cn.Conexion();
    String consulta = "SELECT * FROM productos WHERE  visibilidad = 0 AND idproducto = " + codigoProducto;

    try {
        ps = con.prepareStatement(consulta);
        rs = ps.executeQuery();
        while (rs.next()) {
            producto.setIdProducto(rs.getInt(1));
            producto.setNombreProducto(rs.getString(2));
            producto.setDescripcion(rs.getString(3));
            producto.setPrecioProducto(rs.getInt(4));
            producto.setCantidadProducto(rs.getInt(5));

        }
    } catch (SQLException ex) {
        Logger.getLogger(ProductoDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return producto;
}
```

Fuente: elaboración propia

Al tener la información del producto seleccionado en la vista se procede a la modificación que el usuario crea pertinente. Al presionar el botón de actualizar se captura la información de los inputs de la vista de kárdex esto se realiza por medio de un request.getParameter() que recibe como parámetro el nombre de los input, esta información capturada es envíada a un objeto de la clase Producto, en donde este objeto es enviado a la clase ProductoDAO mediante la función ActualizarKárdex(). Ella es la encargada de enviar el siguiente comando a la base de datos.

- (update productos set nombreproducto=?,precio=?,cantidad=? where idproducto=?).

Se actualizarán los valores diligenciados por el usuario esta función no retorna ningún valor, enseguida el controlador envía la siguiente sentencia request.getRequestDispatcher() en donde se listarán todos los medicamentos del kárdex con el medicamento que el usuario realizó la actualización.

En la Figura 58 se relaciona la funcionalidad de ActualizarKárdex.

Figura 58

Función Actualizar Kárdex

```
public void ActualizarKardex(Producto producto) {
String sentencia = "UPDATE productos set nombreproducto=?,precio=?,cantidad=? WHERE idproducto=?";
try {
    con = cn.Conexion();
    ps = con.prepareStatement(sentencia);

    ps.setString(1,producto.getNombreProducto());
    ps.setInt(2, producto.getPrecioProducto());
    ps.setInt(3, producto.getCantidadProducto());
    ps.setInt(4, producto.getIdProducto());



    ps.executeUpdate();

} catch (SQLException ex) {
    Logger.getLogger(ProductoDAO.class.getName()).log(Level.SEVERE, null, ex);
}

}
```

Fuente de elaboración propia

En la Figura 59 se relaciona la documentación de la prueba de búsqueda y actualización de prodcuto en el kárdex.

Figura 59

Pruebas del Modulo de Kárdex

Nombre del proyecto: GESTION DE INVENTARIOS DROGUERIA ONASSIS		**Caso No: 5**	
Nombre de caso de la prueba: funcionalidades Kárdex			
Estado de la prueba: Finalizada		**módulo de Kárdex**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: Se valida el correcto funcionamiento de las funcionalidades de buscar y actualizar producto			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login ingresar a la opción de Kárdex			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Buscar producto	La aplicación deja ingresar los datos correspondientes a la búsqueda de un producto y al presionar el botón de buscar se listará el producto buscado	Exitoso
2	Actualizar producto	al presionar el botón editar se cargarán los datos del producto correspondiente permitiendo realizar las modificaciones pertinentes y al presionar actualizar se cargará el producto en la lista ya actualizado	Exitoso
Excepciones			
1	presionar buscar sin datos diligenciados	la aplicación avisara de los campos no diligenciados	Exitoso
2	presionar actualizar sin datos diligenciados	la aplicación avisara de los campos no diligenciados	Exitoso

Fuente: elaboración propia

4.4.4. Tercera reunión

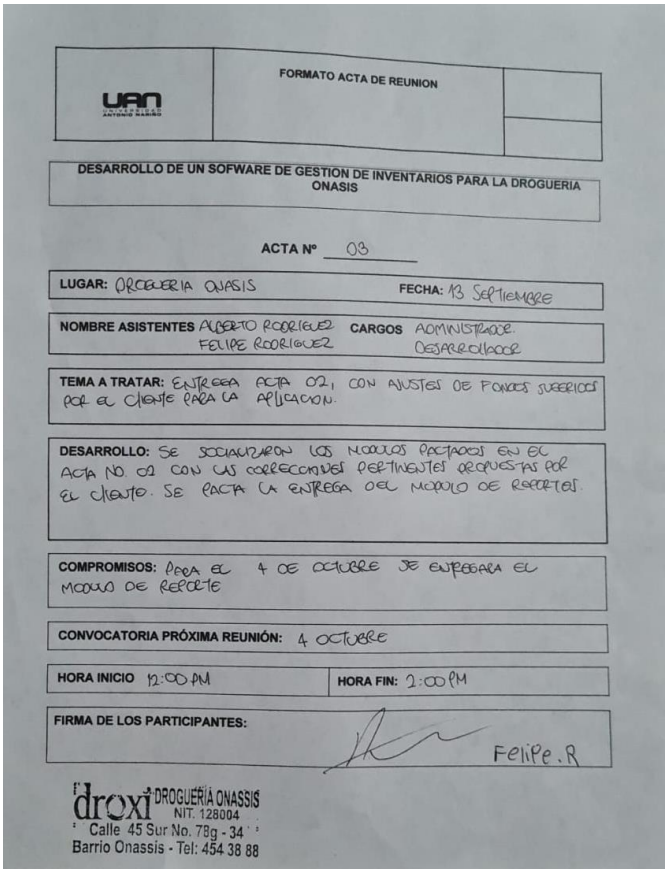
Se socializó con el cliente el proceso de venta de medicamentos y su facturación, se dio a conocer el paso a paso para la búsqueda de productos y actualización en el módulo de kárdex.

Para la reunión del 04 de octubre se programó entregar el de módulo de reporte.

En la Figura 60 se relaciona el acta de reunión número 3.

Figura 60

Acta de Reunión número 3



FORMATO ACTA DE REUNION

DESARROLLO DE UN SOFWARE DE GESTION DE INVENTARIOS PARA LA DROGUERIA ONASIS

ACTA Nº 03

LUGAR: DROGUERIA ONASIS FECHA: 13 SEPTIEMBRE

NOMBRE ASISTENTES ALBERTO RODRIGUEZ FELIPE RODRIGUEZ CARGOS ADMINISTRADOR DESARROLLADOR

TEMA A TRATAR: ENTREGA ACTA 02, CON AJUSTES DE FORMAS SUGERIDOS POR EL CLIENTE PARA LA APLICACION.

DESARROLLO: SE SOCIALIZARON LOS MODULOS PACTADOS EN EL ACTA NO. 02 CON LAS CORRECCIONES PERTINENTES PROPUESTAS POR EL CLIENTE. SE PACTA LA ENTREGA DEL MODULO DE REPORTES.

COMPROMISOS: PARA EL 4 DE OCTUBRE SE ENTREGARA EL MODULO DE REPORTE

CONVOCATORIA PRÓXIMA REUNIÓN: 4 OCTUBRE

HORA INICIO 12:00 PM HORA FIN: 2:00 PM

FIRMA DE LOS PARTICIPANTES: Felipe.R

droxi DROGUERÍA ONASSIS NIT. 128004
Calle 45 Sur No. 78g - 34
Barrio Onassis - Tel: 454 38 88

Fuente: elaboración propia

4.5.Desarrollo Sprint Tres

Para el correcto desarrollo de este proyecto se realizaron 3 sprints, cada uno de 4 semanas, dividiendo en cada uno de los sprints entregas puntuales o avances del software que el cliente, en este caso la Droguería Onassis,representada por su administrador Ángel Alberto Rodríguez, revisó y aprobó el resultado de los entregables del proyecto; el siguiente es el desarrollo del sprint número tres en donde se culminará el desarrollo del software acordado con la Droguería Onassis.

4.5.1. Actualización sprint backlog

De acuerdo al proceso realizado del sprint No.2 se realiza la actualización del sprint backlog, culminando cada una de las tareas plasmadas. Adicionalmente, se plantean las nuevas tareas que ayudarán a la culminación del sprint No.3.

En la Figura 61 se relaciona la actualización y nuevas tareas del sprint backlog.

Figura 61

Sprint Backlog Ciclo 3

11,1	HU11	AGREGAR PRODUCTO PARA LA VENTA	Definir características que se implementaran para agregar un producto a la lista de venta	7	Felipe Rodriguez		Alta	100%	Se definen los pasos a seguir para poder realizar la accion de agregar un producto a la lista de venta	ok
11,2	HU11	AGREGAR PRODUCTO PARA LA VENTA	Desarrollo	7	Felipe Rodriguez	10	Alta	100%	validar que la funcionalidad de agregar un producto a la lista de venta se vea reflejada en el aplicactivo	ok
11,3	HU11	AGREGAR PRODUCTO PARA LA VENTA	Pruebas	7	Felipe Rodriguez	11,2	Media	100%	Validar que el programa no tenga fallos al momento de agregar un producto	ok
11,4	HU11	AGREGAR PRODUCTO PARA LA VENTA	Correcciones	7	Felipe Rodriguez	11,3	Alta	100%	Corregir los errores presentados en las pruebas implementadas	ok
12	HU12	GENERAR VENTA	GENERAR VENTA	28				100%		ok
12,1	HU12	GENERAR VENTA	Definir características que se implementaran para una venta	7	Felipe Rodriguez		Alta	100%	Se definen los pasos a seguir para poder realizaro generar la venta	ok
12,2	HU12	GENERAR VENTA	Desarrollo	7	Felipe Rodriguez	11	Alta	100%	validar que la funcionalidad de generar venta se vea reflejada en el aplicaitvo y base de datos	ok
12,3	HU12	GENERAR VENTA	Pruebas	7	Felipe Rodriguez	12,2	Media	100%	Validar que el programa no tenga fallos al momento de generar una venta	ok
12,4	HU12	GENERAR VENTA	Correcciones	7	Felipe Rodriguez	12,3	Alta	100%	Corregir los errores presentados en las pruebas implementadas	ok
Ciclo de desarrollo Sprint 3 (4 semanas)										
13	HU13	CONSULTAR REPORTE	CONSULTAR REPORTE	28				0%		ok
13,1	HU13	CONSULTAR REPORTE	Definir características que se implementaran para consultar un reporte de los medicamentos	7	Felipe Rodriguez		Alta	0%	Se definen los pasos a seguir para poder realizar o consultar un reporte de lo medicamentos vendidos	ok
13,2	HU13	CONSULTAR REPORTE	Desarrollo	7	Felipe Rodriguez	12	Alta	0%	validar que la funcionalidad de reporte se vea reflejada en el aplicaitvo y base de datos	ok
13,3	HU13	CONSULTAR REPORTE	Pruebas	7	Felipe Rodriguez	13,2	Media	0%	Validar que el programa no tenga fallos al momento de generar un reporte	ok
13,4	HU13	CONSULTAR REPORTE	Correcciones	7	Felipe Rodriguez	13,3	Alta	0%	Corregir los errores presentados en las pruebas implementadas	ok

Fuente: elaboración propia.

4.5.2. Iteración de desarrollo

Se realizó la tercera iteración de desarrollo, donde se implementó el módulo de reporte. Esto se efectuó mediante las siguientes herramientas:

- motor de base de datos realizado en MySQL mediante la herramienta phpMyAdmin.
- XAMPP sistema de gestión de base de datos MySQL e interprete de lenguajes PHP, desde allí gestionando phpMyAdmin.
- IDE NetBeans el cual permite la codificación en lenguaje Java en el que se basa el desarrollo del proyecto.
- GlassFish el cual es un servidor de código abierto para Java y funciona como un web server.
- Desarrollo de reporte

Para este módulo se creó la tabla correspondiente en la base de datos para cumplir con las carecterísticas solicitadas por la Droguería Onassis, la creación de esta tabla es con fin de realizar el módulo de reporte.

En la Figura 62 se relaciona la tabla corrrespondiente a reportes.

Figura 62

Tabla de Reportes

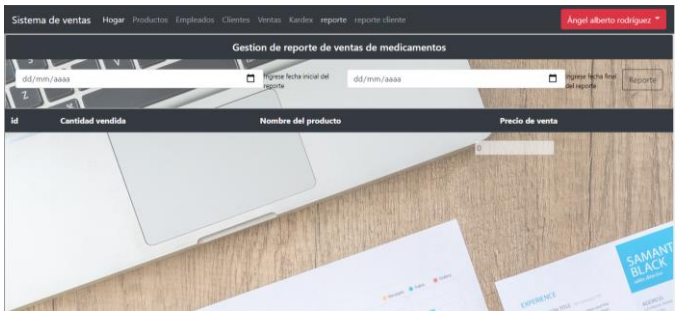
Fuente: elaboración propia.

El paso siguiente fue la creación de la vista de reporte, en donde se gestionan los reportes de venta de productos. Esto se realizó mediante jsp y diseño de bootstrap.

En la Figura 63 se relaciona la vista de reporte.

Figura 63

Vista de Reporte



Fuente: elaboración propia

Ya desarrollada la vista de reporte se prosiguió con el desarrollo de la parte lógica de la generación de reporte. El primer paso es capturar los datos cuando se genera una venta,esto lo realiza el controlador mediante un ArrayList de objetos de ventalos cuales fueron agregados a la venta; también es capturada la fecha actual en que se genera la venta de cada medicamento. Estos datos son enviados a la clase ReporteDAO mediante la función GuardarReporte(), ella es la encargada de envíar la siguiente sentencia a la base de datos:

- (insert into reportes (cantidadventa,producto,producto_id,fecha,precioproducto,factura) values(?,?,?,?,?,?)).

Lo anterior por cada medicamento que se agregó a la venta. El paso siguiente es la generación del reporte mediante el controlador que es el encargado de capturar los datos de la vista de reporte. Estos datos son guardados, se capturaran las fechas diligenciadas por el

usuario dependiendo de cuando quiera el reporte. Esto lo realiza el controlador mediante un request.getParameter() que recibe como parámetro el nombre del input de la vista del reporte donde el usuario esta seleccionando las fechas; ya cuando los datos son capturados estos son envíados a la clase ReporteDAO mediante las funciones Reporte() y ReporteTotalventa(). La primera función es la encargada de solicitar la información del reporte de las fechas ingresadas. Esto lo realiza mediante la siguiente sentencia envíada a la base de datos:

- (select sum(cantidadventa) as totalventa, sum(precioproducto*cantidadventa) as ventap,producto,producto_id,precioproducto from reportes where fecha between"+"'"+fechainicial+"'"+"and"+"'"+fechafinal+"'"+"group by producto_id desc).

El resultado de esta consulta es guardado en un ArrayList de objetos, el cual es retornado y el controlador lo envía a la vista para ser visualizado por el usuario mediante un request.setAttribute().

La siguiente función tiene como objetivo sumar el total de las ventas de cada medicamento para generar el total monetario de ventas del reporte. Esto se logra con la siguiente sentencia enviada a la base de datos:

- (select sum(cantidadventa) as totalventa, sum(precioproducto*cantidadventa) from reportes where fecha between"+"'"+fechainicial+"'"+"and"+"'"+fechafinal+"'"+"group by producto_id desc).

El resultado de esta consulta se guarda en un objeto de la clase Reporte, enseguida se guarda un atributo específico de este objeto en una variable entera el atributo que iremos sumando a esta variable es reporte.getPrecioProducto, el cual tiene el precio de venta de

cada medicamento del reporte, así obtendremos el valor total de las ventas del reporte, este atributo es retornado al controlador, en donde él lo envía a la vista para ser visualizado por el usuario mediante un request.setAttribute().

En la Figura 64 se relaciona la funcionalidad de GuardarReporte.

Figura 64

Función Guardar Reporte

```
public void GuardarReporte(Reporte reporte){
     String sentencia = "INSERT INTO reportes (cantidadventa,producto,producto_id,fecha,precioproducto,factura) VALUES(?,?,?,?,?,?)";
    con = cn.Conexion();
    try {
        ps = con.prepareStatement(sentencia);
        ps.setInt(1,reporte.getCantidadventa());
        ps.setString(2, reporte.getProducto());
        ps.setInt(3, reporte.getIdproducto());
        ps.setString(4, reporte.getFechaVenta());
        ps.setDouble(5, reporte.getPrecioProducto());
        ps.setString(6, reporte.getFactura());
        ps.executeUpdate();

    } catch (SQLException ex) {
        Logger.getLogger(VentaDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
```

Fuente: elaboración propia.

En la Figura 65 se relaciona la funcionalidad de Reporte.

Figura 65

Funcion de Reporte

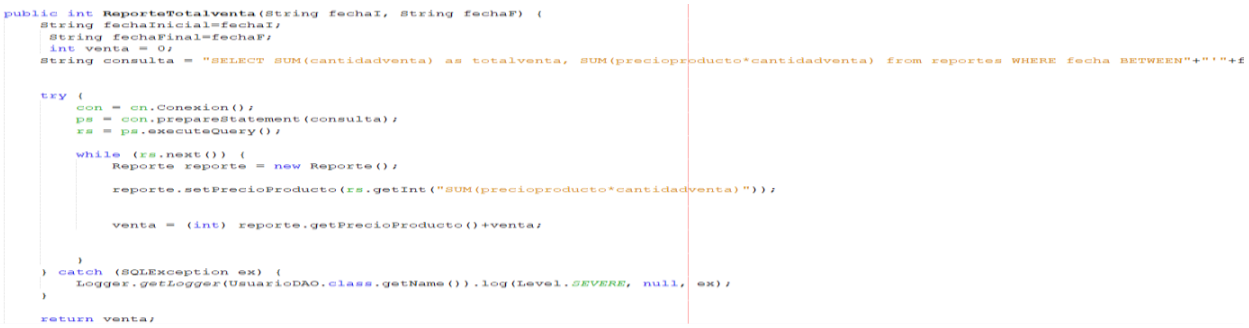
```
public List Reporte(String fechaI, String fechaF) {
    String fechaInicial=fechaI;
     String fechaFinal=fechaF;
    String consulta = "SELECT SUM(cantidadventa) as totalventa, SUM(precioproducto*cantidadventa) as ventap,producto,producto_id,preciop
    List<Reporte> lista = new ArrayList();

    try {
        con = cn.Conexion();
        ps = con.prepareStatement(consulta);
        rs = ps.executeQuery();
        while (rs.next()) {
            Reporte reporte = new Reporte();
            reporte.setIdproducto(rs.getInt("producto_id"));
            reporte.setProducto(rs.getString("producto"));
            reporte.setSalidas(rs.getInt("totalventa"));
            reporte.setPrecioProducto(rs.getInt("ventap"));


            lista.add(reporte);

        }
    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return lista;
```

Fuente: elaboración propia.

En la Figura 66 se relaciona la funcionalidad de ReporteTotalventa.

Figura 66

Funcion Reporte Total



```
public int ReporteTotalventa(String fechaI, String fechaF) {
    String fechaInicial=fechaI;
     String fechaFinal=fechaF;
     int venta = 0;
    String consulta = "SELECT SUM(cantidadventa) as totalventa, SUM(precioproducto*cantidadventa) from reportes WHERE fecha BETWEEN"+"'"+f

    try {
        con = cn.Conexion();
        ps = con.prepareStatement(consulta);
        rs = ps.executeQuery();

        while (rs.next()) {
            Reporte reporte = new Reporte();

            reporte.setPrecioProducto(rs.getInt("SUM(precioproducto*cantidadventa)"));


            venta = (int) reporte.getPrecioProducto()+venta;


        }
    } catch (SQLException ex) {
        Logger.getLogger(UsuarioDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return venta;
```

Fuente: elaboración propia

En la Figura 67 se relaciona la documentación de la prueba del módulo de reporte

Figura 67

Pruebas del Módulo de Reporte

Nombre del proyecto: GESTIÓN DE INVENTARIOS DROGUERÍA ONASSIS		**Caso No: 6**	
Nombre de caso de la prueba: funcionalidades Reporte			
Estado de la prueba: Finalizada		**Módulo de Reporte**	
Escrito por: Iván Felipe Rodríguez		**Ejecutado por:** Iván Felipe Rodríguez	
Descripción del caso de prueba: Se valida el correcto funcionamiento de las funcionalidades reporte			
Configuración de la prueba: Se requiere la ejecución del programa para ser dirigido al login ingresar a la opción de reporte			
flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Generar reporte	La aplicación deja ingresar los datos correspondientes a las fechas del reporte y al presionar el botón de reporte se listará el reporte de los medicamentos vendidos del rango de fechas	Exitoso
Excepciones			
1			Exitoso

Fuente: elaboración propia.

5. Resultados obtenidos

A continuación se muestra un resumen de los resultados que se obtuvieron con el desarrollo del software de inventarios planteado para solucionar las principales fallas que se evidenciaron en la DrogueríaOnassis, este con el fin de facilitar el manejo de los procesos administrativos del establecimiento. Se realizó la muestra del software en la Droguería Onassis con su aprobacion; este software cuenta con los siguientes módulos y funcionalidades:

- Se desarrollo la funcionalidad de validación de usuario para el ingreso de los empleados.
- Se valida el proceso de registro de los diferentes usuarios de acuerdo a su rol (Empleado, Cliente),
- Se implementan las funcionalidades de actualización y eliminación de usuarios.
- Se valida el proceso de registro de productos de acuerdo con las características solicitadas por la Droguería Onassis.
- Se implementan las funcionalidades de actualización y eliminación de productos.
- Se implementó el módulo de kárdex donde el usaurio puede visualizar el stock disponible de cada uno de los productos registrados. Adicionalmente, se puede visualizar los productos que se encuentran próximos a agotarse.
- Se implementó la funcionalidad de búsqueda de productos en específico, con el fin de tener conocimiento de su stock disponible al momento de gestionar algún pedido. Como valor adicional,el cliente puede realizar una búsqueda por el estado de los

productos (En stock, Poca existencia, Agotado), donde se visualizarán todos los productos con el estado seleccionado.

- Se desarrollo el módulo de venta, acompañado de su respectiva factura, la cual contiene información como: número de factura, item, código del producto, nombre, cantidad, subtotal y total. Adicional a esto, al momento de realizar cada venta el sistema internamente genera la actualización de las unidades disponibles de cada referencia. Para concluir la funcionalidad de este módulo se implementó la opción de anular una venta realizada. Al momento de generar este proceso, el kárdex de inventario se actualizára de manera inmedíata.
- Se implementó el módulo de reporte de venta de acuerdo al periodo que se requiera: de forma díaria, semanal o mensual. Este reporte muestra la cantidad y valor de cada uno de las referencias vendidas de acuerdo con el periodo de la consulta realizada.

La Figura 68 se relaciona con el requerimiento No.2: la validación de ingreso de los usuarios con el rol de empleado.

Figura 68

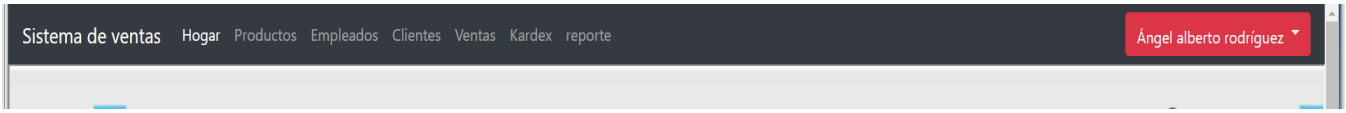
Vista de Ingreso

Fuente: elaboración propia.

En la Figura 69 se relaciona los módulos del sistema desarrollados, el módulo de registro de usuarios dividido en dos como se puede observar: cliente y empleado.

Figura 69

Opciones de Módulos



Fuente: elaboración propia.

En la Figura 70 se relaciona el requerimiento No.1: registro de usuarios de acuerdo a su rol (empleado, cliente).

Figura 70

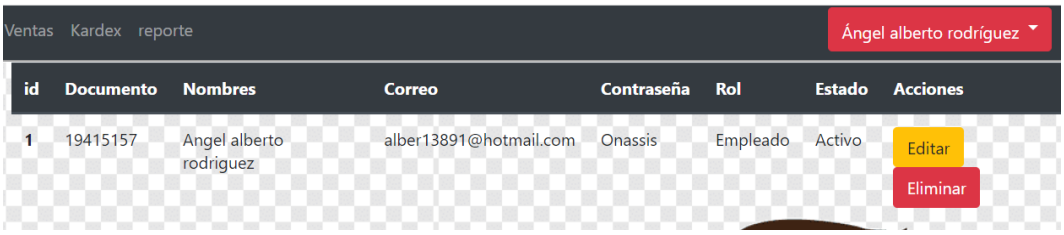
Vista de Registro de Empleados

Fuente: elaboración propia.

En la Figura 71 se relaciona con el requerimiento No.3: funcionalidades de eliminar y actualizar usuarios (empleado,cliente).

Figura 71

Funcionalidades de Eliminar y Actualizar

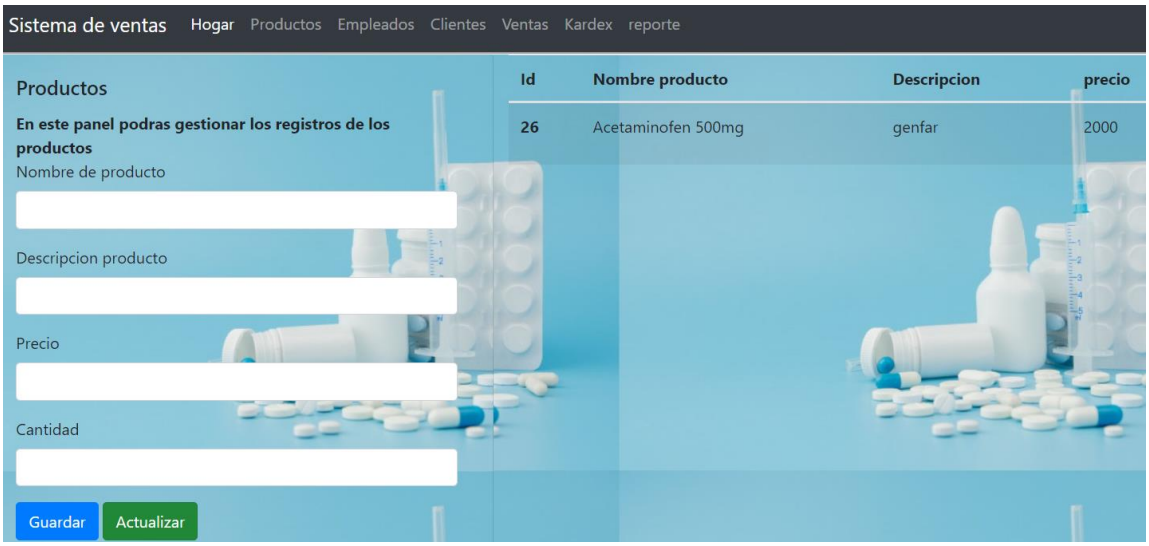


Fuente: elaboración propia.

En la Figura 72 se relaciona el requerimiento No.4: registro de productos de acuerdo con lo especificado por la Droguería Onassis

Figura 72

Vista de Registro de Productos



Fuente: elaboración propia.

En la Figura 73 se relaciona el requerimiento No.5: funcionalidades de eliminar y actualizar prodcutos

Figura 73

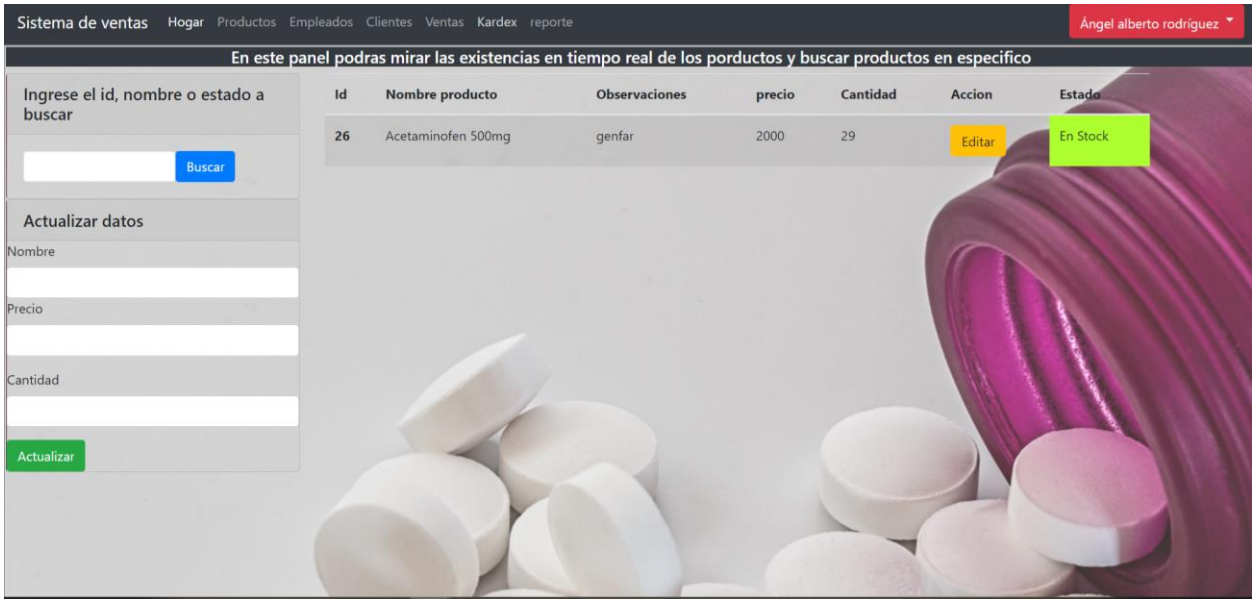
Funcionalidades de Eliminar y Actualizar

Fuente: elaboración propia.

En la Figura 74 se relaciona el requerimiento No.6: visualización de los productos con su stock actual, precio y búsqueda de un producto en específico

Figura 74

Módulo Kárdex

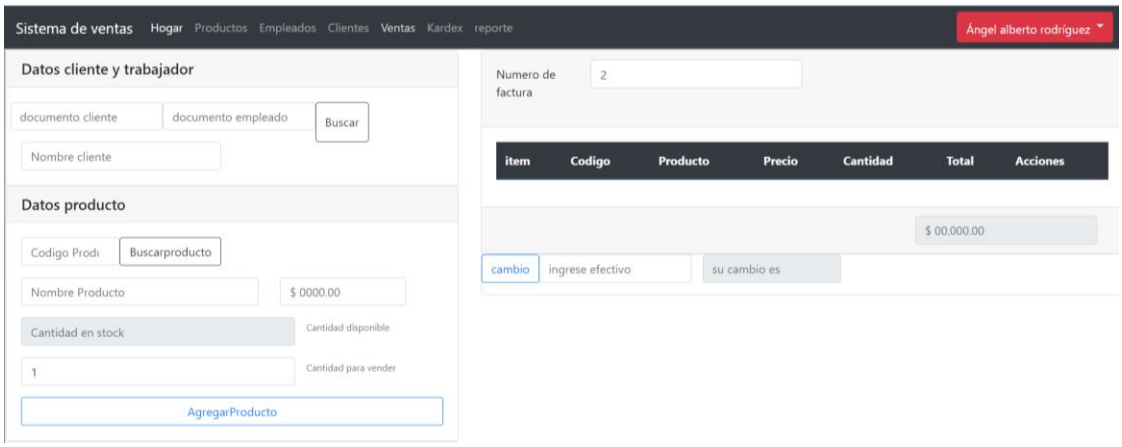


Fuente: elaboración propia.

La Figura 75 se relaciona con el requerimiento No.7: facturación de los productos vendidos.

Figura 75

Módulo de venta



Fuente: elaboración propia.

La Figura 76 se relaciona con el requerimiento No.8: generar un reporte de la cantidad de los productos vendidos ya sea (díaria, semanal o mensual).

Figura 76

Módulo de Reporte

Fuente: elaboración propia.

6. Conclusiónes y recomendaciones

El objetivo de este capítulo es compartir las conclusiónes y recomendaciones obtenidas con el resultado del desarrollo del software de gestión de inventarios para la Droguería onassis, que corresponde a un sistema que brinda mejorar la gestión de inventarios, control de los clientes y reporte de ventas. Adicional a esto, la principal fuente de motivación para el desarrollo de este proyecto es que se identificó desde el inicio ya que la Droguería Onassis cuenta con una antigüedad considerable y sus fundadores son actualmente los encargados de la atención. Este aspecto fue de mayor impacto ya que se quiere romper el paradigma que la tecnología no esta disponible para todo el público y el pánico de algunas personas a ir de la mano con la tecnología.

6.1.Cumplimiento de los Objetivos

En el transcurso de estas 12 semanas se obtuvo el desarrollo de un software capacitado para llevar la correcta gestión de los productos y ventas realizadas por la Droguería Onassis. Esto de una manera óptima, poco compleja para el usuario, reduciendo tiempos de verificacion manual realizados por ellos, eliminando la poca legibilidad de los inventarios y reprocesos.

6.2.Cumplimiento de los Objetivos Específicos

- Se obtuvo el correcto proceso de verificación, identificación y documentación de los requerimientos solicitados por la Droguería Onassis, los cuales permitieron generar una solución óptima y adecuada a las problemáticas presentadas en la Droguería Onassis

- Se realizó el correcto diseño de la aplicación mediante la documentación de Diagramas, la cual permitió el desarrollo del software con las siguientes tecnologías; lenguaje Java mediante el framework de NetBeans, jsp para el diseño de su interfaz y su estructura de datos mediante MySQL gestionada mediante phpMyAdmin. Se implementó de manera adecuada la metodología Scrum, que permitió tener un control adecuado del desarrollo de cada módulo y funcionalidad del proyecto. Esto con el fin de mantener bajo supervisión cualquier novedad, ya que el fin de esta metodología es planificar y controlar el desarrollo del proyecto.
- Se gestionaron las pruebas funcionales las cuales evidenciaron el correcto desarrollo del software en cada uno de los requerimientos solicitados por el usuario; esto mediante pruebas de escritorio.

6.3.Beneficios obtenidos

- El área administrativa del establecimiento minimizará los tiempos invertidos en los procesos de inventarios y reportes de venta gracias al software desarrollado, mediante la confiabilidad de la información registrada.
- Con el software desarrollado, la Droguería Onassis no obtuvo la necesidad de realizar una inversión económica para suplir las necesidades administrativas evidenciadas por ellos.
- Como valor agregado se logró implementar un reporte de ventas que le permitirá al administrador tener un control inmediato sobre las ventas realizadas. Adicional a esto se implementó un reporte de compras realizadas

por los clientes que le permitirá al administrador identificar sus clientes más potenciales y así brindarles los beneficios correspondientes. Mediante el registro de clientes se contiene la garantía de poder utilizar esta información para futuras funcionalidades para la Droguería Onassis.

6.4.Trabajo Futuro

Teniendo en cuenta el desarrollo de este proyecto y por los tiempos de entrega del mismo, se plantean mejoras al software a un futuro, permitiendo que el sistema evolucione y facilite más las actividades realizadas por la Droguería Onassis. Algunas de las mejoras que ya se tienen planteadas son:

- Realizará la implantación en el software en donde el cliente tenga acceso para conocer sobre promociones, realizar pedidos y generar consultas.
- Se desarrollará el control de los medicamentos por medio de las fechas de vencimiento generando alertas pertinentes.
- Se realizará el proceso para que el cliente pueda ingresar a la aplicación de una manera más sencilla.
- Se realizará el diseño de imágenes personalizadas para la aplicación referente a cada módulo.
- Se realizará un modulo de gestion contable.

6.5.Recomendaciones

Se realizan las siguientes recomendaciones a la Droguería Onassis con el fin de minimizar incovenientes que se puedan presentar:

- Se recomienda no comercializar el software de inventarios diseñado para ellos, ya que este se adecúa a las necesidades presentadas en el establecimiento.
- Se recomienda leer el manual de usuario en caso de no acordarse de algún procedimiento del software.
- Se recomienda que las actualizaciones al software se basen o sean guiadas según el manual técnico y por personas con los conocimientos necesarios.

7. Referencias Bibliográficas

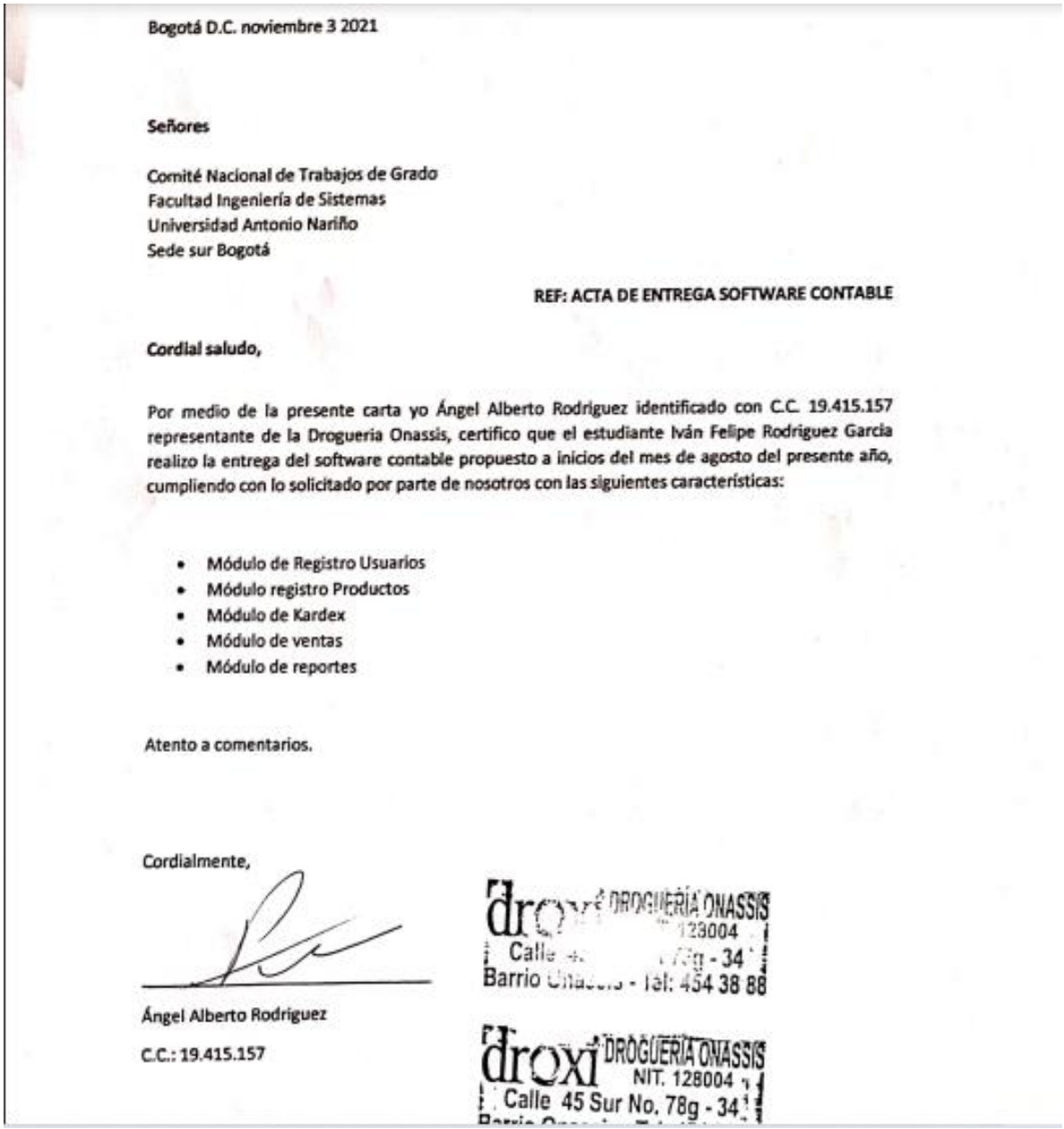
- DURAN, YOSMARY (2012). Administración del inventario: elemento clave para la optimización de las utilidades en las empresas. (2012). Visión Gerencial, 0(1), 55–78.
- Albaladejo, X. (2008). Que es SCRUM. In Proyectos Ágiles (p. 1). https://proyectoságiles.org/que-es-Scrum/.
- CECOLDA - Centro Colombiano del Derecho de Autor - LEY 23 DE 1982 SOBRE DERECHO DE AUTOR. (n.d.). Retrieved March 17, 2021, from http://www.cecolda.org.co/index.php/derecho-de-autor/normás-y-jurisprudencia/normás-nacionales/124-ley-23-de-1982-sobre-derecho-de-autor
- CILSA. (2017). ¿Qué es un programa? | Desarrollar Inclusión. https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-programa/
- Clarís. Pau. (2015). Proceso y Roles de Scrum. In Softeng. https://www.softeng.es/es-es/empresa/metodologías-de-trabajo/metodología-Scrum/proceso-roles-de-Scrum.html
- Consoft - Características de Farmatic. (n.d.). Retrieved March 25, 2021, from https://www2.consoft.es/farmatic/características-de-farmatic/
- Definición de Droguería» Concepto en Definición ABC. (n.d.). Retrieved March 24, 2021, from https://www.definicionabc.com/ciencia/droguería.php
- Evan, A. (2015). Java significado: ¿qué es Java y para qué sirve? - Tokio. https://www.tokioschool.com/noticias/Java-significado-que-es-Java/
- Formulación magistral | BitFARMA, programa informático de gestión para Oficina de Farmacia. (n.d.). Retrieved March 25, 2021, from https://www.bitfarma.com/
- Fundamentos de Scrum | proyectos Ágiles. (n.d.). Retrieved March 18, 2021, from https://proyectoságiles.org/fundamentos-de-Scrum/
- Legal Team Workers. (n.d.). Un Resumen Útil de la Ley de Protección de Datos Personales | L T W Abogados | Colombia. Retrieved March 17, 2021, from https://abogadocolombia.wordpress.com/2017/07/22/un-resumen-util-de-la-ley-de-proteccion-de-datos-personales/

- López Mendoza, M. (2020). Qué es un lenguaje de programación | OpenWebinars. 16 de Julio de 2020. https://openwebinars.net/blog/que-es-un-lenguaje-de-programación /
- ORACLE. (2020). ¿Qué es una base de datos relacional? | Oracle Colombia. https://www.oracle.com/co/database/what-is-a-relational-database/
- Pahino, R. (2020). ¿Qué son Spring framework y Spring Boot? Tu primer programa Java con este framework | campusMVP.es. 2020. https://www.campusmvp.es/recursos/post/que-son-spring-framework-y-spring-boot-tu-primer-programa-Java-con-este-framework.aspx
- Requena Mesa, A. (2018). Qué es un Sprint de Scrum. In OpenWebinars.net. https://openwebinars.net/blog/que-es-un-sprint-Scrum/
- Roche, J. (2020). Artefactos Scrum: las 3 herramientas clave de gestión.
- Saiz, J. (2017). Proyecto Scrum. Una explicación sencilla de la metodología • Jorge Saiz. https://jorgesaiz.com/blog/proyecto-Scrum-una-explicacion-sencilla/
- SIKI Software ERP | Planes y PRECIOS Colombia. (n.d.). Retrieved March 25, 2021, -{from https://sikisoftware.com/planes-precios-colombia/
- Software para Droguerías y Farmacias | j4Pro Administración Inteligente. (n.d.). Retrieved March 25, 2021, from https://j4pro.com/software-para-droguerías-y-farmacias
- Software para Farmacias Descarga Gratis | ManagementPro ®. (n.d.). Retrieved March 25, 2021, from https://www.mproerp.com/software-para-farmacias-descarga-gratis/
- Valdés, D. (2007). ¿Qué son las bases de datos? 26/10/2007, 1. http://www.maestrosdelweb.com/que-son-las-bases-de-datos/

8. Anexos

8.1 Anexo A

A. Se anexa carta de entrega de software a los administradores de la droguería Onassis con cada una de las características tal como se relaciona a continuación.



Bogotá D.C. noviembre 3 2021

Señores

Comité Nacional de Trabajos de Grado
Facultad Ingeniería de Sistemas
Universidad Antonio Nariño
Sede sur Bogotá

REF: ACTA DE ENTREGA SOFTWARE CONTABLE

Cordial saludo,

Por medio de la presente carta yo Ángel Alberto Rodriguez identificado con C.C. 19.415.157 representante de la Drogueria Onassis, certifico que el estudiante Iván Felipe Rodriguez Garcia realizo la entrega del software contable propuesto a inicios del mes de agosto del presente año, cumpliendo con lo solicitado por parte de nosotros con las siguientes características:

- Módulo de Registro Usuarios
- Módulo registro Productos
- Módulo de Kardex
- Módulo de ventas
- Módulo de reportes

Atento a comentarios.

Cordialmente,

Ángel Alberto Rodriguez

C.C.: 19.415.157

droxi DROGUERÍA ONASSIS NIT. 128004 Calle 45 Sur No. 78g - 34 Barrio Onassis - Tel: 454 38 88

droxi DROGUERÍA ONASSIS NIT. 128004 Calle 45 Sur No. 78g - 34

Fuente: Autoría propia.

8.2 Anexo B

B. Se anexa carta aval dirigida al comité nacional de grado.

Bogotá D.C 23 de febrero 2021

Señores:

Comité nacional de trabajos de grado
Facultad Ingeniería de sistemas
Universidad Antonio Nariño
Sede sur Bogotá

Ref. Aval Propuesta de grado

Cordial saludo:

La presente carta es para constar que la Droguería Onassis con su representante legal **Ángel Alberto Rodríguez García** Identificado con cedula de ciudadanía 19.415.157 de Bogotá D.C avalan el proyecto de grado de desarrollar un Software de gestión de inventarios para la Droguería Onassis el cual brindaran permiso al estudiante Iván Felipe Rodríguez García estudiante de ingeniería de sistemas y computación de la Universidad Antonio Nariño, que por solicitud de ellos desean que se les avale el desarrollo del software de gestión de inventarios sin ningún beneficio o remuneración económica

De acuerdo a lo acordado con el estudiante, el proyecto contempla la implementación de las siguientes funcionalidades

- El sistema le permitirá al usuario el registro o entrada de un producto con sus respectivas características las cuales serán brindadas por la droguería Onassis
- El sistema le permitirá Salida de un producto Sin facturación
- El sistema le permitirá Modificación de un producto
- El sistema le permitirá Búsqueda de un producto
- El sistema le permitirá eliminar un producto del inventario

Este proyecto es de suma importancia ya que la droguería Onassis no tiene una herramienta o solución informática que le permita manejar de manera adecuada su gestión de productos lo cual los esta afectando en su día a día

Atentamente:

Ángel Alberto Rodríguez
C.C 19.415.157

droxi DROGUERÍA ONASSIS
NIT. 128004
Calle 45 Sur No. 78g - 34
Barrio Onassis - Tel: 454 38 88

Fuente: autoría propia