UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA

PROYECTO DE GRADO

”APLICACIÓN WEB PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS”

CASO: ASOCIACION DE PRODUCTORES EN METALMECANICA SUR “METAL SUR”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE : HUGO HERNAN AQUINO ACHUMIRI

TUTOR METODOLOGICO : M.Sc. FRANZ CUEVAS QUIROZ

ASESOR : M.Sc. CARLOS MULLISACA CHOQUE

La Paz – Bolivia

2018

UNIVERSIDAD MAYOR DE SAN ANDRES

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMATICA

LA CARRERA DE INFORMATICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRES AUTORIZA EL USO DE LA INFORMACION CONTENIDA EN ESTE DOCUMENTO SI LOS PROPOSITOS SON ESTRICTAMENTE ACADEMICOS

LICENCIA DE USO

El usuario está autorizado a:

a. Visualizar el documento mediante el uso de un ordenador o dispositivo móvil
b. Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c. Copiar textualmente parte(s) de su contenido mencionado la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS, EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR

DEDICATORIA

A mis padres Cruz Aquino y Juana Achumiri por todo el amor, orientación y apoyo incondicional a lo largo de toda mi vida.

A toda mi familia por apoyarme en los momentos buenos y malos de mi vida, en especial a mis hermanos Freddy por sus incansables palabras de aliento y a Angélica por cuidarme desde el cielo

A mi esposa Gloria y a mis hijos Juan David y Matías

AGRADECIMIENTOS

En primer lugar agradezco a Dios por permitir lograr un objetivo más en mi vida

A mi tutor metodológico M.Sc. Franz Cuevas Quiroz por su enseñanza y paciencia al orientarme y guiarme a lo largo del desarrollo de este proyecto.

A mi asesor M.Sc. Carlos Mullisaca Choque por guiarme y darme su apoyo incondicional durante el desarrollo de este proyecto.

A la Universidad Mayor de San Andrés y a todos los docentes de la carrera de informática.

A la Asociación de Productores en Metalmecánica Sur por permitirme desarrollar este proyecto.

Y agradecer a mis padres, hermanos, esposa, hijos y amigos por su apoyo.

A todos ellos muchísimas gracias…

INDICE DE CONTENIDO **PAG.**

CAPTITULO I
MARCO INTRODUCTORIO

1.1 Introducción 1

1.2 Antecedentes 2

1.2.1 Antecedentes institucionales 2

1.2.2 Antecedentes de proyectos anteriores 3

1.3 Planteamiento del problema 4

1.4 Definición de objetivos 5

1.4.1 Objetivo general 5

1.4.2 Objetivos específicos 5

1.5 Justificación 5

1.6 Alcances y limites 6

1.6.1 Alcances 6

1.6.2 Limites 6

CAPITULO II
MARCO TEORICO

2.1 Definición de inventario 7

2.1.1 Modelo de inventarios deternimistico 7

2.1.2 Modelo de inventario probabilístico 9

2.1.3 Clasificación de inventarios 10

2.1.4 Técnicas de valuación de inventario 10

2.2 Compra y venta 11

2.3 Ingeniería del software 12

2.4 Metodología de desarrollo de software 12

2.5 Metodologías agiles 12

2.6 Programación extrema (XP) 13

2.6.1 Valores XP 14
2.6.2 Proceso XP 15
2.6.3 Equipo de trabajo XP 20
2.7 UML (Lenguaje Unificado de Modelado) 21
2.8 Definición de aplicación web 26
2.9 Calidad de del software 27
2.10 Seguridad del software 28
2.11 Herramientas de desarrollo 29

CAPITULO III
MARCO APLICATIVO

3.1 Análisis de situación actual 31
3.2 Fase de planificación 32
3.2.1 Especificación de requerimientos 33
3.2.2 Requerimientos funcionales 33
3.2.3 Historias de usuario 35
3.2.4 Planificación de iteraciones 39
3.3 Fase de diseño 40
3.4 Fase de desarrollo 44
3.4.1 Primera iteración 45
3.4.2 Segunda iteración 52
3.4.3 Tercera iteración 61
3.5 Seguridad 69
3.5.1 Seguridad a nivel de aplicación 69
3.5.2 Seguridad a nivel de la base de datos 69
3.6 Calidad 69

CAPITULO IV
CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones..... 73

4.2 Recomendaciones 74

BIBLIOGRAFIA

ANEXOS

Anexo A Casos de uso

Anexo B Diagramas de secuencia

Anexo C Diagramas de estado

INDICE DE FIGURAS PAG.

DESCRIPCION

Figura 1.1 Estructura organizacional de la Asociación 2

Figura 2.1 Comportamiento del modelo clásico de inventario..... . 8

Figura 2.2 Proceso de desarrollo XP 15

Figura 2.3 Representación de una clase..... 22

Figura 2.4 Diagrama de clases..... 22

Figura 2.5 Diagrama de casos de uso 23

Figura 2.6 Diagrama de secuencia..... 24

Figura 2.7 Diagrama de actividades 25

Figura 3.1 Proceso actual de comercialización METAL SUR..... 31

Figura 3.2 Diagrama de clases propuesto..... 41

Figura 3.3 Modelo físico de la base de datos 42

Figura 3.4 Diseño de la estructura y composición de la aplicación 43

Figura 3.5 Captura del pantalla del inicio de sesión del sistema..... 49

Figura 3.6 Captura del pantalla bienvenida al sistema 49

Figura 3.7 Creación de tablas en la base de datos Ferretería..... 50

Figura 3.8 Código php para validar usuario en el sistema 50

Figura 3.9 Captura de pantalla crear usuarios y rol..... 51

Figura 3.10 Administrar Clientes 57

Figura 3.11 Adicionar nuevo cliente 57

Figura 3.12 Código php adicionar clientes a la base de datos 58

Figura 3.13 Administrar proveedores..... 58

Figura 3.14 Adicionar proveedor..... 59

Figura 3.15 Código php adición de proveedor a la base de datos 59

Figura 3.16 Listado de productos de la categoría productos planos..... 65

Figura 3.17 Adicionar nuevo producto 65
Figura 3.18 Ingreso a opción ventas – ultimas ventas 66
Figura 3.19 Generar nueva venta 66
Figura 2.20 Código php para generar ultimas ventas 67

INDICE DE TABLAS **PAG.**

DESCRIPCION

Tabla: 2.1 Modelo de tarjeta CRC 18
Tabla: 2.2 Herramientas de desarrollo 30
Tabla: 3.1 Historias de usuario propuesto 35
Tabla: 3.2 Historia de usuario control de acceso a usuario 35
Tabla. 3.3 Historia de usuario creación de roles personalizados 36
Tabla: 3.4 Historia de usuario gestión de clientes 37
Tabla: 3.5 Historia de usuario gestión de proveedores 37
Tabla: 3.6 Historia de usuario gestión de productos 38
Tabla: 3.7 Historia de usuario gestión de ventas 38
Tabla: 3.8 Plan de iteraciones 40
Tabla: 3.9 Tarea lectura de usuario y contraseña 45
Tabla: 3.10 Tarea comprobación del usuario en la base de datos 46
Tabla: 3.11 Tarea acceso a menús del sistema 46
Tabla: 3.12 Tarea cuenta administrador 47
Tabla: 3.13 Tarea cuenta vendedor 47
Tabla: 3.14 Tarjeta CRC usuario 48
Tabla: 3.15 Tarjeta CRC roles 48
Tabla: 3.16 Prueba de aceptación primera historia de usuario 51
Tabla: 3.17 Prueba de aceptación segunda historia de usuario 52
Tabla: 3.18 Tarea adaptar la base de datos para clientes 53
Tabla: 3.19 Tarea diseño de interfaz de usuario para clientes 53
Tabla: 3.20 Tarea administrar clientes 54
Tabla: 3.21 Tarea adaptar la base de datos para proveedores 54
Tabla: 3.22 Tarea diseño de interfaz de usuario para proveedores 55

Tabla: 3.23 Tarea administrar proveedores 55
Tabla: 3.24 Tarjeta CRC proveedores 56
Tabla: 3.25 Tarjeta CRC clientes 56
Tabla: 3.26 Prueba de aceptación tercera historia de usuario..... 60
Tabla: 3.27 Prueba de aceptación cuarta historia de usuario..... 60
Tabla: 3.28 Tarea diseño de interfaz grafica para productos..... 61
Tabla: 3.29 Tarea adaptar la base de datos para almacenar los productos 62
Tabla: 3.30 Tarea administrar productos..... 62
Tabla: 3.31 Tarea diseño de interfaz grafica para ventas 63
Tabla: 3.32 Tarea creación de la base de datos para ventas 63
Tabla: 3.33 Tarjeta CRC productos..... 64
Tabla: 3.34 Tarjeta CRC ventas 64
Tabla: 3.35 Prueba de aceptación gestión de productos..... 67
Tabla: 3.36 Prueba de aceptación gestión de ventas 68

RESUMEN

Los sistemas de información se han convertido en una herramienta fundamental para el crecimiento y desarrollo de una grande, mediana o pequeña empresa. Además un sistema de información nos permite tener un control sobre el inventario con el que se cuenta y así saber a todo momento que es lo que se tiene y que es lo que se necesita, para lograr una mejora en el servicio a los clientes, un incremento en las ventas y mejor administración de los recursos económicos de la empresa o negocio.

METAL SUR es una Asociación de Productores en Metalmecánica, se encuentra ubicada en la ciudad de El Alto – Bolivia. Cuenta con un almacén de ferretería para proveer los materiales y artículos a los miembros de esta asociación para la elaboración y producción de: Barandas metálicas, persianas metálicas, metal forjado, puertas de garaje, puertas de ingreso de metal y madera, rejas plegables, mamparas, hojalatería y otros trabajos en general. En la asociación se desarrolló e implementó una aplicación web que permite realizar el registro de sus compras, ventas y realizar un control adecuado de su inventario de forma automatizada.

Con la implementación del nuevo sistema, se logro mejorar el proceso de registro de compras, ventas y el manejo de su inventario de los materiales y artículos existentes. Toda esta información se encuentra almacenada en una base de datos, además se logro reducir el tiempo en la elaboración de reportes con respecto al número de existencias en el inventario.

Para alcanzar este objetivo, nos apoyamos en la metodología de desarrollo de software ágil XP (Xtreme Programming - Programación Extrema) por su gran adaptabilidad en todo el proceso de desarrollo, cuenta con un modelo de desarrollo incremental con entregas rápidas y retroalimentación diaria por parte del cliente.

ABSTRACT

Nowadays, information systems have become a fundamental tool for the growth and development of a large, medium or small company. In addition, an information system allows us to have control over the inventory that is counted and to know at all times what it is that we have and what is needed, to achieve an improvement in customer service, an increase in sales and better management of the economic resources of the company or business.

METAL SUR is an Association of Producers in Metalworking, located in the city of El Alto - Bolivia. It has a hardware store to provide materials and articles to the members of this association for the production and production of: Metal rails, metal shutters, forged metal, garage doors, metal and wood entrance doors, folding bars, screens, tinwork and other jobs in general. In the association, a web application was developed and implemented that allows to register your purchases, sales and to carry out an adequate control of your inventory in an automated way.

With the implementation of the new system, it was possible to improve the process of registering purchases, sales and the management of its inventory of existing materials and articles. All this information is stored in a database; in addition it was possible to reduce the time in the elaboration of reports with respect to the number of inventories in the inventory.

To achieve this goal, we rely on the software development methodology agile XP (Extreme Programming) for its great adaptability throughout the development process, has an incremental development model with fast deliveries and daily feedback by the client.

CAPITULO I MARCO INTRODUCTORIO

1.1 INTRODUCCION

En la actualidad, los sistemas de información se han convertido en un factor importante en el desarrollo de una empresa a nivel mundial, debido a la cantidad de información que se maneja dentro de la misma, por esta razón es importante que el tratamiento automático de la información sea realmente útil y necesario. Además los sistemas de información están basados en computadoras que son objetos de gran consideración en la toma de decisiones oportunas, confiables y efectivas en cuanto a técnicas de planificación, programación y administración con el fin de garantizar su éxito, reducir costos y riesgos e incrementar las ganancias.

Por esta razón, nace la necesidad de automatizar las actividades cotidianas en las organizaciones, cabe mencionar el vertiginoso avance de las telecomunicaciones y el progreso que han experimentado las ciencias informáticas que obliga a entrar al moderno mundo de la tecnología, ser competitivos y no quedarse relegados en las tareas que proporcionan beneficios para proyectarse al futuro.

Así mismo en nuestro país, al igual que el resto del mundo, se van impulsando profundos cambios en las empresas de tal forma que vienen desarrollando sistemas de información automatizados para el mejor tratamiento de su información, así ofrecer a la sociedad más y mejores productos y servicios que impulsen el desarrollo.

Teniendo en cuenta todos los beneficios y oportunidades que brinda un sistema de información. En la Asociación de Productores en Metalmecánica Sur “METAL SUR” se

desarrollara e implementara un sistema de información que permita realizar el registro de sus compras, ventas e inventarios de forma automatizada.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INTITUCIONALES

La Asociación de Productores en Metalmecánica Sur "METAL SUR" fue fundada el 6 de septiembre de 2008, con personería jurídica Nº 039/2010 Afiliado a CO.CE.DAL—C.O.R. El Alto. Se encuentra establecido en la zona Villa Caluyo, entre calle 6 y calle C de la ciudad de El Alto-La Paz. Actualmente cuenta con cuarenta y cuatro socios.

METAL SUR es una asociación de productores en metalmecánica y ofrece a la población trabajos en metal de calidad como ser: Barandas metálicas, persianas metálicas, metal forjado, puertas de garaje, puertas de ingreso de metal y madera, rejas plegables, mamparas, hojalatería y otros trabajos en general.

Por otra parte la asociación cuenta con un almacén de ferretería la misma que está destinada a la venta de toda clase de materiales para la elaboración de productos terminados ya anteriormente mencionados. Materiales como: tubos, angulares, flejes, planchas, perfiles, chapas, bisagras, brocas, discos de corte, clavos, pernos, etc. Cada una de ellas en sus diferentes variedades. Estos materiales se venden al público en general y en particular a los asociados a un precio más económico, con el fin de conseguir ingresos adicionales para todos los socios.

La administración de esta gran ferretería se encuentra bajo el personal autorizado por dicha asociación. Motivo por el cual se hace notar que esta asociación carece de mecanismos de registro y control de sus compras, ventas y la administración de su inventario. La asociación de productores en metal mecánica cuenta con la siguiente estructura organizacional.

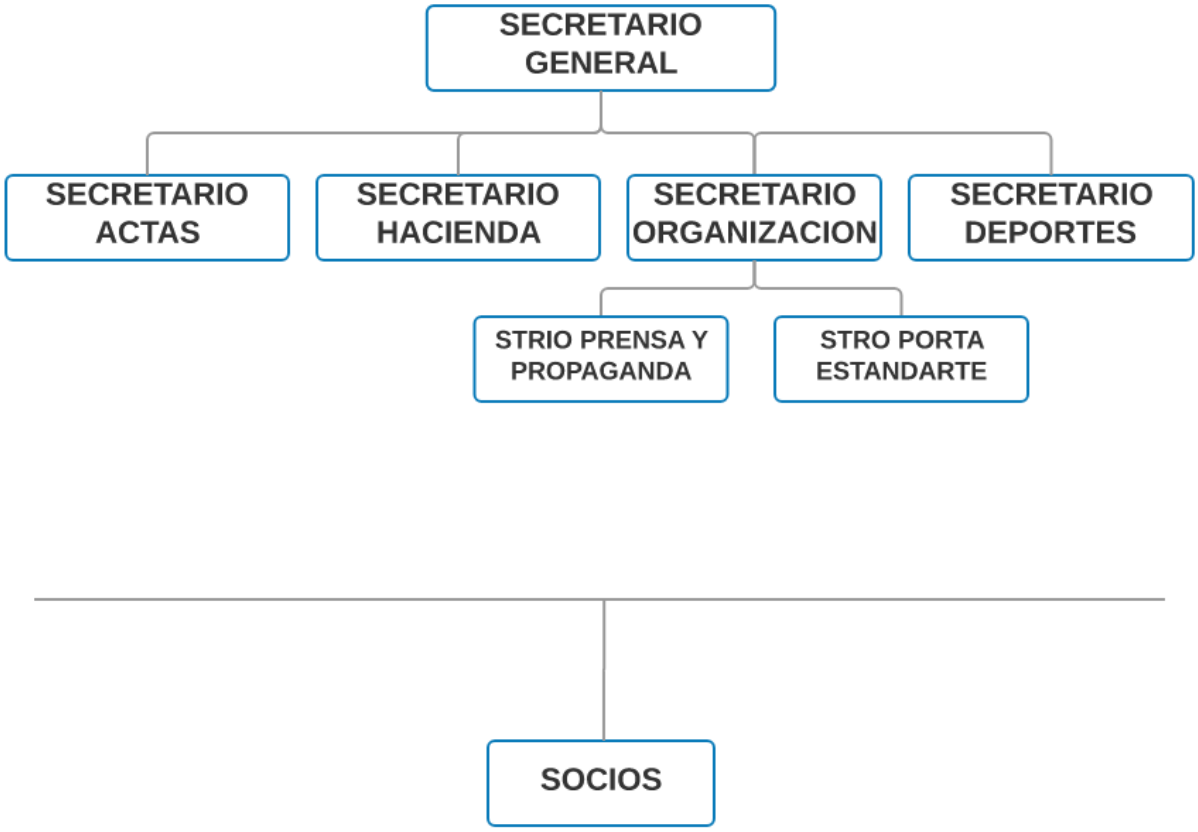


Figura 1.1: Estructura organizacional Metal Sur
Fuente: (Metal Sur, 2018)

1.2.2 ANTECEDENTES DE PROYECTOS ANTERIORES

Existen varios proyectos de grado ya realizados en gestiones pasadas en nuestra universidad con características similares, de ellos podemos mencionar los siguientes.

El proyecto “CONTROL DE VENTAS E INVENTARIOS PARA EL MONITOREO DE PEDIDOS” fue desarrollado para llevar el registro correcto y automatizado del control de ventas e inventario, así saber el ingreso y egreso de productos, de acuerdo a ello poder distribuir de manera eficiente los pedidos realizados por la empresa distribuidora VMCC logrando la empresa mejorar sus ingresos. Se implemento los módulos de registro de proveedores los productos que ofrecen, modulo de control de inventario a través del modulo de compras, del modulo de ventas y el modulo de pedido (Cruz, 2013).

El proyecto “SISTEMA WEB DE COMPRAS, VENTAS E INVIENTARIOS” este proyecto fue implementado en la empresa EDDYMAR este proyecto es un sistema web apoyado en las técnicas y procesos de Control de Compras, Ventas e Inventario con lo cual se busca mejorar el funcionamiento de la empresa “EDDYMAR”, en los procesos de control de compras, ventas e inventario. El sistema brinda una serie de funcionalidades que fueron plasmadas en historias de usuarios, estas funcionalidades buscan satisfacer los requerimientos de la empresa y los funcionarios internos (Rojas, 2014).

El proyecto “SISTEMA WEB DE CONTROL DE COMPRAS E INVENTARIOS”, fue implementado en la Comercial Ariana está ubicada en la ciudad de La Paz y Santa Cruz, donde su sucursal principal se encuentra en la ciudad de La Paz, ofrece diversos productos de belleza y cuidado personal de origen Americano, para niños, mujeres y hombres. Así como muchas empresas fue creciendo a través del tiempo, así mismo fue incrementándose la cantidad de información que maneja en sus distintas áreas organizacionales, por lo que surgió la necesidad de automatizar y centralizar la información sobre sus compras, ventas e inventarios.(Carrillo, 2017).

1.3 PLANTEAMIENTO DEL PROBLEMA

Luego de realizar una serie de visitas a la ASOCIACION DE PRODUCTORES EN METALMECANICA SUR “METAL SUR” y realizar las respectivas entrevistas con el personal encargado de dicha asociación se pudo evidenciar los siguientes problemas que mencionamos a continuación.

- ✓ El registro de la información generada con respecto a las compras, ventas e inventarios se lo realiza de forma manual. Esto provoca una posible pérdida de documentos y en consecuencia perdida de información.
- ✓ El reabastecimiento de materiales se pronostica de manera tardía, generando de esta manera pérdidas económicas a la asociación y causando mal estar en los socios.
- ✓ El directorio se encuentra desinformada al carecer de información precisa a la hora de tomar decisiones.

- ✓ El control manual de compras y ventas de los materiales genera riesgos de pérdida de información sobre el inventario debido a la gran cantidad de productos en almacén.
- ✓ Se ha presentado perdida de materiales, originado principalmente por la falta de políticas de control y un sistema de administración de inventario.
- ✓ La asociación carece de un registro de asociados y de los clientes de la asociación.

Luego del estudio realizado y expuesto anteriormente surge la siguiente interrogante.

¿La implementación de una aplicación web podrá mejorar el proceso de registro y control de compras, ventas e inventarios de la asociación "METAL SUR"?.

1.4 DEFINICION DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar una aplicación web que nos permita realizar el registro y control de las compras, ventas e inventarios de forma automatizada para la asociación de productores en metalmecánica sur "METAL SUR", en base a criterios y tecnologías de información adecuados a las necesidades de la asociación.

1.4.2 OBJETIVOS ESPECIFICOS

- ✓ Analizar todas las actividades, acciones y tareas que se realizan dentro de la asociación, así obtener toda la información necesaria para el desarrollo del proyecto.
- ✓ Diseñar una base de datos, para que sirva de repositorio de los datos que serán utilizados por el sistema de información.
- ✓ Desarrollar los módulos de compras, ventas e inventarios.
- ✓ Generar reportes de ventas, compras diarios, semanales, mensuales o anuales y de acuerdo a un intervalo de tiempo.
- ✓ Diseñar las distintas actividades del sistema de información, mediante interfaces gráficas sencillas y amigables, para facilitar la interacción de los distintos usuarios con el sistema.

1.5 JUSTIFICACION

Este proyecto se realizara debido a la necesidad que tiene la asociación de productores en metalmecánica sur "metal sur", para facilitar el buen control y seguimiento de sus compras y

ventas de sus materiales de producción. De esta forma mejorar la atención a todos sus asociados y clientes.

La implementación del sistema de información se realizara con software libre, además la asociación cuenta con el hardware necesario para implantar este sistema, por lo que no incurrirá a un gasto adicional.

Con el desarrollo de este proyecto el personal de ventas de la asociación realizara sus tareas cotidianas con mayor comodidad y facilidad posibilitando una mayor eficiencia en ellos.

1.6 ALCANCES Y LIMITES

1.6.1 ALCANCES

- ✓ El sistema se desarrollara en un lenguaje visual que permitirá una interfaz amigable con los usuarios.
- ✓ Se restringirá el acceso a ciertas áreas donde solo el personal autorizado tendrá acceso.
- ✓ El sistema efectuara el registro de las compras que se realicen.
- ✓ El sistema efectuara el registro de las ventas de forma automatizada y precisa para el correcto control de ventas
- ✓ Se podrá obtener reportes actualizados en cualquier momento.
- ✓ El sistema podrá responder a consultas que se necesite acerca de las ventas realizadas en diferentes fechas.
- ✓ Se podrá realizar una emisión de reportes de stock de productos.

1.6.2 LIMITES

El sistema se limita a las siguientes excepciones.

- ✓ El sistema no generara los libros de compras, ventas y/o libros diarios de la asociación.
- ✓ El sistema no posee un sistema de facturación.
- ✓ El sistema no realizara el control del personal de la asociación.

CAPITULO II MARCO TEORICO

2.1 DEFINICION DE INVENTARIOS

Hamdy A. Taha (2004) señala que una empresa o una industria por lo común mantienen un inventario razonable de sus productos, para asegurar una operación ininterrumpida. Tradicionalmente, el inventario se considera como un mal necesario, demasiado poco causa costosas interrupciones y el exceso da por resultado un capital ocioso. El problema del inventario determina el nivel de inventario y equilibra estos dos casos extremos.

Un factor importante en la formulación y la solución de un modelo de inventarios es que la demanda (por tiempo de unidad) de un artículo puede ser;

- ✓ Determinîstico
- ✓ Probabilístico

2.1.1 MODELO DE INVENTARIO DETERMINISTICO

Los modelos deterministicos son aquellos donde se toma como supuesto que tenemos certeza de la demanda. Esta puede estar dada por pronósticos de demanda o pedidos reales de los clientes. Este modelo de inventarios se clasifica en dos. La primera donde la demanda puede ser estática, aquella donde la demanda permanece constante. Y la segunda dinámica donde a pesar de ser conocida la demanda, esta varia a través del tiempo.

- ➢ **MODELOS ESTATICOS DE CANTIDAD ECONOMICA DE PEDIDO (EOQ)**

Se considera que la demanda se conoce con certeza y que es siempre la misma, estos son:

- ✓ **Modelo clásico de cantidad económica de pedido:** Es el más sencillo de los modelos de inventario implica una tasa constante de demanda con el surtido instantáneo del pedido y sin faltante.

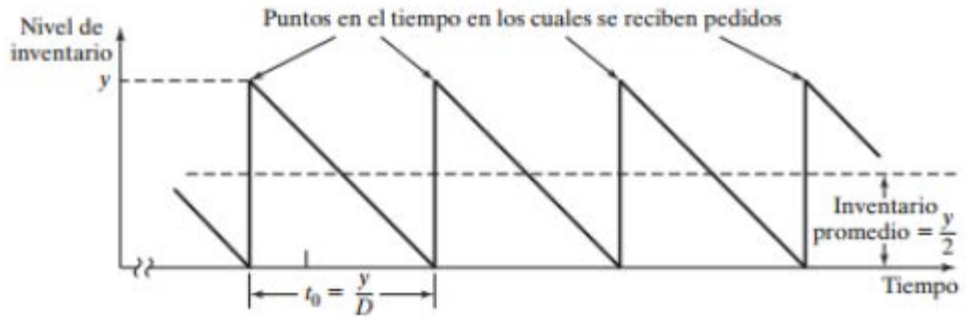


Figura: 2.1: comportamiento en modelo clásico
Fuente: (H. Taha, 2004)

y=cantidad pedida (cantidad en unidades)

D=tasa de demanda (unidades por unidad de tiempo)

To =duración del ciclo de pedido (unidades de tiempo).

- ✓ **EOQ con descuentos por cantidad:** se considera la disminución del costo de compra de un artículo cuando se compra en gran cantidad.
- ✓ **EOQ de artículos múltiples con límite de almacenamiento:** plantea que durante un tiempo la demanda no será satisfecha generando faltantes.

➢ **MODELO DE INVENTARIOS CON DEMANDA DINAMICA (EOQ)**

Estos modelos difieren de los modelos estáticos en dos aspectos: el primero donde el nivel de inventario se revisa periódicamente durante un numero finito de periodos iguales y la segunda debido a que la demanda por periodo, aun cuando es determinîstica, es dinámica, en el sentido de que puede variar de un periodo al siguiente, estos son:

- ✓ **Modelo sin costo de preparación:** Este modelo implica un horizonte de planificación con **n** periodos iguales. Cada periodo tiene una capacidad de producción limitada, que puede incluir varios niveles de producción (por ejemplo, el horario regular y las horas extra representa dos niveles de producción). Un periodo actual puede producir artículos para los periodos posteriores

- ✓ **Modelo con costo de preparación:** esta situación supone que no se permite ningún faltante y que se incurre en un costo de preparación siempre que se inicia un nuevo lote de producción.

2.1.2 MODELO DE INVENTARIO PROBALISTICO

En estos modelos de inventarios probabilísticos, la demanda se escribe con una distribución de probabilidades. Los modelos desarrollados se clasifican en general bajo situaciones de análisis continuo y periódico.

- ➢ **MODELOS DE REVISION CONTINUA**

Estos modelos presentan dos submodelos, una versión probabilizada del EOQ deterministica que utiliza exigencias estabilizadoras para explicar la demanda probabilística, y la otra un EOQ probabilístico más exacto que incluye la demanda probabilística de forma directa.

- ✓ Modelo EOQ probabilizado: trata de adaptar un modelo EOQ determinista para reflejar la naturaleza probabilista de la demanda usando una aproximación que sobre pone existencias estabilizadoras constantes sobre el nivel de inventario en todo el horizonte de la planeación. El tamaño de las existencias estabilizadoras se determina de modo que la probabilidad de agotamiento de existencias durante el tiempo de entrega no exceda un valor predeterminado.
- ✓ Modelo EOQ probabilístico**:** este modelo permite faltante de la demanda. La política requiere ordenar la cantidad y siempre que el inventario caiga a un cierto nivel.

- ➢ **MODELOS DE UN SOLO PERIODO**

Los modelos de inventarios de un solo artículo ocurren cuando una mercadería se ordena solo una vez para satisfacer la demanda del periodo. Por ejemplo los artículos de moda estacional se vuelven obsoletos al final de la estación. Esta sección presenta dos modelos.

- ✓ Modelo sin costo de preparación: la demanda ocurre de manera instantánea al inicio del periodo inmediatamente después de que se recibe el pedido y por otro lado no se contrae ningún costo de preparación.
- ✓ Modelo con costo de preparación: a diferencia del anterior este modelo incurre en un costo de preparación

2.1.3 CLASIFICACION DE INVENTARIOS

De acuerdo con (M. Muller, 2005) existen diferentes clasificaciones de inventarios, a continuación se cita algunas de ellas.

- ✓ **Inventario de Materias Primas:** Lo conforman todos los materiales con los que se elaboran los productos, pero que todavía no han recibido procesamiento.
- ✓ **Inventario de Productos en Proceso de Fabricación:** Lo integran todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales se encuentran en proceso de manufactura. Su cuantificación se hace por la cantidad de materiales, mano de obra y gastos de fabricación, aplicables a la fecha de cierre.
- ✓ **Inventario de Productos Terminados:** Son todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales son transformados para ser vendidos como productos elaborados.

La mayoría de los inventarios se ajusta a una de estas categorías, aunque el tamaño de cada una de ellas varia considerablemente, dependiendo de las características específicas del sector o el negocio a tratarse.

Los negocios de distribución suelen dedicarse principalmente a la reventa de producto terminado, mientras que una fábrica generalmente tiene menos productos terminados y más materias primas y productos en proceso.

2.1.4 TECNICAS DE VALUACION DE INVENTARIOS

Entre los métodos más importantes para valuar los inventarios, tenemos:

- ✓ **Método FIFO o PEPS:** Este método se basa en que lo primero que entra es lo primero en salir. Su apreciación se adapta más a la realidad del mercado, ya que emplea una valoración basada en costos más recientes.
- ✓ **Método LIFO o UEPS:** Contempla que toda aquella mercancía que entra de último es la que primero sale. Su ventaja se basa en que el inventario mantiene su valor estable cuando ocurre algún alza en los precios.
- ✓ **Método del Costo Promedio Aritmético:** El resultado lo dará la media aritmética de los precios unitarios de los artículos.

- ✓ **Método del Promedio Armónico o Ponderado:** Este promedio se calculará ponderando los precios con las unidades compradas, para luego dividir los importes totales entre el total de las unidades.
- ✓ **Método del Costo Promedio Móvil o del Saldo:** Calcula el valor de la mercancía, de acuerdo con las variaciones producidas por las entradas y salidas (compras o ventas) obteniéndose promedios sucesivos.
- ✓ **Método del Costo Básico:** Por medio de este método se atribuyen valores fijos a las existencias mínimas, este método es bastante parecido al LIFO con la diferencia de que se aplica solamente a la cantidad de inventario mínimo.
- ✓ **Método del Precio de Venta al Detalle:** Permite la estimación de inventarios con la frecuencia que se desee. El inventario físico se practicará, basándose en los precios de venta Marcados en los artículos.
- ✓ **Costo de Mercado o el Más Bajo:.** Se toma como base el precio inferior de las existencias, manteniendo el principio contabilístico del conservatismo el cual no anticipa beneficios y prevé posibles pérdidas.

2.2 COMPRA Y VENTA

Compra es la acción y efecto de comprar. Este verbo refiere a obtener algo a cambio de dinero. La compra implica necesariamente otra operación: la venta. Se trata de actividades opuestas: quien compra entrega dinero para recibir un bien o servicio, mientras que quien vende entrega el bien o servicio a cambio del dinero. Esto quiere decir que el comprador entrega dinero al vendedor y éste hace lo propio con el producto.

Por otra parte la palabra venta posee diferentes acepciones según el contexto en el que se aplique. Venta puede referirse a un objeto o servicio que se encuentra a disposición del público, lo cual quiere decir que aún no está vendido, por tanto, se encuentra en venta.

Venta es una acción que se genera de vender un bien o servicio a cambio de dinero. Las ventas pueden ser por vía personal, por correo, por teléfono, entre otros medios.

2.3 INGENIERIA DEL SOFTWARE

Existen varias definiciones para entender el concepto de ingeniería del software de acuerdo a cada autor.

(I. Sommerville, 2005), señala que la ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después que se utiliza.

Según Fritz Bauer, La ingeniería del software es el establecimiento y uso de principios fundamentales de la ingeniería con el objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales.

La IEEE (Institute of Electrical and Electronics Engineers) ha desarrollado una definición más completa y es como sigue: La ingeniería es del software es la aplicación de un enfoque sistemático, disciplinario y cuantificable al desarrollo, operación y mantenimiento del software.

2.4 METODOLOGIAS DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología, tienen la necesidad de venir documentadas, para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir.

2.5 METODOLOGIAS AGILES

Las metodologías agiles universalmente dependen de un enfoque iterativo para la especificación, desarrollo y entrega del software y principalmente fueron diseñados para apoyar al desarrollo de aplicaciones de negocio donde los requerimientos del sistema normalmente cambian rápidamente durante el proceso de desarrollo. Están pensados para entregar software funcional de forma rápida a los clientes (I. Sommerville, 2005).

Así mismo, estas metodologías son adoptadas para proyectos cortos y pequeños, la aparición de los riesgos es inevitable, se puede mitigar relativamente fácil dado que se da importancia a la simplicidad y a la eliminación del trabajo innecesario eliminando procesos complejos.

Desde esta perspectiva, el cliente hace parte integral del equipo de trabajo, lo que le brinda cierta flexibilidad en términos de la claridad que debe tener frente a sus necesidades, pues a medida que el proyecto avanza, y seguramente con la ayuda de los demás integrantes del equipo de trabajo, puede ir dando claridad a sus dudas.

De esta manera se explica que las metodologías agiles tienen cierta ventaja sobre las tradicionales, debido a que permite que el equipo de trabajo se comprometa con mayor grado de exigencia, tiempo organizado de acuerdo con las planificaciones, flexibilidad en el desarrollo para incorporar nuevas entregas.

Por otra parte, existe una amplia variedad de metodologías agiles, cada una con diferente naturaleza. Entre ellos se encuentran los siguientes:

- ✓ Programación extrema (XP)
- ✓ Scrum
- ✓ Modelo de desarrollo de sistemas dinámicos (MDSD)
- ✓ Cristal
- ✓ Desarrollo impulsado por las características (DIC)
- ✓ Proceso unificado ágil (PUA)
- ✓ Desarrollo esbelto de software (DES)

2.6 PROGRAMACION EXTREMA (XP)

Según R. Pressman (2010) la Programación Extrema, es el enfoque más utilizado de desarrollo de software ágil. Aunque las primeras actividades con las ideas y los métodos asociados a XP ocurrieron a finales de los años ochenta, el trabajo fundamental sobre la materia había sido escrito por Kent Beck. Una variante de XP llamada XP industrial, esta ultima mejora la XP y tiene como objetivo el proceso ágil para ser usado específicamente en organizaciones grandes.

2.6.1 VALORES XP

Kent Beck define un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas especificas de XP.

- ✓ **Comunicación:** XP pone énfasis en la en la colaboración estrecha pero informal (verbal) entre los clientes y los desarrolladores, en el establecimiento de metáforas para comunicar conceptos importantes, en la retroalimentación continua en evitar la documentación voluminosa como medio de comunicación.
- ✓ **Simplicidad:** XP restringe a los desarrolladores para que diseñen solo para las necesidades inmediatas, en lugar de considerar las del futuro. El objetivo es crear un diseño sencillo que se implemente con facilidad en forma de código. Si hay que mejorar el diseño, se rediseñara en un momento posterior.
- ✓ **Retroalimentación:** la retroalimentación se obtiene de tres fuentes; el software implementado, el cliente y otros miembros del equipo de software. Al diseñar e implementar una estrategia de pruebas eficaz, el software da retroalimentación al equipo ágil. XP usa la prueba unitaria como su táctica principal de pruebas. A medida que se desarrolla, el equipo implementa una prueba unitaria para ejecutar cada operación de acuerdo con su funcionalidad especificada. Cuando se entrega un incremento a un cliente, las historias del usuario o casos de uso que se implementa con el incremento se utilizan como base para las pruebas de aceptación. El grado en el que el software implementa la salida, función y comportamiento del caso de uso es una forma de retroalimentación. Por último, conforme se obtienen nuevos requerimientos como parte de la planeación iterativa, el equipo da al cliente una retroalimentación rápida con miras al costo y al efecto en la programación de actividades.
- ✓ **Valentía:** Beck afirma que la adhesión estricta a ciertas prácticas de XP requiere valentía. Un término más apropiado seria disciplina. Por ejemplo, es frecuente que haya mucha presión para diseñar requerimientos futuros. La mayor parte de equipos de software sucumben a ella y se justifican porque “diseñar para el mañana” ahorrara tiempo y esfuerzo en el largo plazo. Un equipo XP ágil debe tener la disciplina

(valentía) para diseñar para hoy y reconocer que los requerimientos futuros tal vez cambien mucho, por lo que demandaran repeticiones sustanciales del diseño y del código implementado.

- ✓ **Respeto:** Al apegarse a cada uno de estos valores el equipo ágil inculca respeto entre sus miembros, entre participantes y los integrantes del equipo, e indirectamente para el software en sí mismo. Conforme logra la entrega exitosa de incrementos de software, el equipo desarrolla más respeto para el proceso XP.

2.6.2 PROCESO XP

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales; planeación, diseño, codificación y pruebas. La figura 2.1 ilustra el proceso XP y resalta alguna de las ideas y tareas clave que se asocian con cada actividad estructural.

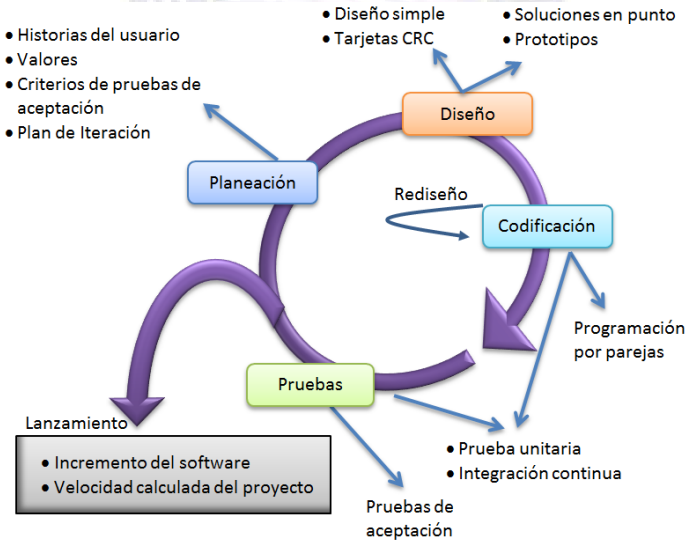


Figura 2.2: Proceso de desarrollo XP

Fuente: (R. Pressman, 2010)

- ✓ **PLANEACION:** La actividad de planeación comienza escuchando para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionales que se requieran. Escuchar lleva a la creación

de algunas “historias” (también llamadas historias de usuario) que describen la salida necesaria, características y funcionalidad del software que se va a elaborar. Cada historia es escrita por el cliente y colocada en una tarjeta indizada. El cliente asigna un valor es decir una prioridad a la historia con base en el valor general de la característica o función para el negocio. Después los miembros del equipo XP evalúan cada historia y le asignan un costo medido en semanas de desarrollo. Si se estima que la historia requiere más de tres semanas de desarrollo, se pide al cliente que la descomponga en historias más chicas y de nuevo se asigna un valor y costo. Es importante observar que en cualquier momento es posible escribir nuevas historias.

Una vez realizadas estas historias de usuario, se organiza una planificación con los diversos actores del proyecto (clientes, desarrolladores, gerentes), a efectos de establecer un plan o cronograma de entregas (release plan), en los que todos estén de acuerdo. Una vez acordado este cronograma comienza una fase de iteraciones, en donde en cada una de ellas se diseña, desarrolla y prueba.

(J. Joskowicz, 2008) establece conceptos básicos dentro de la fase de planificación y son los siguientes:

- **Historias de usuarios (User stories):** las historias de usuarios sustituyen a los documentos de especificación funcional, y a los casos de uso. Estas historias son escritas por el cliente, en su propio lenguaje, como descripciones cortar de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como pare los programadores puedan realizar una estimación poco riesgosa del tiempo que llevara su desarrollo. Las historias de usuario deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.
- **Plan de entregas (Release plan):** el cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto XP

- **Plan de iteraciones (Iteration plan):** las historias de usuario seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia se traduce en tareas específicas de programación. Así mismo para cada historia de usuario se establecen las pruebas de aceptación.
- **Reuniones diarias de seguimiento (Stand-up meeting):** el objetivo de tener reuniones diarias es mantener la comunicación entre el equipo y compartir problemas y soluciones.

✓ **DISEÑO**:

En XP se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio, el segundo motivo dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

- **Diseños Simples:** La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.
- **Glosarios de términos:** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.
- **Riesgos:** Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- **Funcionalidad extra:** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

- **Refactorizar:** Es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.
- **Tarjetas C.R.C. (Clase-Responsabilidad-Colaboración):** La principal funcionalidad que tiene estas tarjetas, es ayudar a dejar el pensamiento procedimental para incorporarse el enfoque orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. La siguiente tabla nos muestra un ejemplo de una tarjeta CRC.

Tabla: 2.1 Modelo de tarjeta CRC

NOMBRE DE CLASE	
DESCRIPCION	
Responsabilidad	**Colaboradores**

✓ **CODIFICACION**: La codificación es un proceso que se realiza en forma paralela al diseño. La codificación debe hacerse atendido a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente, facilita su comprensión y escalabilidad.

- **Reuniones con el cliente:** Como ya se mencionó anteriormente, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará

y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

- **Tests:** Se deben hacer pruebas unitarias al código comprobando que los resultados cumplan con las historias de usuario.
- **Desarrollo colectivo:** XP sugiere que todos los programadores estén implicados en todas las tareas, de manera que cualquiera pueda modificar o mejorar el código ajeno.
- **Estándares de codificación:** En esta fase deben seguirse estándares de codificación de manera que el código se mantenga ordenado, consistente y de manera que sea de fácil compresión y escalabilidad.
- **Programación en pares:** XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que esta práctica duplica el tiempo el tiempo asignado al proyecto, al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

✓ **PRUEBAS**: XP enfatiza muchos aspectos relacionados con las pruebas, clasificándolas en diferentes tipos, funcionalidades específicas, indicando quien, cuando y como deben ser implementadas y ejecutadas.

Del buen uso de las pruebas depende el éxito de otras prácticas, tales como la propiedad colectiva del código y la refactorización. Cuando se tienen bien implementados las pruebas no habrá temor de modificar el código del otro programador en el sentido que si se daña alguna sección, las pruebas mostraran el error y permitirán encontrarlo.

Según XP se debe ser muy estricto con las pruebas. Sólo se deberá liberar una nueva versión si esta ha pasado con el cien por ciento de la totalidad de las pruebas. En caso contrario se empleara el resultado de estas para identificar el error y solucionarlo (L. Echeverry y L. Delgado, 2007).

- **Pruebas Unitarias:** Estas pruebas se aplican a todos los métodos no triviales de todas las clases de proyecto con la condición que no se liberara ninguna clase que tenga asociada su correspondiente paquete de pruebas. Uno de los elementos mas importantes en estas es que idealmente deben ser construidas antes que los métodos mismo, permitiendo al programador tener máxima claridad sobre lo que va ha programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberá pasar.
- **Pruebas de Aceptación**: También llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. Las pruebas de aceptación son pruebas de caja negra, que representa un resultado esperado de determinada transacción con el sistema.

2.6.3 EQUIPO DE TRABAJO XP

A continuación describimos el equipo de trabajo de acuerdo a la propuesta original de Kent Beck, los cuales son:

- ✓ **Programador**: El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- ✓ **Cliente**: El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuales se implementa en cada iteración centrándose en aportar mayor valor al negocio. El cliente es solo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- ✓ **Encargado de pruebas (Tester)**: El encargado de pruebas ayuda al cliente a escribir las pruebas, difunde los resultados en el equipo y es el responsable de las herramientas de soporte para pruebas.
- ✓ **Encargado de seguimiento (Tracker):** Proporciona realimentación al equipo en el proceso XP, su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del proceso de cada iteración y evalúa si

los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuando es necesario realizar algún cambio para lograr los objetivos de cada iteración.

- ✓ **Entrenador (Coach):** Es el responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las practicas XP y se siga el proceso correctamente.
- ✓ **Consultor**: Regularmente el consultor no formaba parte del equipo, bueno de hecho no lo integra. El consultor es un miembro externo del equipo, pero que cuenta con conocimientos específicos y que será capaz de ayudar en la solución de problemas.
- ✓ **Gestor (Big boss):** Posiblemente el líder más alto. Si se trata de unir a los clientes con los programadores, el gestor es el intermedio, es decir. Es el encargado de vincular e interrelacionar al cliente con los programadores creando las condiciones adecuadas, su labor principal es la coordinación.

2.6 UML (Lenguaje Unificado de Modelado)

El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del7negocio y otros sistemas no software [OMG01].

UML es la herramienta más utilizada en la actualidad para el desarrollo de sistemas. Esto se debe a que permite a los creadores generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas.

UML está compuesto por diversos elementos gráficos que se combinan para formar diagramas. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos. En lugar de indicarle cuales son los elementos y las reglas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se las conoce como modelo. Un modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema (J. Schmuller, 2000).

A continuación se describe los diagramas más comunes del UML y los conceptos que representan.

- ✓ **Diagrama de clases:** una clase es una categoría o grupo de cosas, está representada por un rectángulo, este tiene un nombre, atributos y acciones similares, en la figura 2.2 muestra un ejemplo de la notación del UML que captura los atributos y acciones de una lavadora.

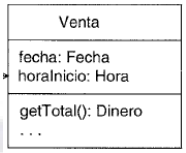


Figura 2.3: representación de una clase
Fuente: Fuente: (G. Larman, 2004)

Por lo tanto, un diagrama de clases está formado por varias clases (rectángulos) conectados por líneas que muestran la manera en la que las clases se relacionan entre si.

Los diagramas de clases facilitan las representaciones a partir de las cuales los desarrolladores podrán trabajar. A su vez los diagramas de clases colaboran en lo referente al análisis. Permiten al analista hablarle a los clientes en su propia terminología, lo cual hace posible que los clientes indiquen importantes detalles de los problemas que requieren ser resueltos.

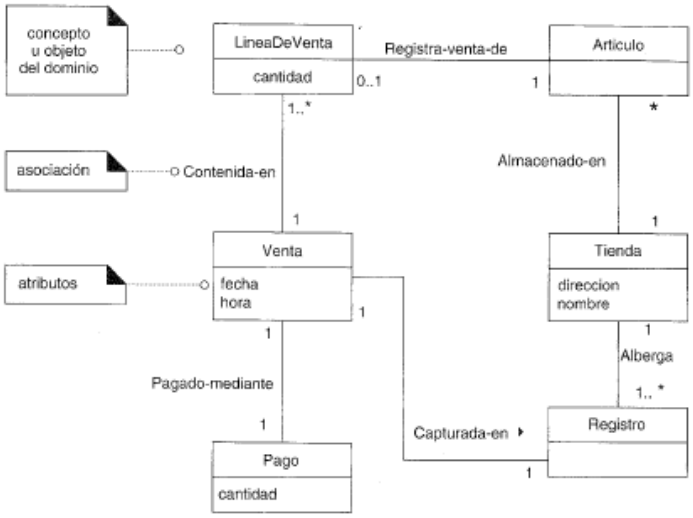


Figura 2.4: diagrama de clases
Fuente: (G. Larman, 2004)

Casos de uso: un caso de uso es una estructura que ayuda a los analistas a trabajar con los usuarios para determinar la forma en que se usara un sistema. Con una colección de casos de uso se puede hacer un bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él. Estos casos de uso son una excelente herramienta para estimular a que los usuarios hablen, de un sistema, desde sus propios puntos de vista. Debido a que no siempre es fácil para los usuarios explicar como pretenden utilizar un sistema.

- ✓ **Diagrama de casos de uso**: Los diagramas de casos de uso describen las acciones de un sistema desde el punto de vista del usuario. UML proporciona notación para los diagramas casos de uso con el fin de ilustrar los nombres de los caso de uso y los actores y las relaciones entre ellas.

Figura 2.5: Diagrama de casos de uso
Fuente: (J. Schmuller, 2000)

Un diagrama de caso de uso muestra los confines entre el sistema y el mundo exterior. Generalmente los actores están fuera del sistema, mientras que los casos de uso están dentro de e. los componentes de estos diagramas son: los actores, los casos de uso y las líneas de interconexión.

- ✓ **Diagrama de estados:** Presenta los estados en los que puede encontrarse un objeto junto con las transiciones entre los estados y muestran los puntos inicial y final de una secuencia de cambios de estado.

Los símbolos UML en un diagrama de estados son: el icono para el estado es un rectángulo de vértices redondeadas, y el símbolo de una transición es una línea continua y una punta de fecha, un circulo relleno se interpreta como el punto inicial de una secuencia de estados y una diana representa el punto final.

- ✓ **Diagrama de secuencias**: un diagrama de secuencia es un artefacto creado de manera rápida y fácil, que muestra los eventos de entrada y salida relacionados con el sistema que se está estudiando. UML incluye la notación de los diagramas para representar los eventos que parten de los actores externos hacia el sistema.

  Por otra parte, un diagrama de secuencia es un dibujo que muestra para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas.

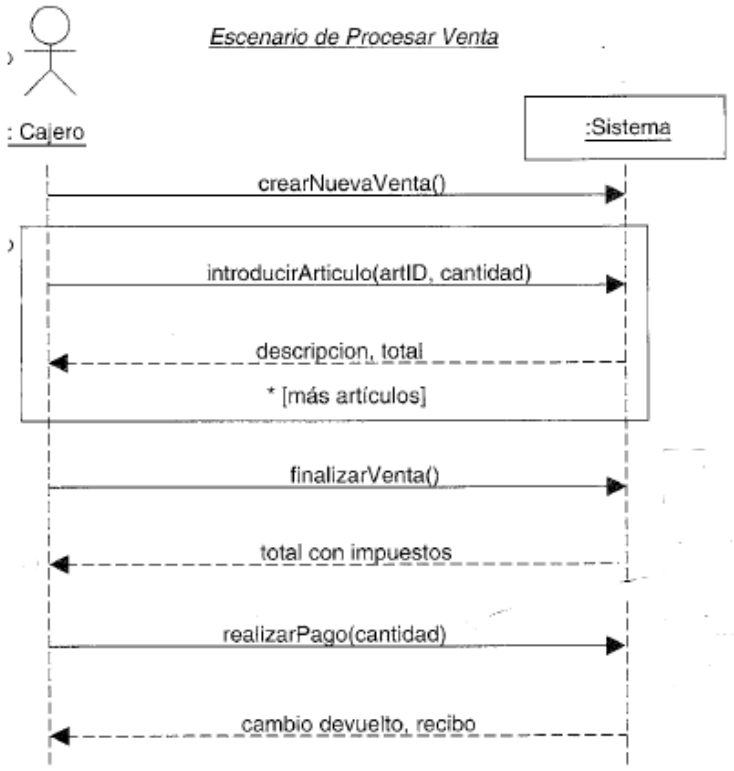


Figura 2.6: Diagrama de secuencia
Fuente: (G. Larman, 2004)

- ✓ **Diagrama de actividades:** Muestra los flujos de trabajo de negocios u operativos representados gráficamente para mostrar la actividad de alguna parte o componente del sistema. Los diagramas de actividades se usan como una alternativa a los diagramas de estado. Cada actividad se lo representa por un rectángulo con las esquinas redondeadas (mas angosta y ovalado que la representación del diagrama de estado). El procesamiento dentro de una actividad se lleva a cabo y, al realizarse, se continúa con

la siguiente actividad. Al igual que los diagramas de estado, el de actividad cuenta con un punto de inicio y un final.

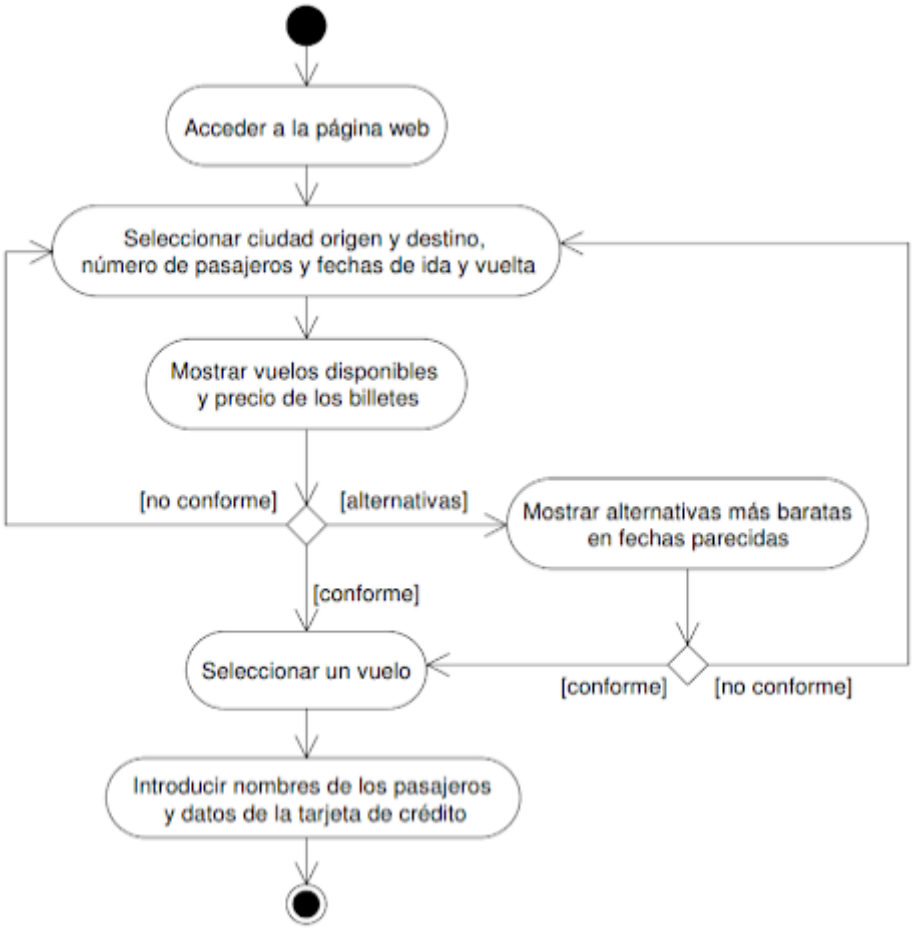


Figura 2.7: Diagrama de actividades
Fuente: (G. Larman, 2004)

- ✓ **Diagrama de distribución:** El diagrama de distribución UML, muestra la arquitectura física de un sistema informático. Puede representar los equipos y dispositivos, mostrar sus interconexiones y el software que se encontrar en cada máquina. Cada computadora esta representada por un cubo y las interconexiones están representadas por líneas que conectan los cubos.

2.8 DEFINICION DE APLICACION WEB

Las aplicaciones Web (WebApps) es una categoría de software centrado en redes, agrupa una amplia gama de aplicaciones. Es un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y graficas limitadas. Sin embargo desde que existió Web 2.0 las WebApps están evolucionando hacia ambientes de computo sofisticado que no solo proveen características aisladas, funciones de computo y contenido para el usuario final, sino que también están integradas con base de datos corporativas y aplicaciones de negocio (R. Pressman, 2010).

CARACTERISTICAS DE UNA APLICACIÓN WEB

- ✓ Pueden ser ejecutadas en múltiples plataformas ya que no hacen uso del sistema operat1ivo del terminal o equipo, sino del navegador implementado para su ejecución.
- ✓ No se instalan en el dispositivo y consiguen una experiencia de operación muy similar a una aplicación nativa, pero requieren conexión constante a Internet.
- ✓ La inversión de recursos para su desarrollo se realizará una sola vez para múltiples plataformas, pero se deberá optimizar en cada una de ellas para obtener el mejor rendimiento de cada ambiente.
- ✓ El diseño de interfaz está condicionado por necesidades de claridad y simplicidad sobre la de impacto.
- ✓ Su funcionalidad puede variar desde navegación/consulta de ABMs hasta complejos servicios transaccionales de comercio electrónico.
- ✓ Al igual que en la Ingeniería de Software, en la Ingeniería Web también se realizan las denominadas Actividades CGC (de Control y Garantía de la Calidad) como el establecimiento y supervisión de estándares, revisiones técnicas formales, análisis y seguimiento/registro de informes, entre otros; sin embargo otros aspectos valorativos que le son exclusivos están relacionados a criterios de usabilidad, funcionabilidad, fiabilidad, seguridad, eficiencia y mantenibilidad, en el marco de la escalabilidad (Olsina,1999).

2.9 CALIDAD DEL SOFTWARE

Calidad del software se define como un proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (R. Pressman, 2010)

Los productos de software tienen un cierto número de atributos asociados que reflejan la calidad de ese software. Estos atributos no están directamente asociados con lo que el software hace. Mas bien, reflejan su comportamiento durante su ejecución, en la estructura y en la organización del programa fuente y en la documentación asociada. Estos atributos son: mantenibilidad, confiabilidad, eficiencia y usabilidad (I. Semmerville, 2005)

David Garvin (Gar87) sugiere que la calidad debe tomarse en cuenta, adoptando un punto de vista multidimensional que comience con la evaluación de la conformidad y termine con una visión trascendental (estética). Garvin toma en cuenta ocho atributos que se deben tomar en cuenta, los cuales son: calidad del desempeño, calidad de las características, confiabilidad, conformidad, durabilidad, servicio, estética y percepción.

El estándar ISO 9126 se desarrollo con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos, estos son:

- ✓ **Funcionalidad**. Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- ✓ **Confiabilidad**. Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: Madurez, tolerancia a fallas y recuperación.
- ✓ **Usabilidad**. Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.
- ✓ **Eficiencia**. Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los siguientes sub atributos: comportamiento del tiempo y de los recursos.

- ✓ **Facilidad de recibir mantenimiento**. Facilidad con la que pueden efectuarse reparaciones del software, según lo indican los siguientes subatributos: analizable, confiable, estable y susceptible de someterse a pruebas.
- ✓ **Portabilidad**. Facilidad con la que el software puede llevarse de un ambiente a otro, según lo indican los atributos siguientes: adaptable, instalable, conformidad y sustituible.

Estos factores de calidad ISO 9126 del software estudiados anteriormente no necesariamente conducen a una medición directa. Sin embargo proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema.

2.10 SEGURIDAD DEL SOFTWARE

La seguridad lógica, consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo se permita el acceso a personas autorizadas.

Según la UNE-ISO /IEC **17779**, cuyos principales puntos para la seguridad de un Sistema son:

- ✓ **Confidencialidad:** En términos de seguridad de la información, la confidencialidad hace referencia a la necesidad de ocultar o mantener secreto sobre determinada información o recurso. El objetivo de la confidencialidad es, entonces, prevenir la divulgación no autorizada de la información. El mecanismo que garantice la confidencialidad es la Criptografía, para garantizar la confidencialidad en el sistema se utilizó la encriptación Asimétrica Existen dos claves, una para cifrar el mensaje y otra para descifrarlo, generalmente la primera es pública, es decir, solo la conoce el emisor, en cambio la segunda se denomina Privada y solo la posee a quien van dirigidos los mensajes enviados entre los que disponen de la clave pública, por tanto, solo el poseedor de la clave privada podrá leer los mensajes.

  La criptografía simétrica es más vulnerable que la asimétrica por el hecho de usar una única llave, por otro lado en la encriptación simétrica es más rápida que la asimétrica y esto la favorece ya que el tiempo de descifrado es bastante importante. Por tanto las

contraseñas de cada usuario del sistema serán controlada a través de una librería creada para JavaScript donde maneja los roles de usuario.

- ✓ **Integridad:** En términos de seguridad de la información, la integridad hace referencia a la fidelidad de la información o recurso, y normalmente se expresa en lo referente a prevenir el cambio impropio o desautorizado. La integridad hace referencia a:
    - La integridad de los datos (el volumen de la información)
    - La integridad del origen (la fuente de los datos, llamada autenticación)

  Para mantener la integridad del Sistema se han definido perfiles de usuario para el acceso con roles específicos de acuerdo al nivel del usuario, garantizando así la integridad en los datos, para que el personal no autorizado no tenga acceso a modificar, eliminar datos sin autorización.

- ✓ **Disponibilidad:** En términos de seguridad de la información, la disponibilidad hace referencia a que la información del sistema debe permanecer accesible a elementos autorizados. Para asegurar la disponibilidad del Sistema, se implementó el uso de variables de sesión permitiendo el acceso a la información autorizada sin restricciones.
- ✓ **Autenticidad:** Este atributo nos permite asegurar el origen de la información, es decir validar la identidad del usuario para esto el sistema debe verificar su nombre de usuario y la contraseña, para la seguridad en la autenticación se inscripto la contraseña, una vez validado el usuario y la contraseña el sistema permite el acceso a la página principal del Sistema.
- ✓ **Auditabilidad:** Este atributo nos permite determinar qué acciones se han ejecutado en el Sistema Web, quien lo ha ejecutado, que fecha se ejecutó, y cuáles son los registros afectados. Para esto se lleva un registro de todas las acciones ejecutadas (O. Conde, 2017).

2.11 HERRAMIENTAS DE DESARROLLO

Para la implementación de este proyecto se utilizará las siguientes herramientas tecnológicas: Plataforma Windows, Lenguaje de Programación PHP, Sistema Gestor de Base de Datos MySQL además de otras herramientas necesarias para el diseño y implementación del sistema web como se muestra en la tabla

Tabla 2.2: Herramientas de desarrollo

Herramientas de desarrollo		
Producto	Uso	Descripción
Apache Server	Servidor de aplicaciones Web	Licencia: GLP
PHP	Lenguaje de programación	Licencia: GLP
MySQL	Software de Base de Datos	Licencia: GLP
Navegaror Firefox	Navegador Web	
Windows 10	desarrollo	Licencia: comercial arquitectura de 32 bits

CAPITULO III MARCO APLICATIVO

3.1 ANALISIS DE SITUACION ACTUAL

La Asociación Metal Sur cuenta con un almacén de ferretería, el cual se encarga de suministrar los materiales para el diseño y producción de todo tipo de estructuras en hierro y acero inoxidable a todos sus socios y a toda la comunidad en general.

La ferretería de la asociación Metal sur realiza los procedimientos de compra, venta y administración de inventario de forma manual, cuando el cliente pide un determinado producto la persona que está atendiendo realiza la búsqueda del producto en la ferretería. Si el producto está disponible se procede a registrar la venta en un cuaderno los productos vendidos y costo total. En la Figura 3.1 se muestra el proceso de venta de la ferretería.

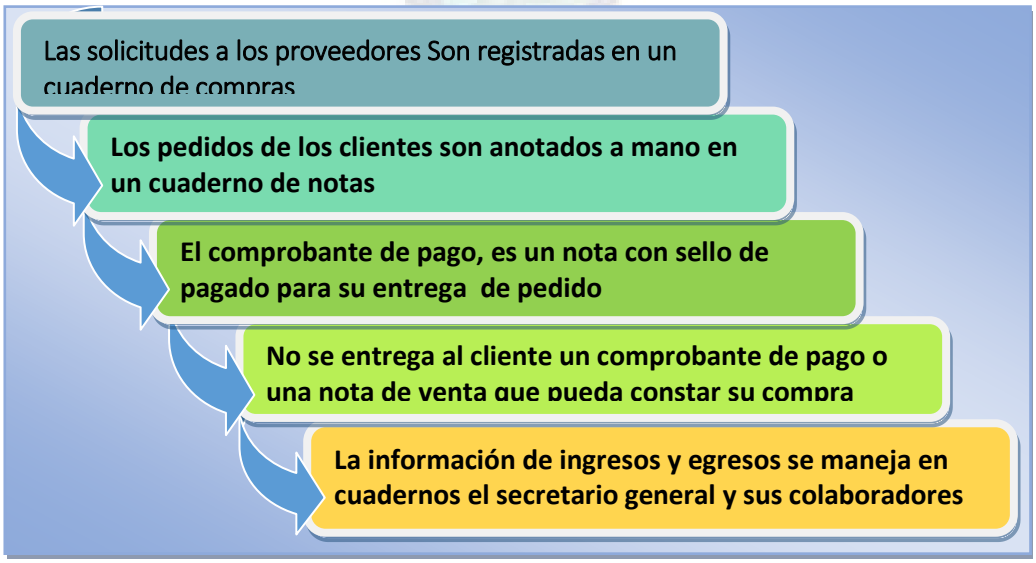


Figura: 3.1 Proceso actual de comercialización “METAL SUR”

En el caso del inventario se realizan los siguientes procedimientos:

- ✓ Los nuevos productos se registran en un cuaderno de inventario, allí se escriben los datos más importantes de cada producto, como el costo, el precio de venta, la cantidad en existencia, y las características más relevantes de ese producto.
- ✓ Cuando se desea hacer revisión de los productos vendidos (salidas del inventario) se tiene que contar físicamente los productos o revisar el cuaderno donde está el registro de los productos vendidos.
- ✓ Para conocer si hay productos agotados se debe contar uno a uno los productos.
- ✓ El inventario se revisa de forma periódica.
- ✓ Cuando hay nuevas existencias de un producto determinado se vuelve a hacer otro registro en el cuaderno del mismo producto, se vuelven a llenar los datos y se revisa el producto anterior para sumar las existencias

Este método, no ofrecía las funcionalidades que necesitaba esta asociación. Motivo por el cual se acordó desarrollar un software de ventas y con elementos de administración de inventario que cumpliera las necesidades especificas de la asociación.

3.2 FASE DE PLANIFICACION

En esta primera fase de ciclo de vida de la metodología XP, se determino los requerimientos funcionales del proyecto propuesto, requerimientos que surgió luego de realizar una serie de entrevistas con el personal que se encuentra directamente involucrado en los procedimientos de venta, almacén y el reabastecimiento de la ferretería.

Así mismo, se contó con la participación activa de dos miembros de la asociación Metal Sur. El secretario general en calidad de administrador y un personal de venta los cuales tenían pleno conocimiento del funcionamiento del almacén de ferretería, lo cual facilito la comunicación para el desarrollo del proyecto.

Por otra parte, se procedió a organizar y clasificar los productos que se tiene en existencia, esto permitirá tener una visión más clara de la situación de los productos en inventarios. Se procede a actualizar el catalogo de productos eliminando los productos que ya no se maneja y

los productos que se encuentran repetidos, finalmente los productos se clasificaron en familias y/o grupos de productos relacionados con cada uno de los proveedores.

3.2.1 ESPECIFICACION DE REQUERIMIENTOS

Esta fase del proceso de desarrollo, hace referencia a las necesidades que tiene la ferretería de la asociación especificando los requerimientos funcionales, procedimientos de trabajo, plataforma y la tecnología con la que se implementara el sistema.

Por otra parte, En la investigación realizada se considera que existen diversos enfoques de elaboración de los inventarios, este proyecto particularmente se basa en el primer modelo presentado llamado Modelo clásico de cantidad económica de pedido (EOQ) sin faltante, el cual representa mayor rentabilidad y por lo mismo se considero que la aplicación de dicho modelo significa la generación de una transformación a la organización de las compras de la asociación.

3.2.2 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales son aquellos que definen el comportamiento interno del sistema, detalles técnicos, manipulación de datos y todas las funcionalidades específicas.

Para establecer estos requerimientos se procedió a realizar varias reuniones con los usuarios del sistema y una serie de visitas a la ferretería de la asociación, en el cual se pudo evidenciar los siguientes requerimientos funcionales para el desarrollo del proyecto.

- ✓ el sistema permitirá el acceso solo a usuarios autorizados, luego de validar los datos del usuario.
- ✓ el sistema por medio de menú de opciones, permitirá la creación, modificación y/o eliminación de los usuarios.
- ✓ los usuarios podrán acceder a ciertas áreas del sistema dependiendo el nivel de usuario.
- ✓ El sistema debe permitir el registro de nuevos proveedores que abastecen el almacén de ferretería como también modificar y eliminar.
- ✓ El sistema también permitirá el registro de nuevos artículos o materiales, asignando a cada uno de ellos un código de artículo. Así como también se pueda realizar las acciones de crear nuevo, modificar o eliminar un determinado artículo o material.

- ✓ El proceso de compra debe registrar la siguiente información: nombre de material, cantidad, precio de compra, precio de venta final, código de material, nombre de proveedor entre otros.
- ✓ Las compras y las ventas deben modificar de manera directa al inventario, disminuyendo o incrementando el stock de cada producto de forma automática en cada transacción.
- ✓ El sistema debe permitir realizar el reporte de existencias de materiales de la ferretería de la asociación para fines de información y para ser utilizados en las órdenes de compra.
- ✓ El registro de venta debe realizarse mediante una interfaz grafica donde se pueda realizar la búsqueda de un determinado material, descripción, precio de venta, cantidad, nombre de cliente, cedula de identidad entre otros. Por otra parte el sistema debe poder realizar la venta de varios materiales en un solo pedido de venta.
- ✓ El sistema debe permitir relacionar las ventas con los clientes registrados, es decir que para cada venta el sistema debe capturar los datos del cliente mediante una búsqueda por cedula de identidad. En cado de no existir el cliente se debe registrar como nuevo.
- ✓ El sistema debe relacionar las ventas y compras directamente con el inventario, de tal modo que disminuya la cantidad de un material de acuerdo a la cantidad vendida o incrementar la cantidad de un mismo producto por una compra realizada.
- ✓ El sistema también debe permitir hacer correcciones a la venta o por defecto anular la venta.
- ✓ El sistema debe realizar la impresión de un comprobante de pago.
- ✓ El sistema debe cuantificar la venta total diaria, semanal y/o de acuerdo a un intervalo de fecha de cada vendedor.
- ✓ El sistema realiza el reporte de inventarios donde se muestra la cantidad de existencia de cada material, así mismo resaltara los materiales que se encuentran con una cantidad baja.
- ✓ Cada uno de estos reportes debe contener una opción de impresión.
- ✓ El sistema debe facilitar el registro de los clientes por medio de una interfaz grafica.
- ✓ El sistema debe brindar los datos de los clientes al momento que se realiza una venta.

3.2.3 HISTORIAS DE USUARIO

A partir del conjunto de requerimientos mencionados anteriormente, se pudo construir las historias de usuario que se muestra en la tabla 3.1, estas historias de usuario serán la base para el desarrollo de este proyecto.

Tabla: 3.1 Historias de Usuario propuestos

Nº	Nombre de historia de usuario
H1	Control de acceso a usuarios
H2	Creación de roles personalizados
H3	Gestión de clientes
H4	Gestión de proveedores
H5	Gestión productos y artículos
H6	Gestión de pedidos de ventas

Cada una de estas historias de usuario que mencionamos los desglosamos en los siguientes tablas, donde se realiza una breve descripción del mismo, el tiempo de tomara en el desarrollo y los usuario que se ven involucrados.

Tabla: 3.2 Historia de usuario Control de acceso a usuario

HISTORIA DE USUARIO	
Número: 1	**Usuario:** administrador del sistema
Nombre historia: Control de acceso a usuarios	
Días de desarrollo estimados: 3	**Iteración asignada: 3**
Descripción: Para iniciar sesión en la aplicación primero se solicita el nombre de	

usuario y una clave para que tenga acceso a los datos que corresponda a su categoría de usuario. Se establecen dos tipos de usuario: administrador y vendedor, ambos con diferentes permisos de acceso a los menús y funcionalidades que corresponda.
Observaciones: Confirmado por el cliente

Tabla: 3.3 Historia de usuario Creación de roles personalizados

HISTORIA DE USUARIO		
Número: 2	**Usuario:** administrador del sistema	
Nombre historia: Creación de roles personalizados		
Días de desarrollo estimados: 4		**Iteración asignada: 3**
Descripción: El Sistema debe tener la capacidad de asignar roles a un usuario. Estos roles se deben personalizar de tal manera que pueda efectuar tareas puntuales y a la vez diferentes a otro rol. Para ello cada rol puede involucrar el acceso de algunos de los módulos presentes en la aplicación y denegar el acceso a otro módulo según sea requerido en el momento.		
Observaciones: Confirmado por el cliente		

Gestión de clientes: El sistema debe facilitar el registro de todos los clientes que realizan sus pedidos en la ferretería con el objetivo de agilizar las ventas posteriores de modo que solo tengan que facilitar el numero de cedula de identidad y el sistema deberá identificar al cliente.

Tabla: 3.4 Historia de usuario Gestión de clientes

HISTORIA DE USUARIO		
Número: 3	**Usuario:** Administrador del sistema, usuario vendedor	
Nombre historia: Gestión de clientes		
Días de desarrollo estimado: 2.5		**Iteración asignada: 1**
Descripción: El sistema debe tener una interfaz de usuario para el registro de los clientes que solicitan un pedido. Además debe contar con las opciones de adicionar nuevo, modificar o eliminar.		
Observaciones: ninguna		

Gestión de proveedores: la gestión de proveedores nos permite recabar y almacenar la información de todos los proveedores que suministran productos al almacén de ferretería de la asociación Metal Sur. Esta información nos permitirá llevar el control de los productos y su respectivo proveedor.

Tabla: 3.5 Historia de usuario Gestión de proveedores

HISTORIA DE USUARIO		
Número: 4	**Usuario: administrador del sistema**	
Nombre historia: Gestión de proveedores		
Días de desarrollo estimados: 4		**Iteración asignada: 1**
Descripción: El sistema propuesto podrá gestionar la información de los proveedores que suministran los diferentes productos y artículos al almacén de ferretería. La		

información obtenida se podrá almacenar, editar o eliminar. Este proceso debe realizarse mediante una interfaz sencilla y fácil de utilizar.
Observaciones: Solo se debe recabar los datos de los proveedores ya que los artículos se lo realiza en otra historia de usuario.

Gestión de productos/artículos: la gestión de productos/ artículos, hace referencia a los materiales para la producción que necesita la ferretería de la asociación. La siguiente historia de usuarios describe la gestión de productos **y** la clasificación de los productos en categorías.

Tabla: 3.6 Historia de usuario Gestión de productos y artículos

HISTORIA DE USUARIO	
Número: 5	**Usuario:** Administrador del sistema
Nombre historia: Gestión de productos/artículos	
Días de desarrollo estimados: 6	**Iteración asignada: 2**
Descripción: Se una desarrollara una interfaz sencilla y fácil de utilizar. En la que nos muestre un listado de productos. Los primeros en la lista serán los que tengan un número menor de diez unidades, así mismo debe facilitar la búsqueda de algún producto mediante la selección de diferentes características además podrá permitir la adición de un nuevo producto, modificación o eliminación de acuerdo a las necesidades del usuario.	
Observaciones: ninguna.	

Gestión de pedidos de ventas: La siguiente historia de usuario describe la gestión de pedido de ventas dentro del almacén de ferretería.

Se podrá realizar un reporte o un listado de todos los artículos en existencia en el almacén y con todas las cantidades existentes por producto, precio y proveedor. El sistema debe registrar las ventas que se realizan y los artículos con los que cuenta.

Tabla: 3.7 Historia de usuario Gestión de pedido de venta

HISTORIA DE USUARIO	
Número: 6	**Usuario:** Administrador del sistema, usuario vendedor
Nombre historia: Gestión de pedidos de ventas	
Días de desarrollo estimados: 3	**Iteración asignada: 3**
Descripción: El administrador y el vendedor podrán acceder a la base de datos del sistema para consultar la existencia y precio de un producto. Así mismo sistema podrá realizar el registro de las ventas y en una venta se podrá registrar varios productos mediante un interfaz grafica. Por otra parte el sistema imprimirá un comprobante de pago para el cliente y uno para el encargado de despacho de los pedidos.	
Observaciones: si el producto y/o articulo se encuentra con una cantidad igual a cero el sistema debe alertar con un aviso	

3.2.4 PLANIFICACION DE ITERACIONES

Una vez establecidas las historias de usuario, se realiza la planificación de las iteraciones para el desarrollo del proyecto. Se establecieron un número de tres iteraciones las que se agruparon de la siguiente forma.

ITERACION Nº 1

Historia de usuario Nº 1: **Gestión de usuarios**
Historia de usuario Nº 2: **Asignación de roles**

ITERACION Nº 2

Historia de usuario Nº 3: **Gestión de clientes**
Historia de usuario Nº 4: **Gestión de proveedores**

ITERACION Nº 3

Historia de usuario Nº 5: **Gestión de productos y artículos**
Historia de usuario Nº 6: **Gestión de ventas**

Tabla: 3.8 Plan de iteraciones

	Iteración 1	**Iteración 2**	**Iteración 3**
Historias de usuario	H1, H2	H3, H4	H5, H6
Fecha inicio	26/02/2018	26/03/2018	27/04/2018
Fecha fin	23/03/2018	24/04/2018	07/06/2018
Tiempo estimado en semanas calendario	4	4	7
Tiempo de desarrollo en días	24	22	36
Horas de desarrollo (se toma en cuenta un total de 8hs/día**)**	**192**	**176**	**280**

3.3 FASE DE DISEÑO

Para esta fase de diseño nos apoyamos en diagramas UML que nos ayudara a comprender mejor el contexto del proyecto a desarrollar, vea ANEXO.

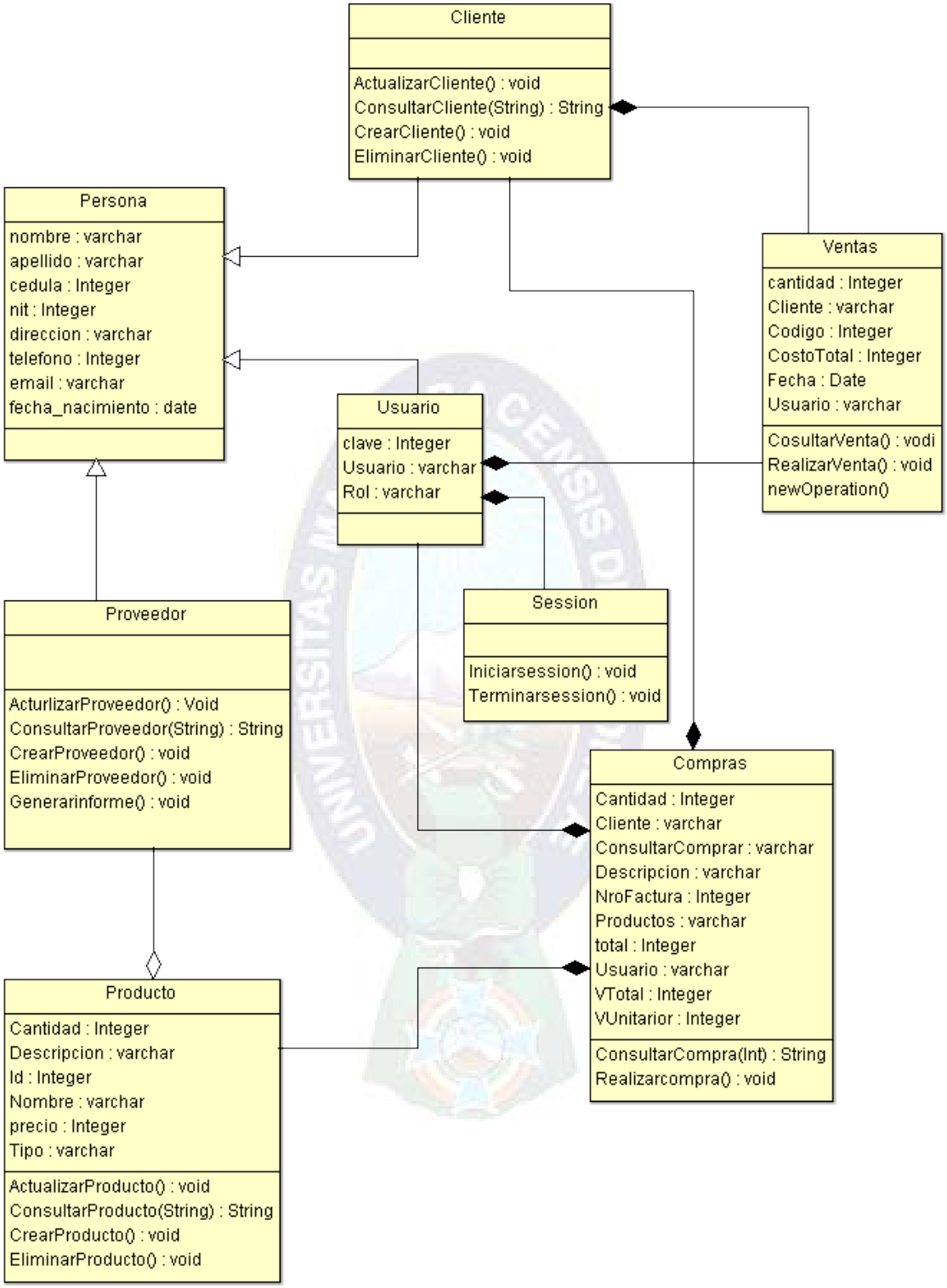


Figura 3.2 Diagrama de clases

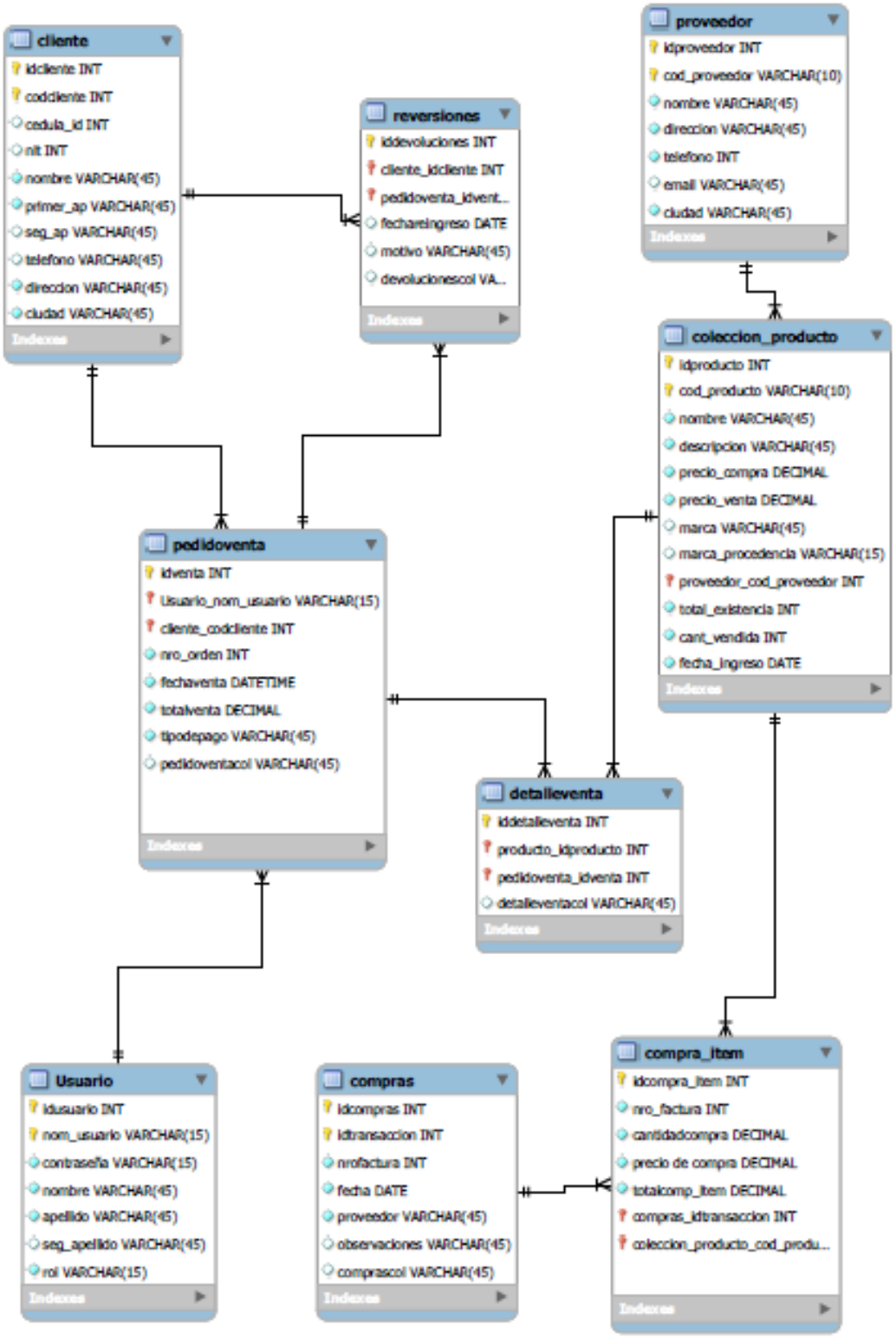


Figura: 3.3 Modelo físico de la base de datos

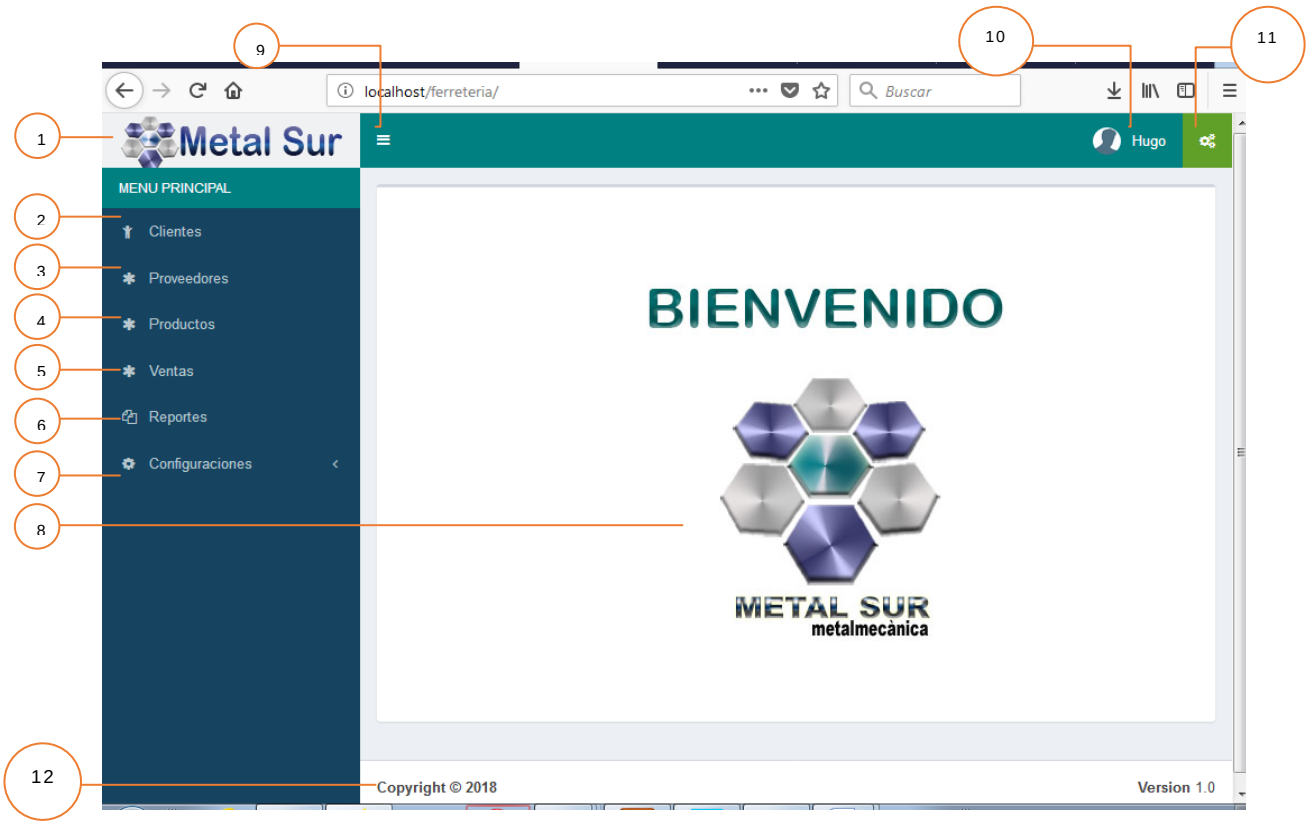


Figura: 3.4 Diseño de la estructura y composición del sistema

En la figura anterior se observa el diseño de la estructura y composición del sistema web de registro de compras, ventas y administración de inventario, en el cual se muestra la forma en la que está organizada los elementos que la componen. A continuación describimos cada uno de estos elementos.

1. Es el logotipo de la asociación, esta imagen acompaña la navegación por todo el sitio.
2. La opción clientes es parte del menú principal, este menú muestra los datos de los clientes ya registrados en el sistema y las opciones de adición de nuevos clientes así como modificar o eliminar.
3. La opción proveedores permite ver los proveedores que tenemos registrados en el sistema y se puede realizar el registro de nuevos proveedores, modificar o eliminar.
4. La opción productos se clasifica en familia de productos del mismo dodo nos permite el registro de un nuevo producto, modificar y eliminar. Por otra parte esta opción de

menú permitirá asignar un código para cada productos y otros datos que son importantes para el registro.

5. La opción de ventas permite registrar las ventas que realiza la asociación. Esta operación se realiza mediante una búsqueda por código de producto o su descripción que se adicionan al conjunto de ítems por comprar.
6. La opción de reportes nos permite realizar reporte de ventas o ingreso de productos comprados. Estos reportes podrán ser diarias, semanales, mensuales o de acuerdo a un intervalo de fecha a seleccionar.
7. La opción configuración, permite realizar el registro de nuevos usuarios al sistema o su respectiva modificación o eliminación. Esta opción estará permitida solo para el administrador del sistema
8. En esta sección se muestra el contenido de cada opción del menú principal del sistema, una vez ingresando al sistema se muestra la imagen de bienvenida al sistema.
9. Es una opción de recortar o ampliar el menú principal.
10. En este sector se observa el nombre del usuario que inicio sesión en el sistema, Es la opción mediante el cual cada usuario puede ver los datos de su usuario o salir del sistema.
11. Esta opción nos permite ver los datos de usuario que inicio la sesión. permite realizar el cambio de la contraseña de usuario y el cierre de la sesión o salir del sistema.
12. En esta parte se muestra el copyright y derechos reservados de los propietarios del sistema o menús secundarios del sistema.

3.4 FASE DE DESARROLLO

En esta fase de desarrollo, se realizara la codificación de cada una de las historias de usuario, tomando en cuenta la agrupación de ellas por medio de iteraciones tal cual se lo realizo en la fase de planeación.

Para alcanzar este propósito se conto con la participación activa de todos los miembros que se encuentran directamente relacionado a los procedimientos de venta dentro del almacén de ferretería de la asociación, permitiendo de esta forma comprender con más detalle lo que los usuarios del sistema quieren.

3.4.1 PRIMERA ITERACION

Durante esta primera iteración hubo varios inconvenientes, debido a la susceptibilidad de varios miembros de la asociación. Este hecho dificulto de forma gradual la facilidad de dialogo con los actores principales del sistema. Felizmente se pudo concretar del desarrollo con satisfacción.

En esta primera iteración se desarrolla las historias de usuario H1 y H2, junto con sus pruebas funcionales y capturas de pantalla.

H1: Control de acceso a usuarios
H2: Creación de roles personalizados

i. Tareas de ingeniería

Para el desarrollo de estas historias de usuario, se planteo un conjunto de tareas. Estas se planteo juntamente con el personal de la asociación quienes serán los usuarios del sistema, estas tareas que mencionamos las describimos a continuación.

Tabla: 3.9 Tarea- Lectura de usuario y contraseña

TAREA	
Número tarea: 1	**Número historia: 1**
Nombre tarea: Lectura de usuario y contraseña	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 25**
Fecha inicio: 27/02/2018	**Fecha fin: 1**/03/2018
Descripción: para ingresar al sistema o iniciar una sesión en ella, se muestra una ventana en el cual por medio de un formulario de datos se solicita el nombre de usuario y una contraseña. Además de dos botones o iconos los cuales indican ingresar al sistema o cancelar.	

Tabla: 3.10 Tarea comprobación de usuario en la base de datos

TAREA	
Número tarea: 2	**Número historia: 1**
Nombre tarea: Comprobación del usuario en la base de datos	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 55**
Fecha inicio: 02/03/2018	**Fecha fin: 07**/03/2018
Descripción: Una vez introducidos el usuario y contraseña, y pulsado el botón ingresar, se comprueban que ese usuario exista en la base de datos los cuales son administrador y vendedor, ambos con determinados permisos de acceso a la funcionalidades del sistema. Si el usuario no existe entonces el sistema vuelve a pedir el ingreso de usuario y contraseña	

Tabla: 3.11 Tarea-acceso a menús en el sistema

TAREA	
Número tarea: 3	**Número historia: 1**
Nombre tarea: acceso a menús que corresponda al usuario	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 60**
Fecha inicio: 08/03/2018	**Fecha fin:** 16/03/2018
Descripción: Una vez comprobada los datos del usuario, en pantalla se muestran sólo los menús de acceso a las partes de la aplicación que le corresponden al usuario, ocultando aquellos menús que no le correspondan.	

Tabla: 3.12 Tarea- cuenta administrador

TAREA	
Número tarea: 1	**Número historia: 2**
Nombre tarea: Cuenta administrador	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 30**
Fecha inicio: 17/03/2018	**Fecha fin:** 20/03/2018
Descripción: El sistema verificar los datos del usuario Administrador, luego el sistema muestra todos módulos y menús del sistema estos son: Administrar usuarios, inventarios, ventas, reportes.	

Tarea: 3.13 Tarea- cuenta vendedor

TAREA	
Número tarea: 2	**Número historia: 2**
Nombre tarea: Cuenta vendedor	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 22**
Fecha inicio: 21/03/2018	**Fecha fin:** 23/03/2018
Descripción: El sistema verifica los datos del usuario vendedor, luego el sistema muestra los menús correspondientes a este rol de usuario como ser: venta nueva, modificar venta y reversión. Además se habilita el botón de reportes	

ii. Tarjetas CRC

A continuación se muestra las tarjetas CRC de la primera iteración. En esta tarjeta CRC describe todas las clases que están involucradas para el desarrollo de esta iteración

Tabla: 3.14 Tarjeta CRC usuario

TARJETAS CRC	
NOMBRE DE LA CLASE: Usuario **DESCRIPCION:** Esta clase se encarga de todo lo relacionado a la gestión de usuarios y sus características	
RESPONSABILIDADES	**COLABORADORES**
• Crear usuario • Actualizar usuario • Consultar usuario • Eliminar usuario • Cambiar contraseña	• Roles

Tabla: 3.15 Tarjeta CRC roles

TARJETAS CRC	
NOMBRE DE LA CLASE: Roles DESCRIPCION: Esta clase se encarga de asignar un rol a un usuario y restringe ciertos módulos de acuerdo al tipo de usuario	
RESPONSABILIDADES	**COLABORADORES**
• Crear usuario administrador • Crear usuario vendedor	• Usuarios

iii. Desarrollo de prototipos

Para la implementación de esta primera iteración del sistema, se trabajo con todas las tareas de las historias de usuario planteadas. Así mismo como resultado de esta iteración se puede observar las siguientes vistas del sistema.

Figura: 3.5 Captura del pantalla del inicio de sesión del sistema

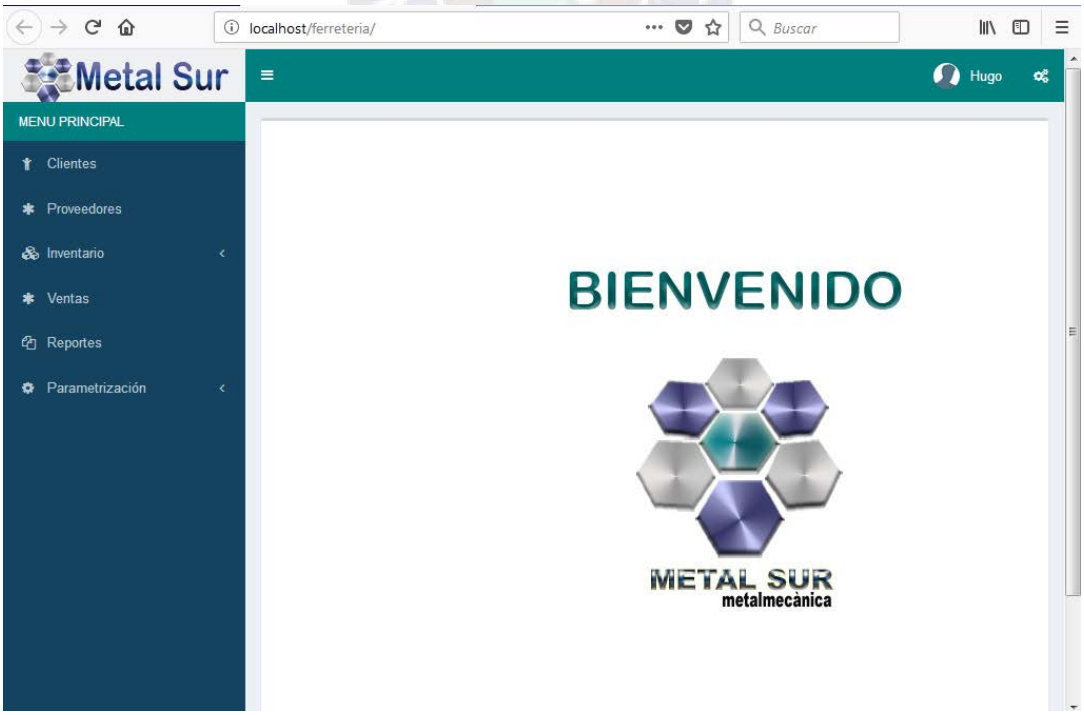


Figura: 3.6 Captura del pantalla bienvenida al sistema

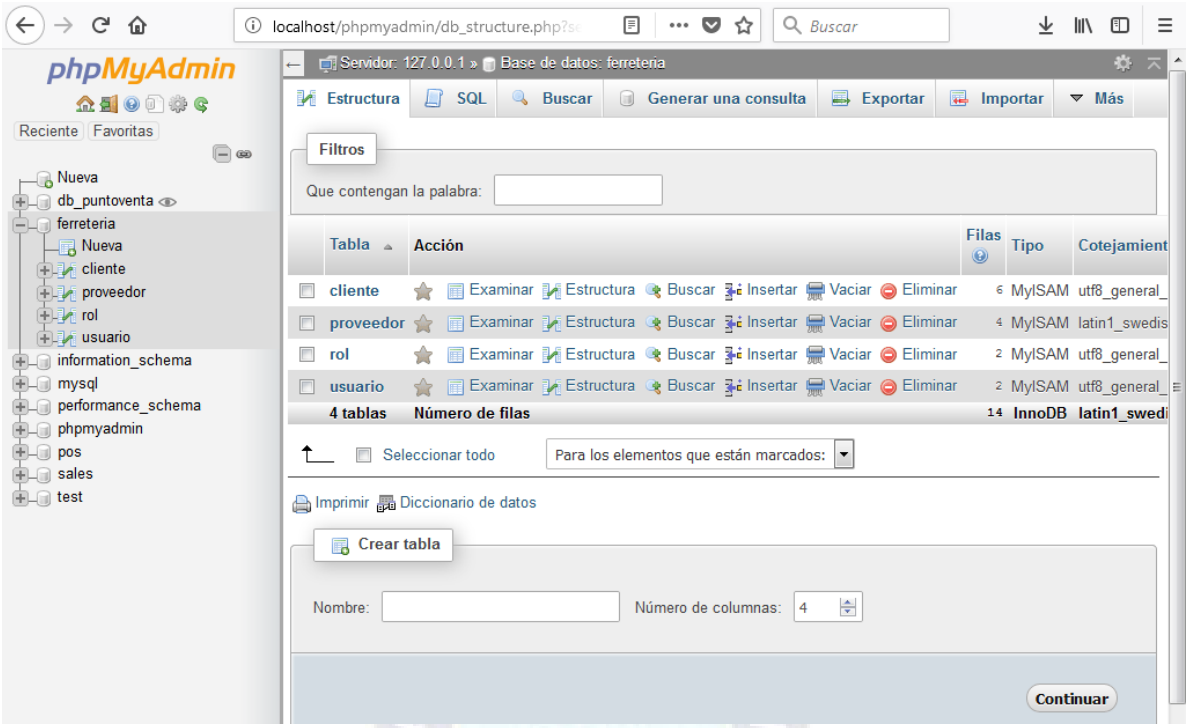


Figura: 3.7 Creación de tablas en la base de datos Ferretería

usuarioDAO.php

```php
<?php

include_once ('configRedBeanPHP.php');

class UsuarioDao {

    public static function recoverAll() {
        $sql = "SELECT *,u.id AS id FROM usuario AS u, rol AS r WHERE u.rol_id=r.id AND u.status='ACTI
        $arrayUsuario = R::getAll($sql);
        return($arrayUsuario);
    }

    public static function addupdate($usuario) {
        $exito = R::store($usuario);
        return $exito;
    }

    public static function recoverLoginPassword($user, $pass) {
        $sql = "SELECT *,u.id AS idUsuario FROM usuario AS u,rol AS r ";
        $sql.= "WHERE u.login='" . $user . "' AND password='" . $pass . "' AND u.rol_id=r.id ";
        $sql.= "AND u.status='ACTIVO'";
        $arrayUsuario = R::getAll($sql);
        return($arrayUsuario);
    }

}

?>
```

Figura: 3.8 código php para validar usuario en el sistema

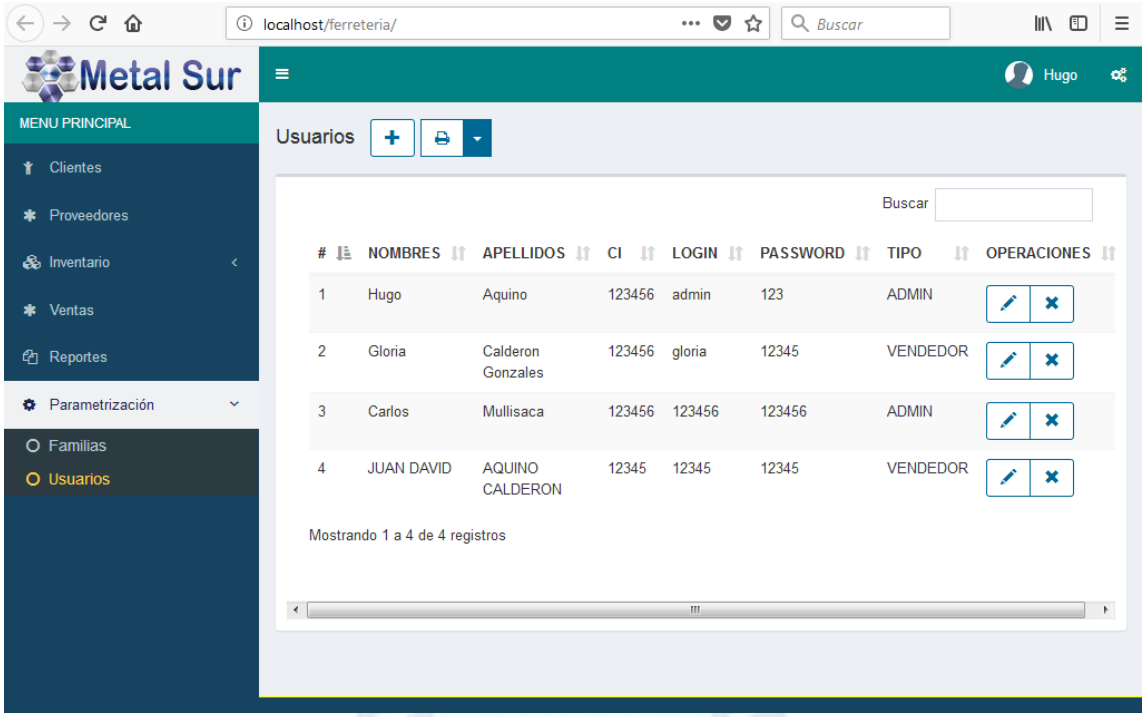


Figura: 3.9 Captura de pantalla – usuarios y roles

iv. Pruebas de aceptación

Tabla: 3.16 Prueba de aceptación de primera historia de usuario

CASO DE PRUEBA	
Nº1	**Nº Historia de usuario 1**
Nombre de historia de usuario: Control de acceso a usuario	
Condiciones de ejecución: Cada usuario debe contar con un nombre de usuario y su contraseña para acceder al sistema.	
Entrada/Pasos de ejecución: Escribir en el formulario de acceso al sistema el nombre de usuario y la contraseña y hacer clic en el botón aceptar.	
Resultado esperado: Acceso a las funcionalidades del sistema de acuerdo al tipo de usuario y el rol que desempeña.	
Evaluación de la prueba: La prueba se realizo satisfactoriamente.	

Tabla: 3.17 Prueba de aceptación segunda historia de usuario

CASO DE PRUEBA	
Nº2	**Nº Historia de usuario 2**
Nombre de historia de usuario: Administración de rol a usuario	
Condiciones de ejecución: El administrador del sistema asigna un rol a cada usuario los cuales pueden ser usuario administrador y usuario vendedor cada uno con diferentes opciones de operabilidad dentro del sistema.	
Entrada/Pasos de ejecución: Escribir en el formulario de acceso al sistema el nombre de usuario y la contraseña y hacer clic en el botón aceptar. Si el usuario es administrar se desplegara todas las opciones y menús del sistema por otra parte si es un usuario vendedor solo accederá al menú de ventas y consultas.	
Resultado esperado: Acceso a las funcionalidades del sistema de acuerdo al tipo de usuario y el rol que desempeña.	
Evaluación de la prueba: La prueba se realizo satisfactoriamente.	

3.4.2 SEGUNDA ITERACION

Durante esta segunda iteración se pudo trabajar con mayor fluidez ya que de cierto modo se construyo un lazo de amistad con los miembros de la asociación, por esta razón se hizo más fácil entender y corregir las necesidades del cliente para luego proceder con la codificación de todas las historias de usuario.

En esta iteración se desarrolla las historias de usuario H3 y H4 junto con sus pruebas funcionales y capturas de pantalla.

H3: Gestión de clientes
H4: Gestión de proveedores

i. Tareas de ingeniería

A continuación se muestran un conjunto de tareas que se implementara para cada historia de usuario

Tabla: 3.18 Tarea: adaptar la base de datos para clientes

TAREA	
Número tarea: 1	**Número historia: 3**
Nombre tarea: Adaptar la base de datos para clientes	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 30**
Fecha inicio: 11/04/2018	**Fecha fin: 12/04/2018**
Descripción: se debe implementar las relaciones necesarias en la base de datos para almacenar los datos de los clientes para el uso adecuado de su información y realizar las consultas respectivas por los usuarios del sistema.	

Tabla: 3.19 Tarea-Diseño de interfaz de usuario clientes

TAREA	
Número tarea: 2	**Número historia: 3**
Nombre tarea: Diseño de interfaz de usuario para registro de clientes	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 35**
Fecha inicio: 11/04/2018	**Fecha fin: 12/04/2018**
Descripción: Se debe implementar un formulario para el registro de los datos del cliente. Estos datos serán el nombre, dirección, nro. de cedula de identidad , NIT y otros datos necesarios.	

Tabla: 3.20 Tarea-Administrar clientes

TAREA	
Número tarea: 3	**Número historia: 3**
Nombre tarea: administrar clientes	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 35**
Fecha inicio: 11/04/2018	**Fecha fin: 12/04/2018**
Descripción: Se debe implementar los mecanismos necesarios para acceder a la base de datos, de esta forma se pueda realizar la alta, modificación o eliminación de un cliente.	

Tabla: 3.21 Tarea-Adaptar la base de datos para proveedores

TAREA	
Número tarea: 1	**Número historia: 4**
Nombre tarea: Adaptar la base de datos para almacenar la información respecto a los proveedores.	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs): 25**
Fecha inicio: 11/04/2018	**Fecha fin: 12/04/2018**
Descripción: se debe implementar las relaciones necesarias en la base de datos para almacenar los datos de los proveedores para el uso adecuado de su información y realizar las consultas respectivas por los usuarios del sistema.	

Tabla: 3.22 Tarea-Diseño de interfaz para registro de proveedores

TAREA	
Número tarea: 2	**Número historia: 4**
Nombre tarea: Diseñar una interfaz de usuario para el registro de los proveedores	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs):25**
Fecha inicio: 12/04/2018	**Fecha fin: 16/04/2018**
Descripción: Se implementara un formulario datos que permita introducir el nombre del proveedor, con toda la descripción de los dato del proveedor, además se comprobara que estos datos son almacenados y guardados correctamente en la base de datos.	

Tabla: 3.23 Tarea-Administrar proveedores

TAREA	
Número tarea: 3	**Número historia: 4**
Nombre tarea: Administrar proveedores	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo (Hrs):25**
Fecha inicio: 16/04/2018	**Fecha fin: 17/04/2018**
Descripción: Se debe implementar los mecanismos necesarios para acceder a la base de datos, de esta forma se pueda realizar la alta, baja y eliminación de un proveedor. Por otra parte los datos necesarios para la creación de un nuevo producto serán un código de proveedor, nombre, dirección, teléfono y otros.	

ii. Tarjetas CRC

A continuación se muestra el diseño de las tarjetas CRC para esta iteración, en la que nos muestra un número de responsabilidades y sus colaboradores

Tabla: 3.24 Tarjeta CRC-Proveedores

TARJETAS CRC	
NOMBRE DE LA CLASE: Proveedores **DESCRIPCION:** Esta clase contiene los datos con relación a los proveedores	
RESPONSABILIDADES	**COLABORADORES**
• Crear nuevo proveedor • Actualizar dato de proveedor • Buscar proveedor • Eliminar proveedor	• Productos • personas

Tabla: 3.25 Tarjeta CRC-Clientes

TARJETAS CRC	
NOMBRE DE LA CLASE: Clientes **DESCRIPCION:** Esta clase contiene los datos con relación a los clientes que realizan sus pedidos de compra a la ferretería de la asociación.	
RESPONSABILIDADES	**COLABORADORES**
• Crear nuevo Cliente • Actualizar dato de clientes • Buscar cliente • Eliminar cliente	• clientes • personas

iii. Desarrollo y captura de pantallas

El resultado que se obtuvo para esta iteración son las siguientes pantallas.

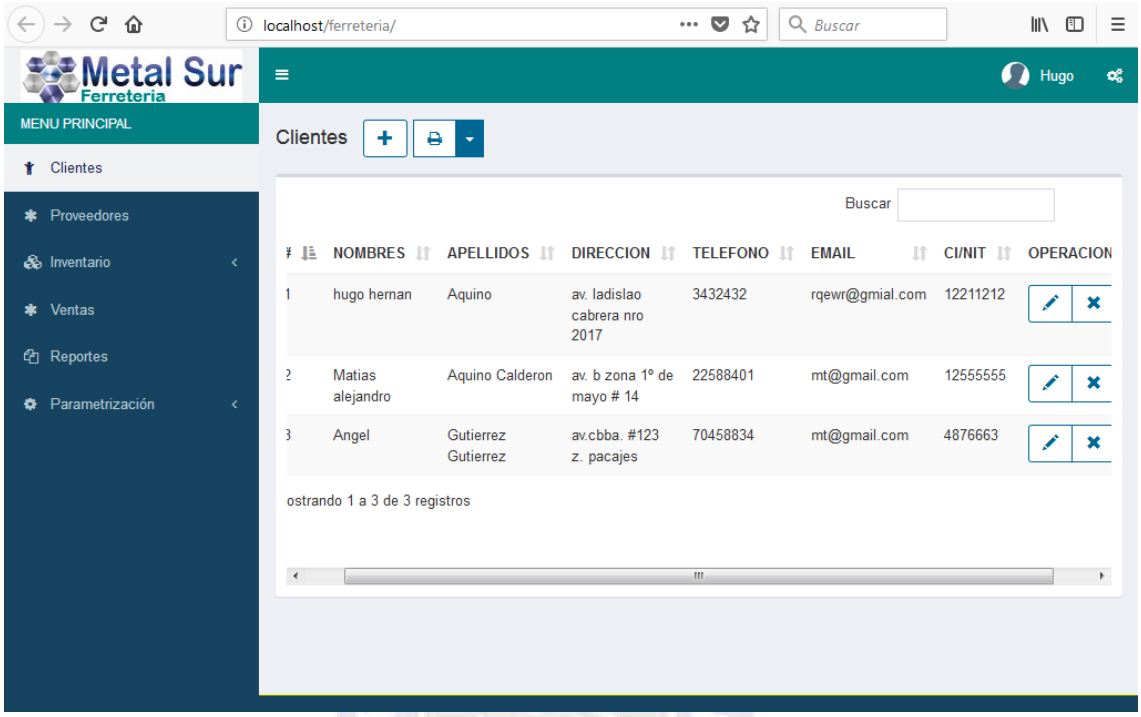


Figura: 3.10 Administrar clientes

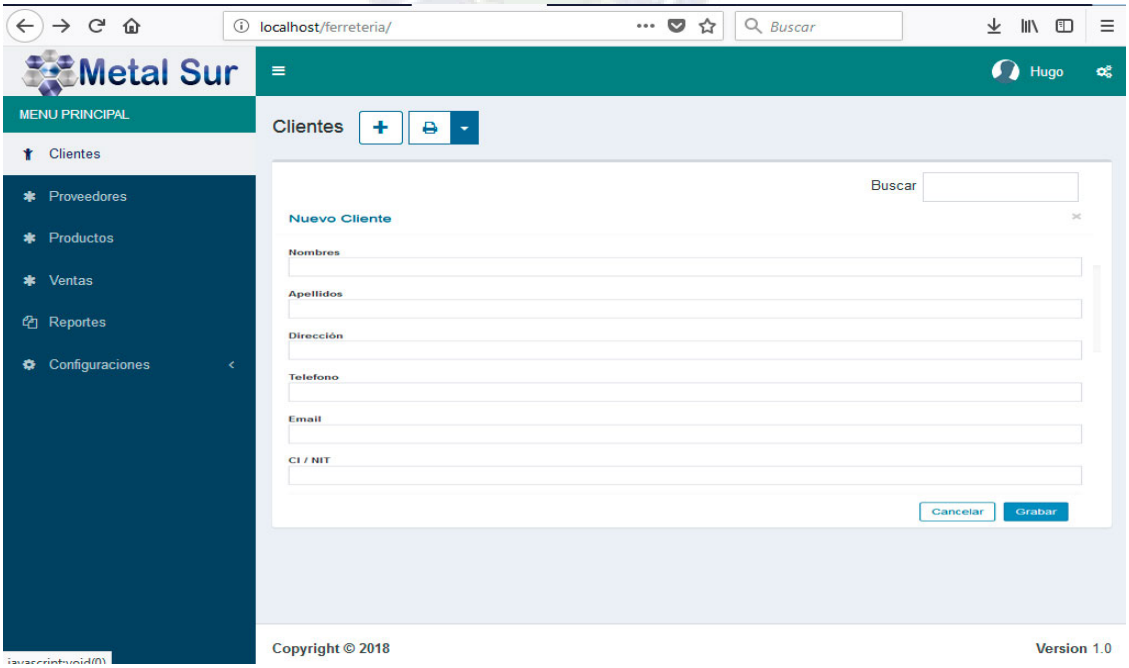


Figura: 3.11 Adicionar nuevo cliente

clienteDAO.php

```php
<?php

include_once ('configRedBeanPHP.php');

class ClienteDao {

    public static function getAll() {
        $sql = "SELECT * FROM cliente AS c WHERE c.estado='ACTIVO' ";
        $sql .= " ORDER BY c.id DESC";
        $arrayCliente = R::getAll($sql);
        return($arrayCliente);
    }

    public static function addupdate($cliente) {
        $exito = R::store($cliente);
        return $exito;
    }

    public static function getById($id) {
        $cliente = R::load("cliente", $id);
        return($cliente);
    }

    public static function existeCiNit($cinit) {
        $sql = "SELECT * FROM cliente AS c";
        $sql .= " WHERE c.cinit='" . $cinit . "' AND c.estado='ACTIVO' ";
        $arrayCliente = R::getAll($sql);
        if (count($arrayCliente) == 1) {
            return true;
        } else {
            return false;
```

Figura: 3.12 Código php Adición de clientes a la base de datos

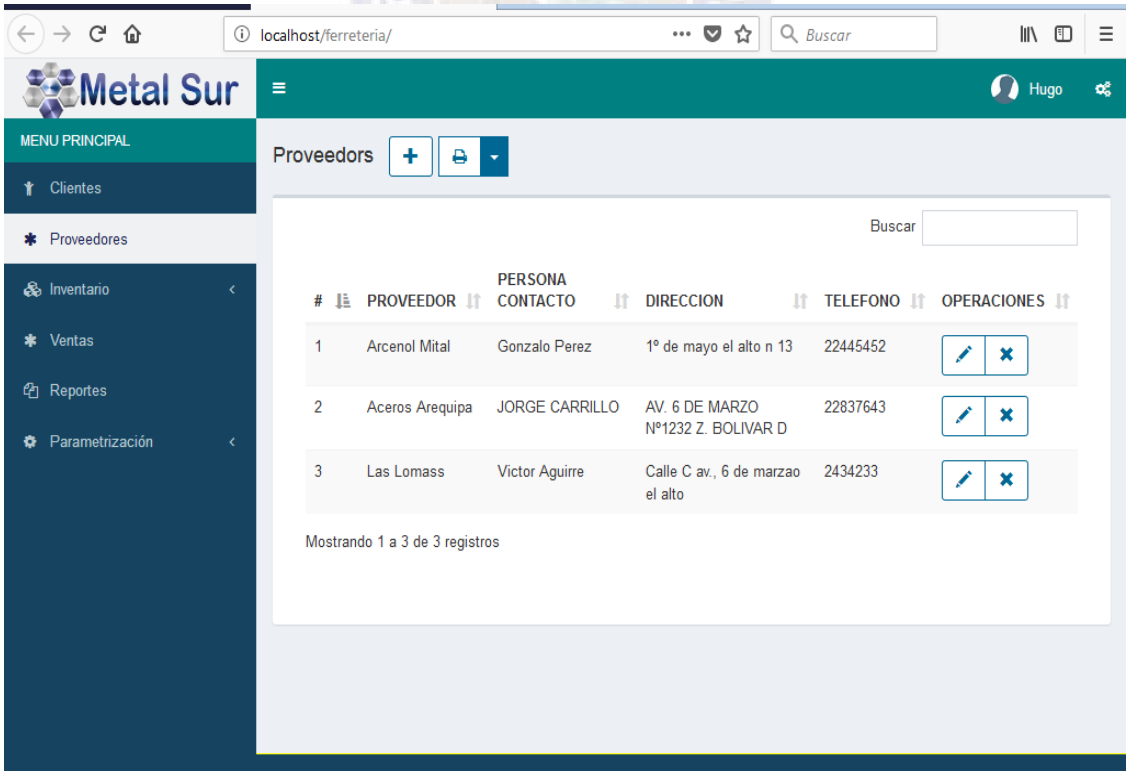


Figura: 3.13 administrar proveedores

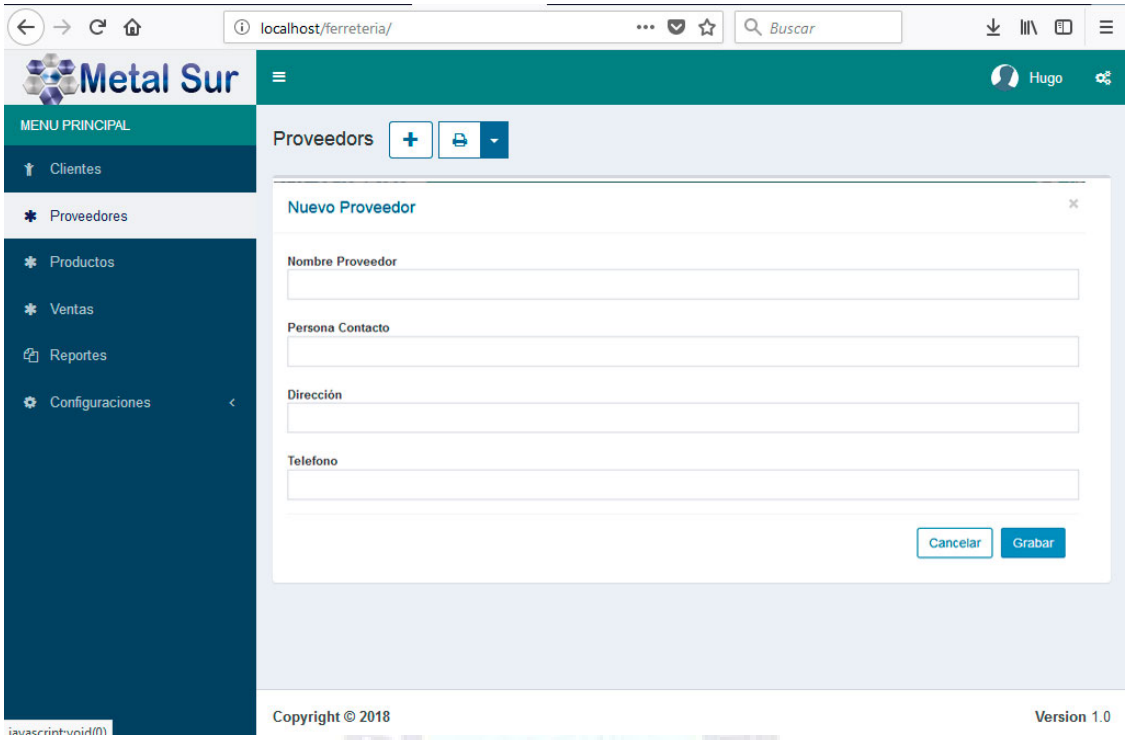


Figura: 3.14 adicionar proveedor

proveedorDAO.php

```php
<?php

include_once ('configRedBeanPHP.php');

class ProveedorDao {

    public static function getAll() {
        $sql = "SELECT * FROM proveedor AS p WHERE p.estado='ACTIVO' ";
        $sql .= " ORDER BY p.id DESC";
        $arrayProveedor = R::getAll($sql);
        return($arrayProveedor);
    }

    public static function addupdate($proveedor) {
        $exito = R::store($proveedor);
        return $exito;
    }

    public static function getById($id) {
        $proveedor = R::load("proveedor", $id);
        return($proveedor);
    }

    public static function existeNombreProveedor($nombreProveedor) {
        $sql = "SELECT * FROM proveedor AS p";
        $sql .= " WHERE p.nombre_proveedor='" . $nombreProveedor . "' AND p.estado='ACTIVO' ";
        $arrayProveedor = R::getAll($sql);
        if (count($arrayProveedor) == 1) {
            return true;
        } else {
            return false;
```

Figura: 3.15 Adición de proveedores a la base de datos

iv. Pruebas de aceptación

Tabla 3.26 Prueba de aceptación gestión de clientes

CASO DE PRUEBA	
Nº1	**Nro. Historia de usuario 3**
Nombre de historia de usuario: gestión de clientes	
Condiciones de ejecución: El administrador del sistema o el vendedor de la ferretería deben haber iniciado una sesión en el sistema.	
Entrada/Pasos de ejecución: Selecciona la opción clientes, una vez ahí se muestra la opción de adicionar y en el caso de que existiera ya varios clientes se habilitan la opción de modificar o eliminar a algún cliente. También existe la opción de búsqueda por el nombre de cliente.	
Resultado esperado: se almaceno la información de un nuevo registro y se habilito las opciones de modificar o eliminar.	
Evaluación de la prueba: La prueba se realizo satisfactoriamente.	

Tabla: 3.27 Prueba de aceptación gestión de proveedores

CASO DE PRUEBA	
Nº1	**Nro. Historia de usuario 4**
Nombre de historia de usuario: gestión de proveedores	
Condiciones de ejecución: : El administrador del sistema o el vendedor de la ferretería debe haber iniciado una sesión en el sistema.	
Entrada/Pasos de ejecución: del menú principal se selecciona la opción proveedores, una vez ahí se muestra la opción de adicionar y en el caso de que existiera ya varios proveedores se habilitan la opción de modificar o eliminar. También existe la opción	

de búsqueda por el nombre de proveedor.
Resultado esperado: se almaceno la información de un nuevo registro y se habilito las opciones de modificar o eliminar.
Evaluación de la prueba: La prueba se realizo satisfactoriamente.

3.4.3 TERCERA ITERACION

Por último desarrollamos las historias de usuarios H5 y H6, para los cuales se plantearon las siguientes tareas que desarrollamos a continuación juntamente con sus capturas de pantallas y pruebas funcionales.

H5: Gestión de productos y artículos
H6: Gestión de ventas

i. Tareas de ingeniería

Tabla: 3.28 Tarea: Diseño de una interfaz para productos

TAREA	
Número tarea: 1	**Número historia: 5**
Nombre tarea: Diseño de una interfaz para el registro de productos y/o artículos al sistema	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo** (Hrs):40
Fecha inicio: 28/04/2018	**Fecha fin: 02/05/2018**
Descripción: Se debe implementar una interfaz para el usuario, donde se podrá hacer una búsqueda de un determinado producto y se pueda listar los productos existentes.	

Tabal: 3.29 Tarea-Adaptar la base de datos para el almacenamiento de productos

TAREA	
Número tarea: 2	**Número historia: 5**
Nombre tarea: Adaptar la base de datos para el almacenamiento de productos y artículos	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo** (Hrs):48
Fecha inicio: 03/05/2018	**Fecha fin:** 08/05/2018
Descripción: Se deben crear las relaciones correspondientes entre los productos y los proveedores en la base de datos, de tal modo que al momento de registrar un producto se pueda identificar cual es el proveedor.	

Tabla: 3.30 Tarea: administrar productos

TAREA	
Número tarea: 3	**Número historia: 5**
Nombre tarea: administrar productos	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo** (Hrs):64
Fecha inicio: 10/05/2018	**Fecha fin:** 18/05/2018
Descripción: Se debe implementar los mecanismos necesarios para acceder a la base de datos, de esta forma se pueda realizar la alta, baja y eliminación de un producto. Por otra parte los datos necesarios para la creación de un nuevo producto serán un código de producto, nombre, descripción entre otros.	

Tabla: 3.32 Tarea-Creación en la base de datos la tabla ventas

TAREA	
Número tarea: 1	**Número historia: 6**
Nombre tarea: Adecuar la base de datos para el almacenamiento y registro de las ventas.	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo** (Hrs):48
Fecha inicio: 21/05/2018	**Fecha fin:** 26/05/2018
Descripción: Se implementara en la base de datos las relaciones adecuadas para almacenar los datos que corresponde a una venta como: número de venta, código de cliente, fecha y otros datos necesarios.	

Tabla: 3.31 Tarea-Diseño de interfaz para las ventas

TAREA	
Número tarea: 2	**Número historia: 6**
Nombre tarea: Diseño de interfaz y formulario de venta	
Tipo de tarea : Desarrollo	**Tiempo de desarrollo** (Hrs):80
Fecha inicio: 29/05/2018	**Fecha fin:** 07/06/2018
Descripción: Se crea una interfaz grafica que muestre un listado de las ultimas ventas realizadas de cada usuario cada una de ellas con las opciones de ver el detalle de venta. Se adicionara la opción de generar una nueva venta en la que además de debe adicionar un botón que permita introducir un nuevo producto o artículos al pedido de venta. En esta interfaz se mostrara el nombre del producto, con la descripción necesaria de cada producto que se vende, cantidad, precio unitario y el costo total.	

ii. Tarjetas CRC

Tabla: 3.33 Tarjeta CRC-Productos

TARJETAS CRC	
NOMBRE DE LA CLASE: Productos/artículos **DESCRIPCION:** Esta clase contiene los datos más relevantes de los productos	
RESPONSABILIDADES	**COLABORADORES**
• Crear nuevo producto • Modificar producto • Listar producto • Eliminar producto	• compras • proveedores

Tabla: 3.34 Tarjeta CRC-Venta

TARJETAS CRC	
NOMBRE DE LA CLASE: ventas **DESCRIPCION:** Esta clase contiene los datos más relevantes de las ventas	
RESPONSABILIDADES	**COLABORADORES**
• Generar nueva venta • Modificar venta • Listar venta • Eliminar venta • Detalle de venta	• Cliente • Producto • Usuario • Categoría

iii. Desarrollo y captura de pantallas

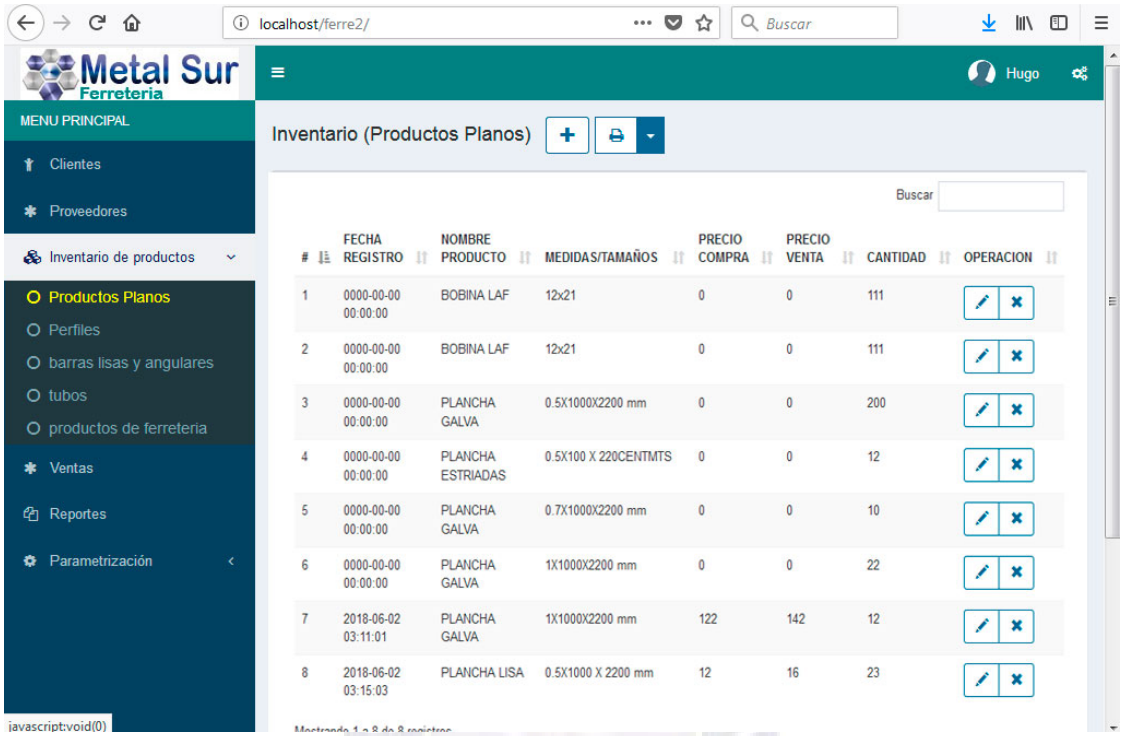


Figura: 3.16 listado de productos de la categoría productos planos

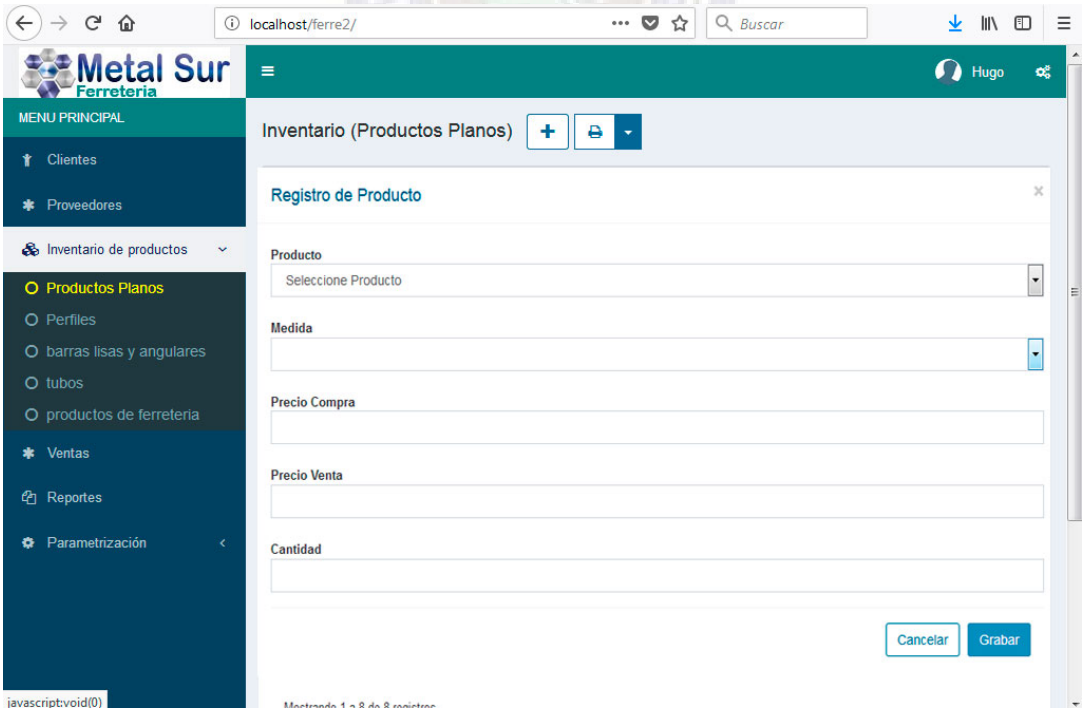


Figura: 3.17 Captura de pantalla adicionar nuevo producto

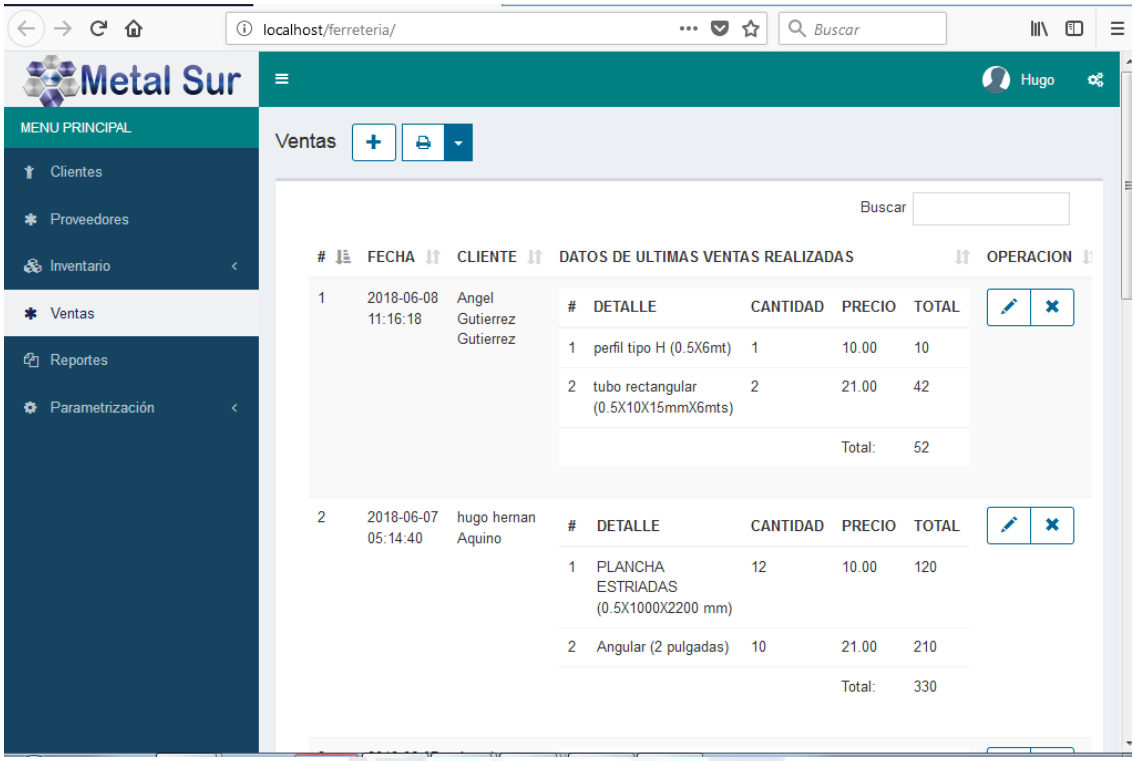


Figura: 3.18 Ingreso a opción ventas – ultimas ventas

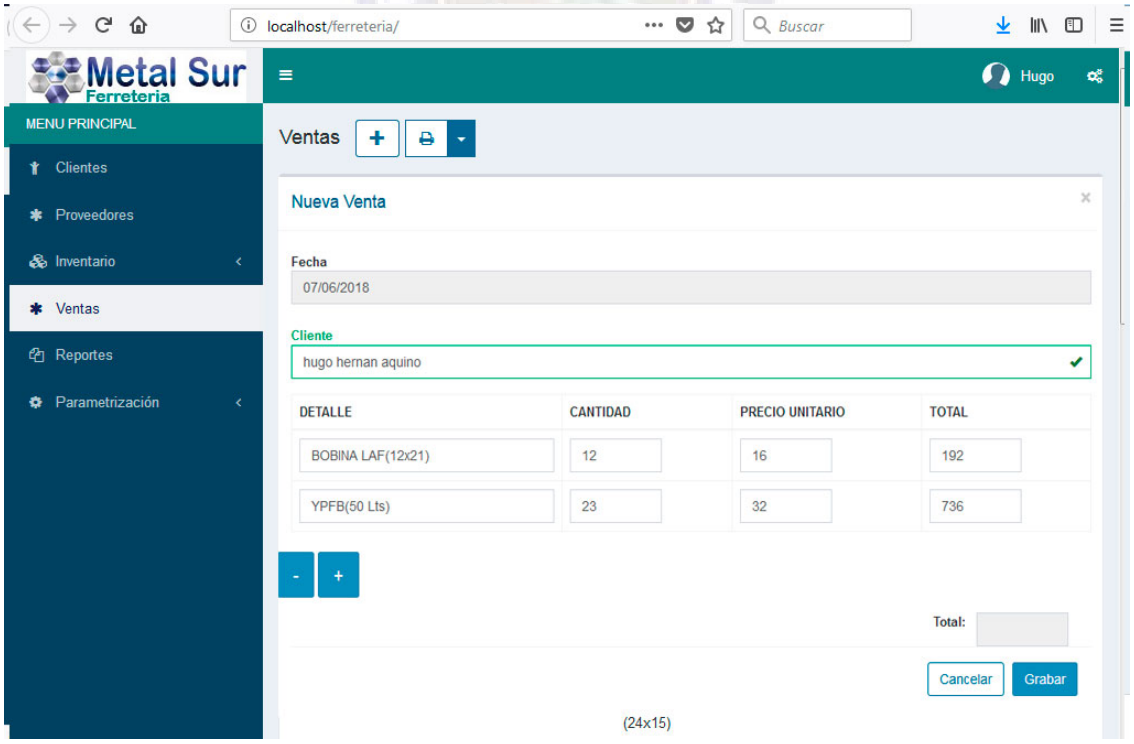


Figura: 3.19 Generar nueva venta

```
ventas.php
50      </thead>
51      <tbody>
52          <?php for ($i = 0; $i < count($arrayVenta); $i++) {
53              ?>
54              <tr>
55                  <td><?php echo ($i + 1); ?></td>
56
57                  <td><?php echo $arrayVenta[$i]['fecha_registro']; ?></td>
58                  <td><?php echo $arrayVenta[$i]['nombres'] . " " . $arrayVenta[$i]['apellidos']; ?></td>
59                  <td>
60                      <table class="table">
61                          <thead>
62                              <tr>
63                                  <th class = "a66-number">#</th>
64                                  <th>DETALLE</th>
65                                  <th>CANTIDAD</th>
66                                  <th>PRECIO</th>
67                                  <th>TOTAL</th>
68                              </tr>
69                          </thead>
70                          <tbody>
71                              <?php
72                              $arrayVentaDetalle = VentaDetalleDao::getByIdVenta($arrayVenta[$i]['idVen']);
73                              $total = 0;
74                              for ($k = 0; $k < count($arrayVentaDetalle); $k++) {
75                                  $totalRow = $arrayVentaDetalle[$k]['cantidad'] * $arrayVentaDetalle[$k]['pre
76                                  ?>
77                                  <tr>
78                                      <td><?php echo ($k + 1); ?></td>
79                                      <td><?php echo $arrayVentaDetalle[$k]['nombre_producto'] . " (" . $array
80                                      <td><?php echo $arrayVentaDetalle[$k]['cantidad']; ?></td>
HTML
```

Figura: 3.20 Código php para generar ultimas ventas

iv. Pruebas de aceptación

Tabla: 3.35 Caso de prueba Gestión de Productos

CASO DE PRUEBA	
Nº1	**Nro. Historia de usuario 5**
Nombre de historia de usuario: Gestión de Productos	
Condiciones de ejecución: El administrador del sistema de la ferretería debe haber iniciado una sesión en el sistema.	
Entrada/Pasos de ejecución: Selecciona la opción ventas, luego se despliega las opciones de tipo de familia para luego seleccionar alguna de ellas, en el caso de que existiera ya varios productos registrados estos se muestran de acuerdo al último producto registrado y se habilitan	

las opciones de adicionar nuevo, modificar y eliminar a algún producto. También existe la opción de búsqueda por el código o nombre de producto.
Resultado esperado: al ingresar a esta opción se puedo evidenciar que se almaceno la información de un nuevo registro y se habilito las opciones de modificar o eliminar para los productos existentes.
Evaluación de la prueba: La prueba se realizo satisfactoriamente.

Tabla: 3.36 Caso de prueba - Gestión de Ventas

CASO DE PRUEBA	
Nº1	**Nro. Historia de usuario 6**
Nombre de historia de usuario: Gestión de Ventas	
Condiciones de ejecución: El administrador del sistema o el usuario vendedor del sistema de la ferretería deben haber iniciado una sesión en el sistema.	
Entrada/Pasos de ejecución: Selecciona la opción ventas, dentro de esta opción se muestra las ultimas ventas realizadas por el usuario y se ve el icono de realizar una nueva venta en la que se llena un formulario con los siguientes datos: fecha, nombre de cliente, CI o NIT y la cantidad del producto a vender con su descripción y el precio total de venta.	
Resultado esperado: se almaceno la información de un nuevo registro de ventas y se habilito las opciones de modificar o eliminar.	
Evaluación de la prueba: La prueba se realizo satisfactoriamente.	

3.5 SEGURIDAD

La seguridad en el sistema es un conjunto de medidas preventivas que se aplican para resguardar y proteger la información, de esta forma se busca la confidencialidad, disponibilidad y la integridad de información.

3.5.1 SEGURIDAD A NIVEL DE LA APLICACION

- ✓ **Verificación de acceso:** El sistema tiene desarrollado un módulo de verificación y validación de usuarios (login y password), cada usuario inicia una sesión para hacer uso del sistema.
- ✓ **Niveles de acceso**: Al sistema pueden acceder diferentes tipos de usuarios (Administrador del sistema y el usuario de ventas) de la institución, el sistema restringe el acceso a procesos en base a la asignación de un rol.

3.5.2 SEGURIDAD A NIVEL DE LA BASE DE DATOS

El acceso a la base de datos debe ser protegido, para evitar la pérdida de información

Copias de seguridad de la base de datos: Una de las prioridades es contar con copias de seguridad, el proceso consiste en guardar un backup de la base de datos cada cierto periodo de tiempo en un medio de almacenamiento como el disco duro y también un medio almacenamiento extraíble.

La copia de seguridad se guardara como resguardo a cualquier percance no previsto. Por lo cual el administrador o encargado del sistema deberá sacar una copia de la base de datos. Los archivos deberán contener los siguientes datos al momento de grabarlos: Fecha y hora.

3.6 CALIDAD

Para evaluar la calidad de este proyecto se establecieron diferentes aspectos o atributos que nos refleja la calidad de este software como: funcionalidad, usabilidad, mantenibilidad y satisfacción del usuario.

- ✓ **Funcionalidad.** La funcionalidad se refiere a la capacidad del producto software proveer los servicios necesarios para cumplir con los requisitos funcionales.

Con respecto a la adaptabilidad se puede decir que el sistema es multiplataforma, la interfaz de usuario se adapta a cualquier navegador web, como también a cualquier dispositivo que se utilice como terminal.

La exactitud consiste la capacidad de entregar resultados correctos. Después de que el sistema fue probado por un periodo de tiempo por los usuarios, no se encontraron errores en cuanto a los resultados. En consecuencia cumple con la exactitud.

La interoperabilidad es la habilidad de interactuar con determinados sistemas. El sistema está construido de tal manera que puede crearse módulos para interactuar con otros sistemas como por ejemplo la generación y el consumo de servicios web.

La seguridad es la habilidad de prevenir uso no autorizado, tanto intencional como accidental, de programas y datos. Al sistema solo se puede acceder a través de un usuario y contraseña, el sistema restringe el acceso a procesos en base a la asignación de un rol, la configuración adecuada del servidor y el acceso a la base de datos aumentan el nivel de seguridad. Entonces el sistema cumple con la seguridad.

- ✓ **Usabilidad:** Se realiza una evaluación del sistema en base a encuestas de un grupo de usuarios.

Tabla: 3.35 Preguntas de usabilidad

PREGUNTA	PONDERACION (%)
¿La interfaz de usuario es amigable?	**90**
¿El sistema puede ser usado facialmente con pocos conocimientos sobre computación?	**95**
¿El acceso a los distintos menús es fácil y sencillo?	**99**
¿Los iconos permiten reconocer con facilidad las funciones del sistema?	**99**
¿El cumple con todos los requerimientos?	**99**

¿Los datos de salida son confiables?	98
¿Los botones de enlace trabajan correctamente?	95

La fórmula para calcular la usabilidad de un sistema es el siguiente

$$U = \frac{\sum_{i=1}^{n} Xi}{n}$$

Donde:

$\sum_{i=1}^{n} Xi$ = es la suma de ponderaciones

n = cantidad de preguntas

Reemplazando datos nos queda de la siguiente forma

$$\sum_{i=1}^{n} Xi = 90 + 95 + 99 + 99 + 99 + 98 + 95 = 675$$

n=7

$$U = \frac{\sum_{i=1}^{n} Xi}{n} = \frac{675}{7} = 96.43$$

El resultado nos muestra la usabilidad y es de un 96.43 %

- ✓ **Facilidad de mantenimiento:** Para calcular la facilidad de mantenimiento se tomó los datos de la segunda iteración.

  M_T = numero de módulos de la liberación actual

  F_C = numero de módulos de la liberación actual que cambiaron

  F_A = numero de módulos de la liberación actual que se agregaron

  F_D = numero de módulos de la liberación anterior que se borraron en la liberación actual

  M_T = 3

M_C = 1

M_A = 0

M_D = 0

El Índice de Madurez del Software (IMS) se calcula mediante la siguiente ecuación

.

$$IMS = \frac{\mathrm{MT}-(\mathrm{FC}+\mathrm{FA}+\mathrm{FD})}{\mathrm{MT}} = \frac{3-(1+0+0)}{3} = 0.66$$

Por tanto el Índice de Madurez de Software es de un 66%, lo que significa que es mantenible.

CAPITULO IV CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Luego de analizar y observar los inconvenientes presentados dentro del almacén de ferretería de la Asociación de Productores en Metalmecánica Sur ”METAL SUR” y tomando en cuenta los objetivos planteados se tiene las siguientes conclusiones y recomendaciones.

- ✓ Con la implementación del nuevo sistema, se logro mejorar el proceso de registro de las ventas y el manejo del inventario de los productos de la ferretería. Con esta información se podrá conocer con exactitud la cantidad de productos y artículos existentes de cada uno de ellos.
- ✓ Toda la información de la ferretería respecto al registro de los productos, clientes, proveedores, ventas y el inventario de existencias ahora se almacena en una base de datos.
- ✓ Se diseñó de manera eficiente la estructura de la aplicación web, mediante interfaces sencillas de manipular para el usuario, de tal modo que se pudo facilitar la interacción de los usuarios con la aplicación web.
- ✓ Con el nuevo sistema se logro reducir el tiempo en la elaboración de reportes del inventario, anteriormente lo se realizaba en un día, ahora se puede realizar en cinco minutos.
- ✓ Se logro poner en funcionamiento la aplicación web en la asociación METAL SUR, el cual queda a disposición del personal administrativo y velar que se cumpla el objetivo por el cual fue diseñado e implementado.

4.2 RECOMENDACIONES

- ✓ Se recomienda que una vez puesto en funcionamiento la aplicación web de registro de compras, ventas e inventario, este se alimente de la manera más rápida con la información actual del inventario. Esto contribuirá a que la asociación pueda realizar un control inmediato de sus productos.
- ✓ Una vez implementado la aplicación web en la asociación METAL SUR, se recomienda que este sea manipulado por un personal idóneo en el manejo del mismo, ya que de la eficiente manipulación se alcanzara los objetivos planteados en el proceso de registro y control de inventarios.
- ✓ Se recomienda realizar el mantenimiento preventivo de los equipos donde será implantado el nuevo sistema, para evitar posibles fallas durante el funcionamiento de éste.

BIBLIOGRAFIA

[1] Ian Sommerville. Ingeniería del software 7ma ed

[2] Roger Pressman. Ingeniería del software un enfoque practico 6ta ed. (2005)

[3] Roger Pressman. Ingeniería del software un enfoque practico 7ta ed.(2010)

[4] G. Choque. Ingeniería del software principios y conceptos 2002

[5] J. Joskowicz. Reglas y Practicas en eXtreme programming

L. Echeverry y L. Delgado. Metodología agil XP al desarrollo de software (universidad Tecnologica de Pereira 2007).

[6] Willam Stallings. Criptografía y Seguridad de la Red 4ta ed. Principios y Práctica

[7] Hamdy A. Taha. Investigación de operaciones 7ta edición. (2004)

[8] Estudio de UWE (UML-based web Engineering) Universidad Carlos Tercero Madrid

[9] Moya Navarro, Marcos Javier. Control de inventarios y teoría de colas, EUNED, 1999

[10] Muller, Max. Fundamentos de administración de inventarios, Editorial Norma, 2005

REFERENCIAS WEB

[1] http://oness.sourceforge.net/proyecto/html/ch05.htm

[2] http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp

[3] http://programacionextrema.tripod.com

[4] http://www.extremeprogramming.org

ANEXOS

ANEXO A DIAGRAMAS DE CASOS DE USO

DIAGRAMA DE CASO DE USO GENERAL

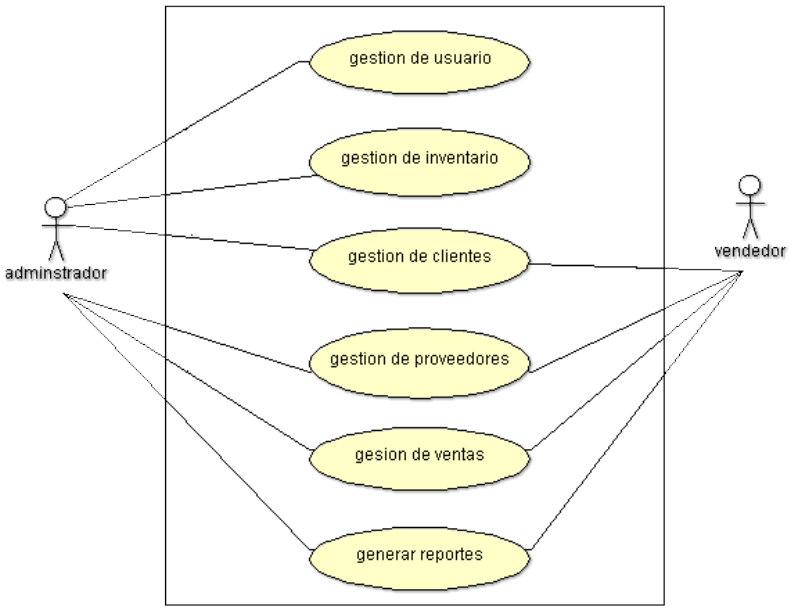


DIAGRAMA DE CASO DE USO EXPANDIDO ADMINISTRAR USUARIO

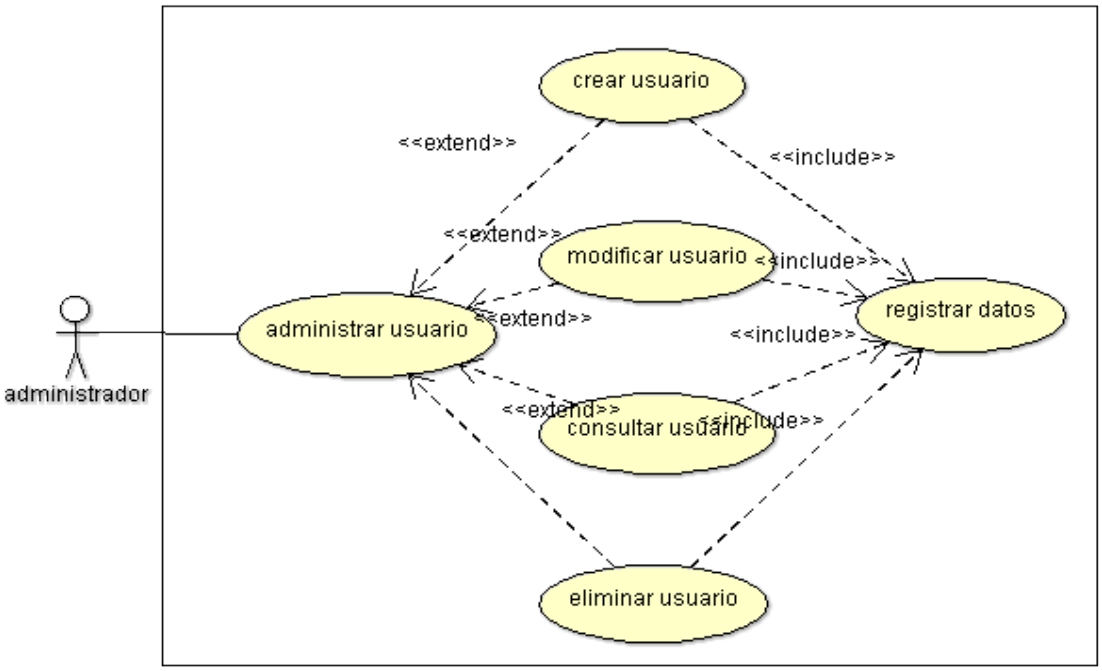


DIAGRAMA DE CASO DE USO EXPANDIDO GESTIONAR INVENTARIO

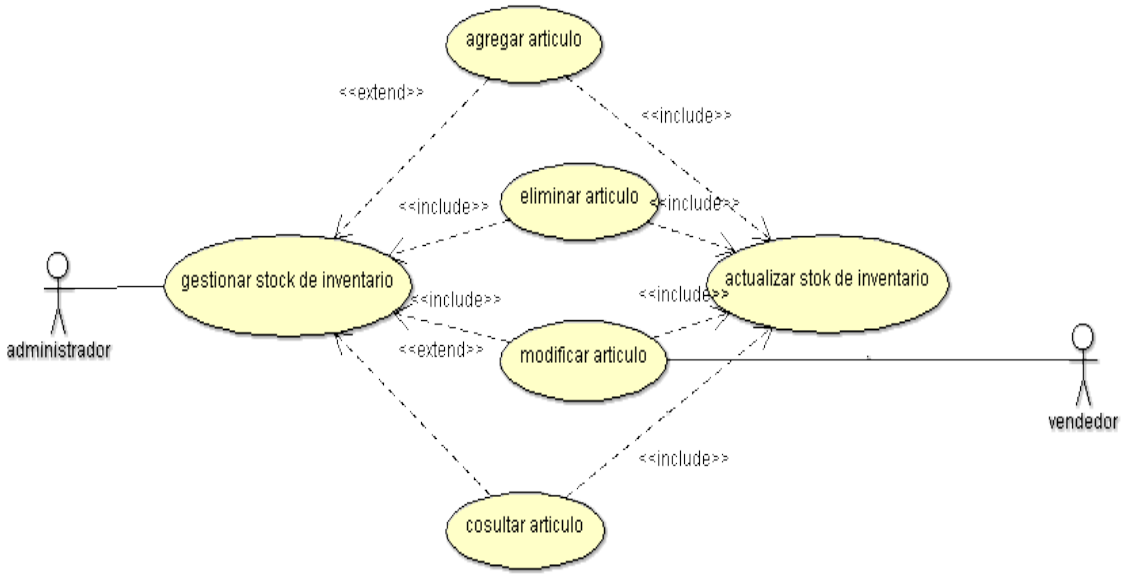


DIAGRAMA DE CASO DE USO EXPANDIDO GESTION DE CLIENTES

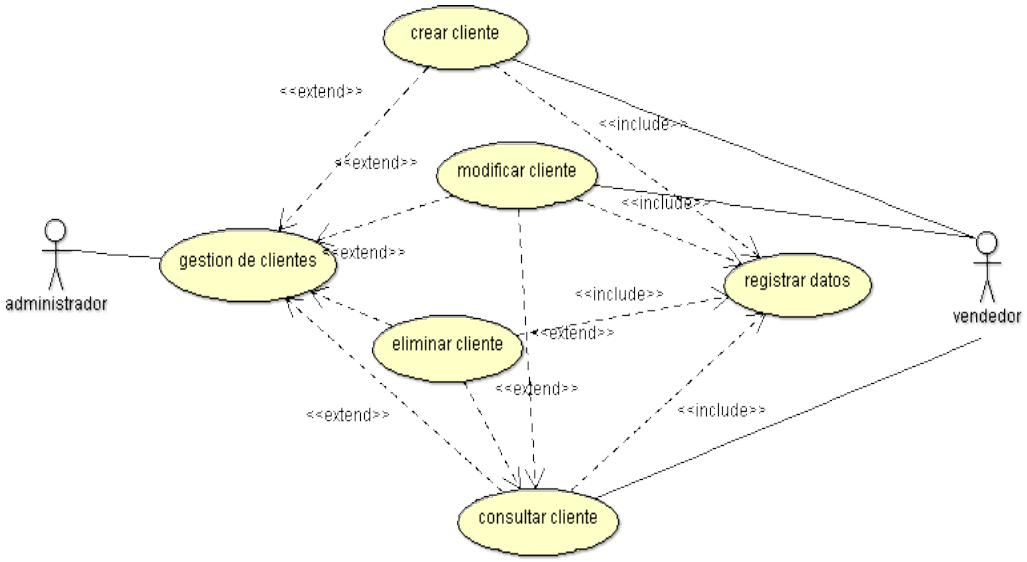


DIAGRAMA DE CASO DE USO EXPANDIDO GESTION DE PROVEEDORES

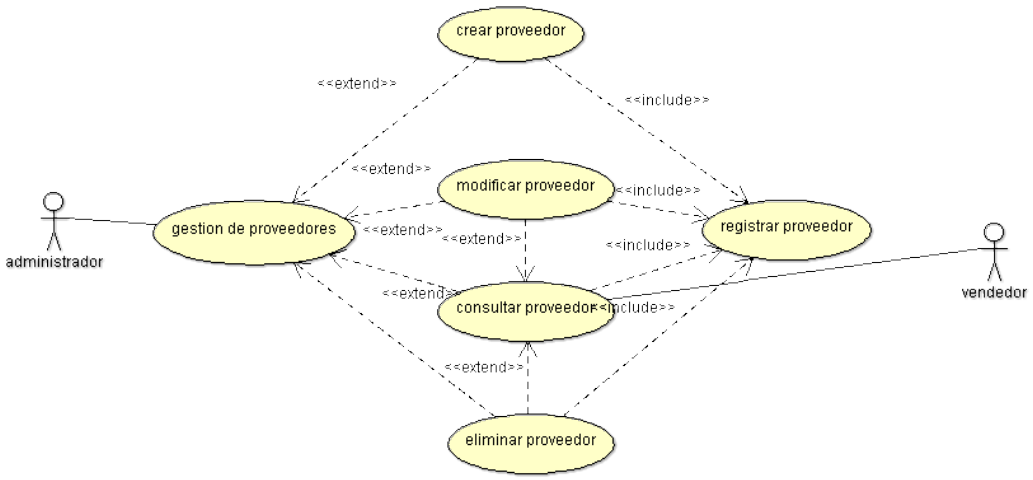
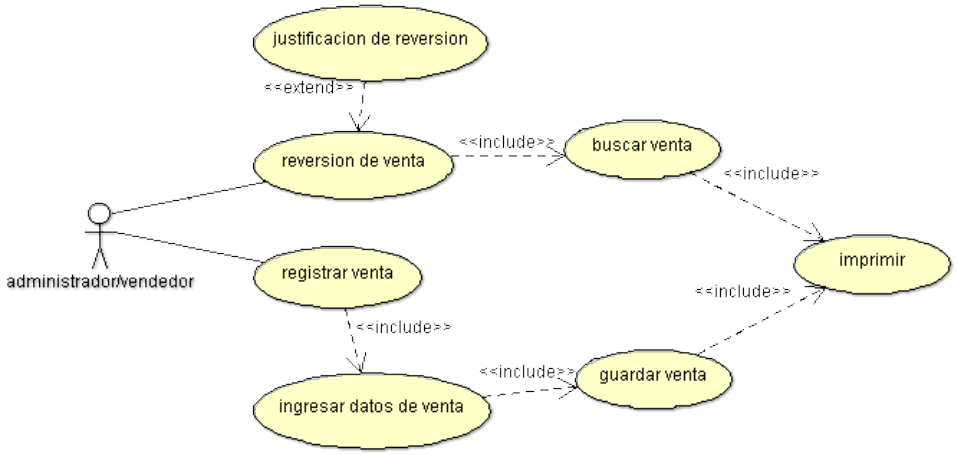


DIAGRAMA DE CASO DE USO REGISTRAR VENTA



ANEXO B DIAGRAMA DE ACTIVIDADES

DIAGRAMA DE ACTIVIDADES GENERAL

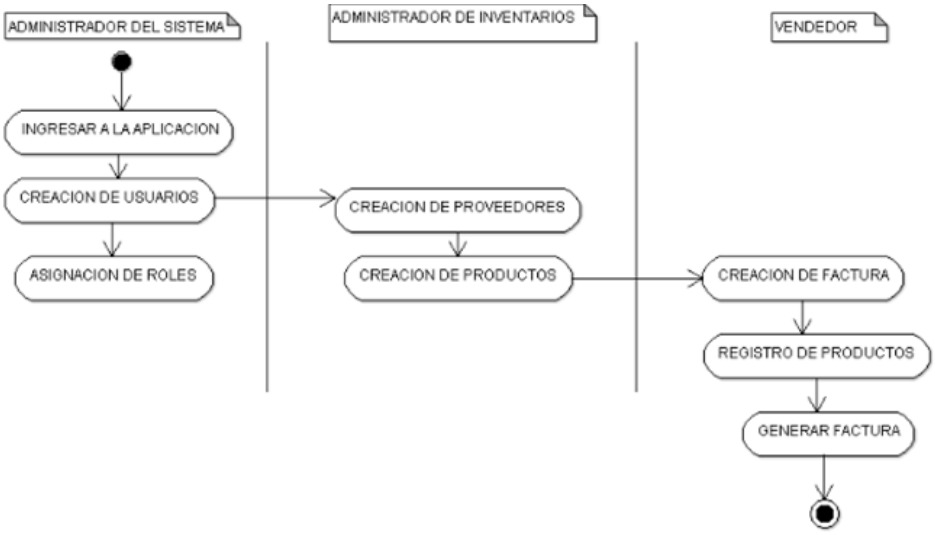


DIAGRAMA DE ACTIVIDADES CREACION DE USUARIOS

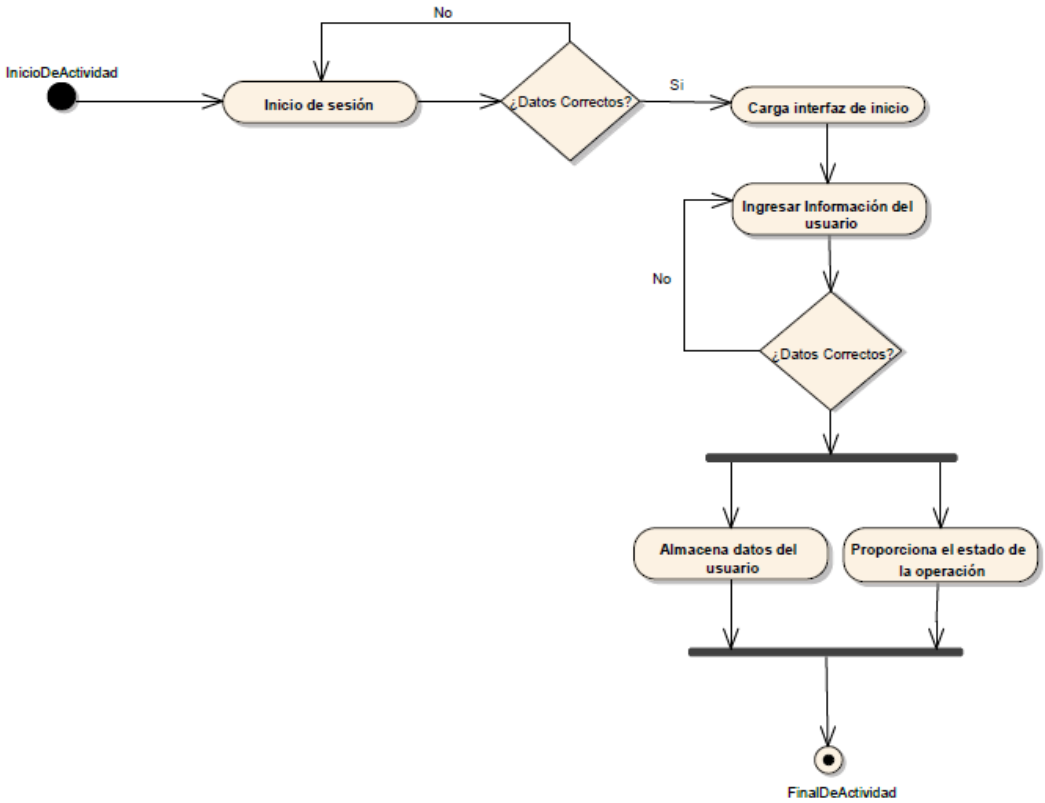


DIAGRAMA DE ACTIVIDADES CONSULTAR USUARIO

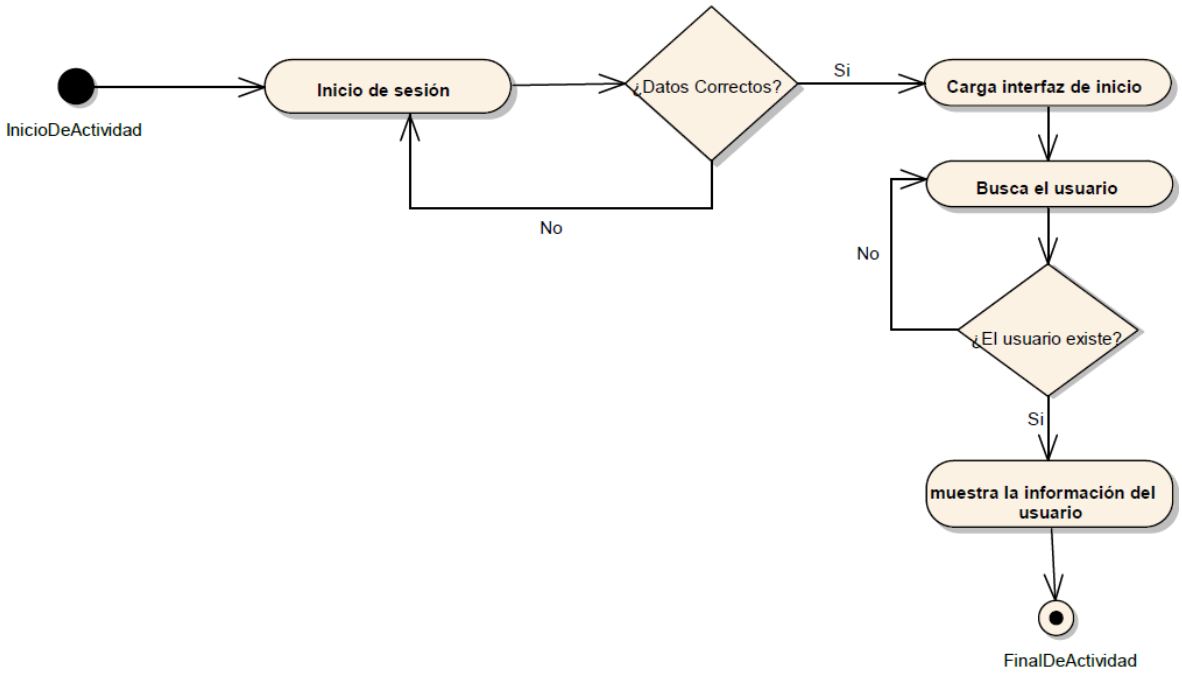


DIAGRAMA DE ACTIVIDADES ELIMINAR USUARIO

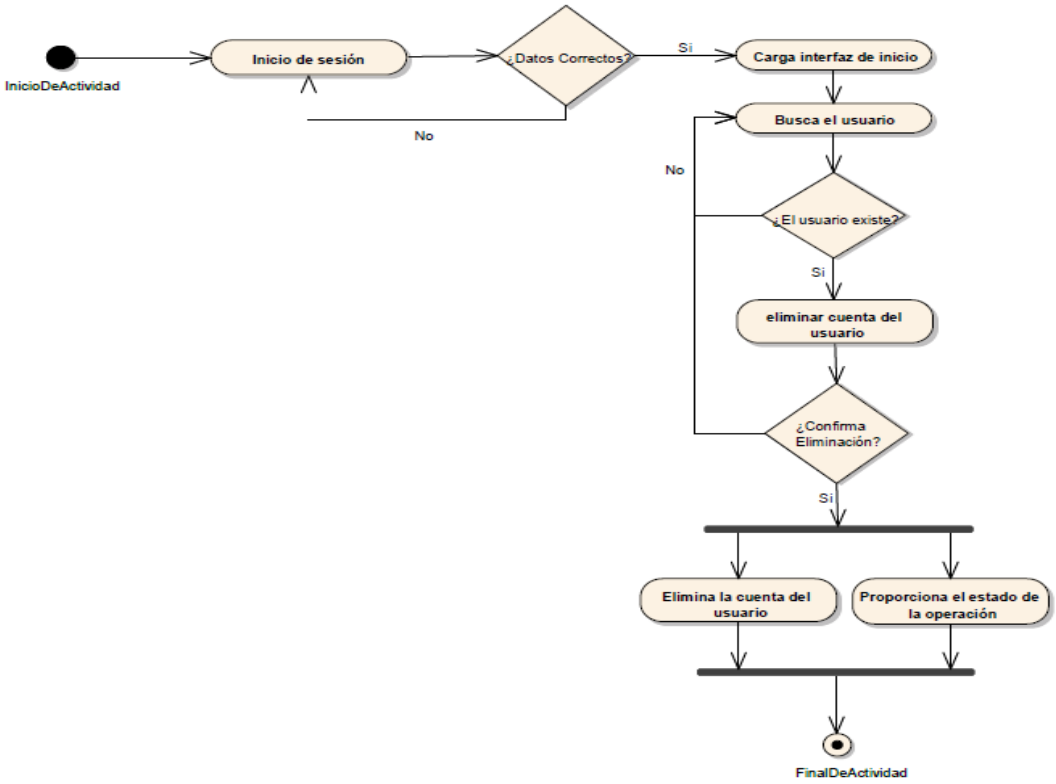


DIAGRAMA DE ACTIVIDADES CREACION DE PROVEEDORES

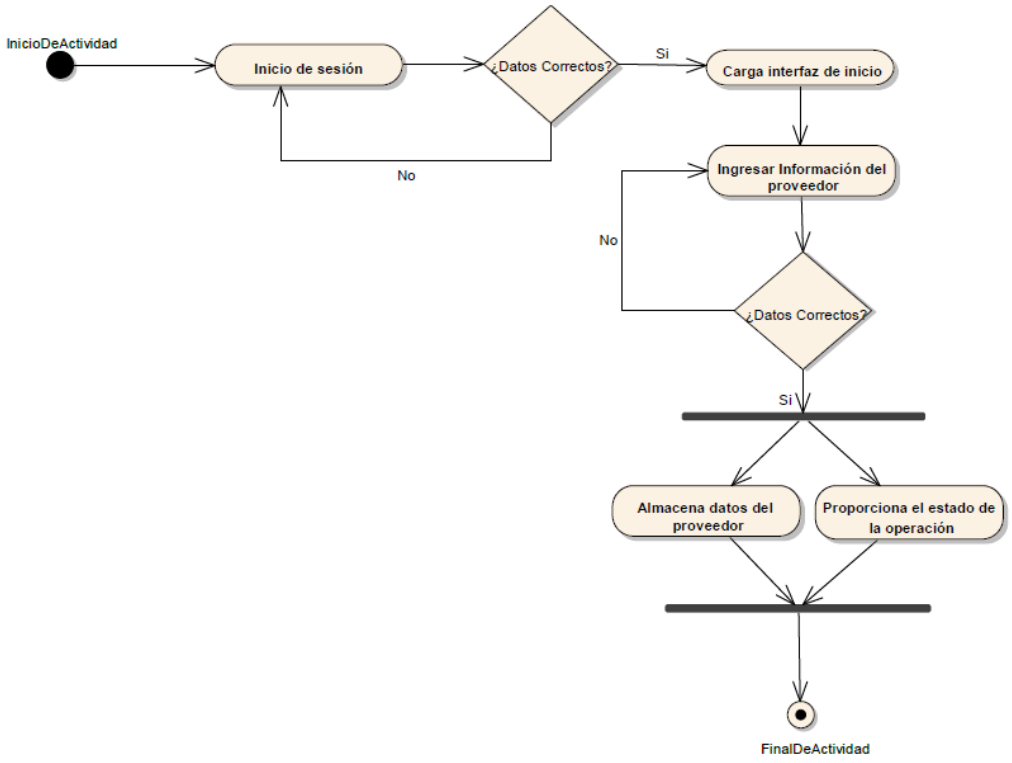


DIAGRAMA DE ACTIVIDADES MODIFICACION DE PROVEEDORES

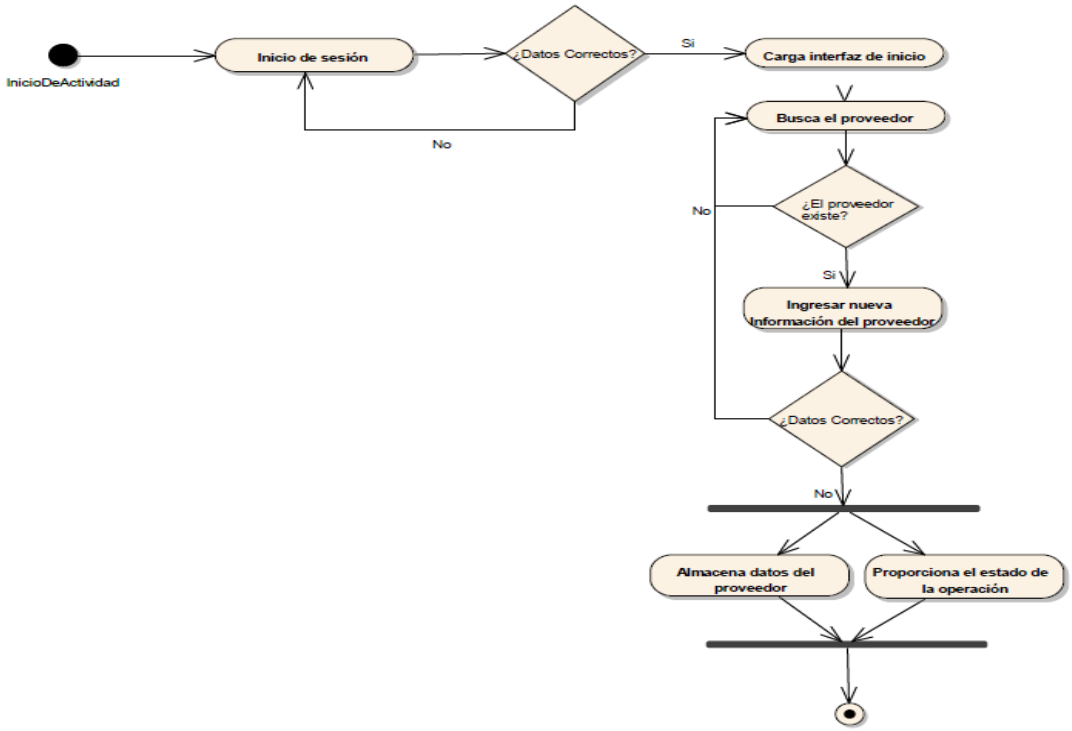


DIAGRAMA DE ACTIVIDADES CREACION DE PRODUCTOS

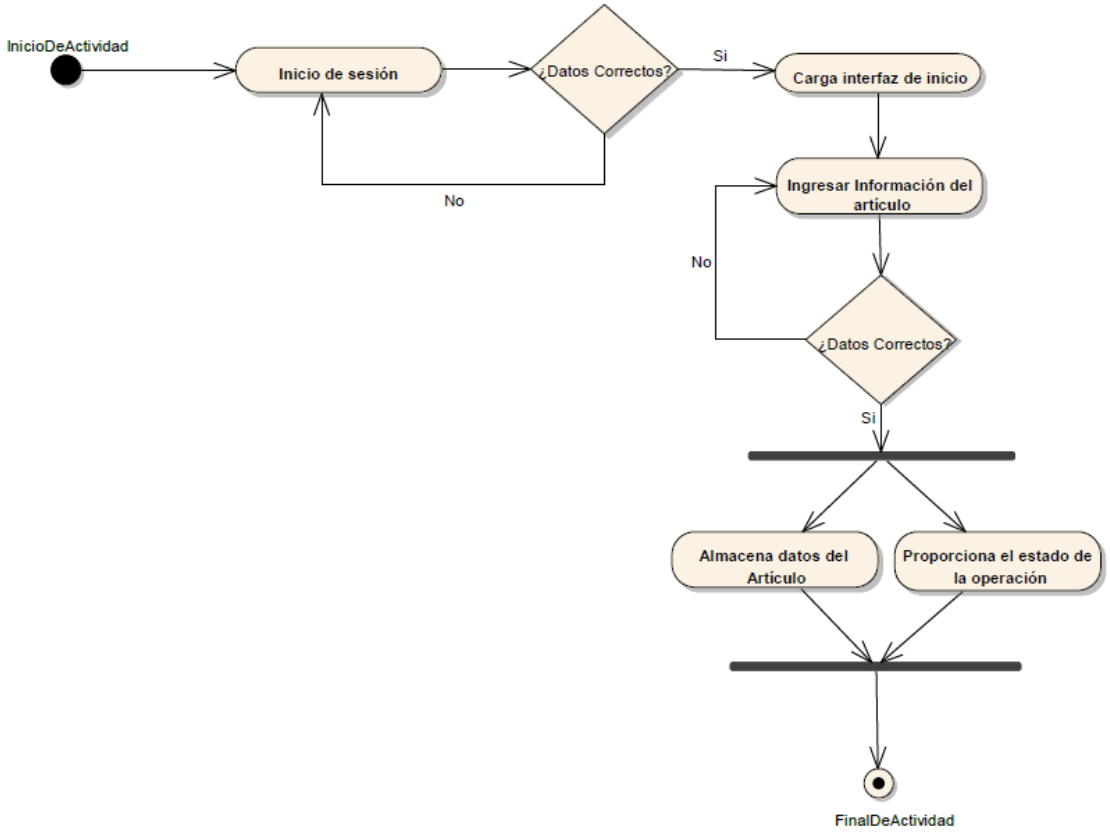


DIAGRAMA DE ACTIVIDADES CONSULTA DE PRODUCTOS

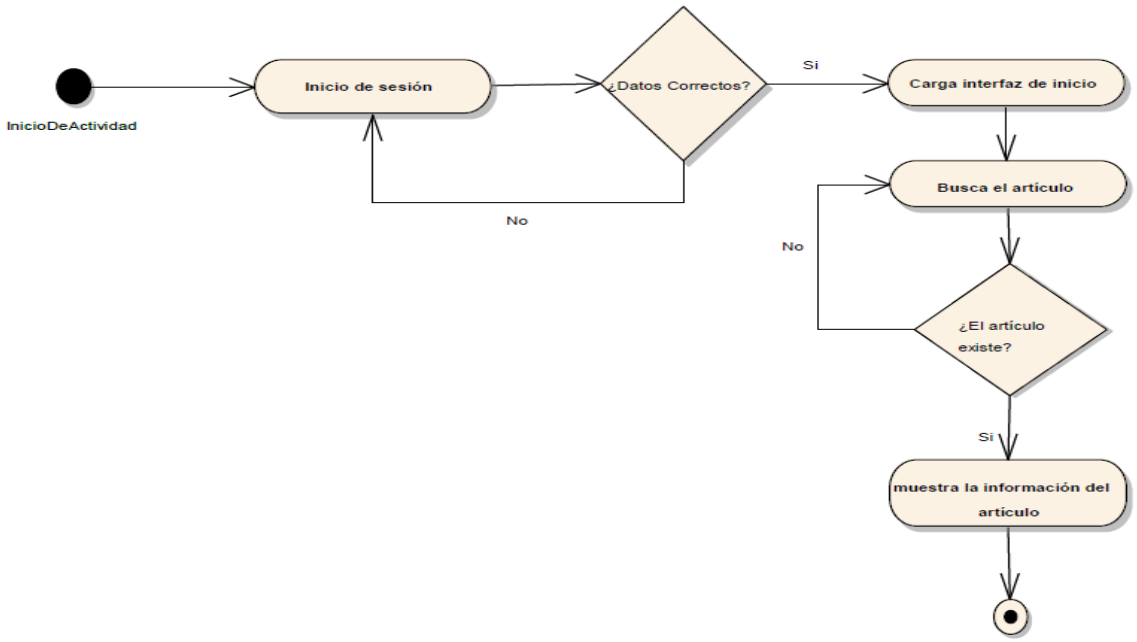
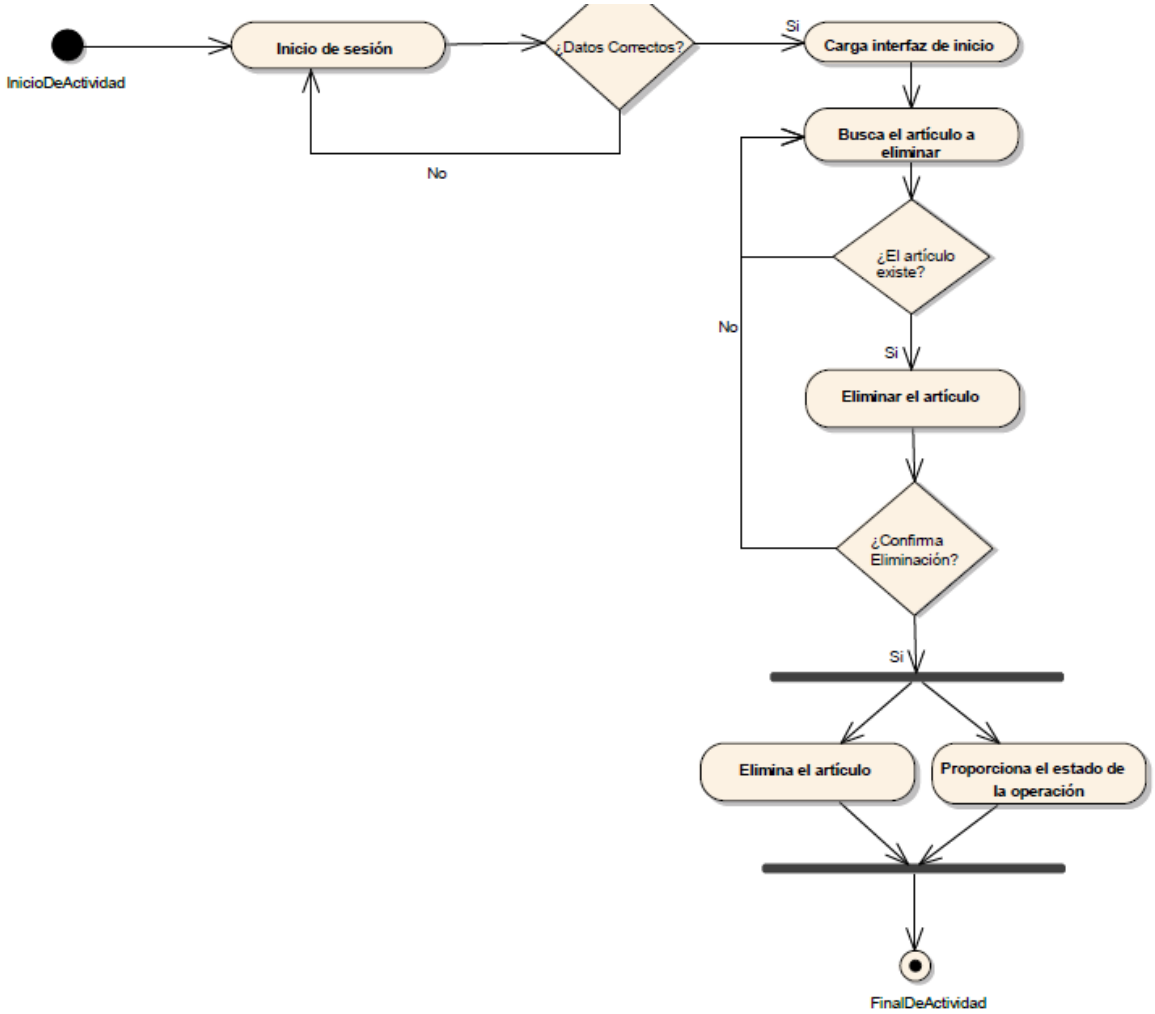


DIAGRAMA DE ACTIVIDADES ELIMINACION DE PRODUCTOS



ANEXO C DIAGRAMAS DE ESTADO

DIAGRAMA DE ESTADOS INICIO DE SESSION

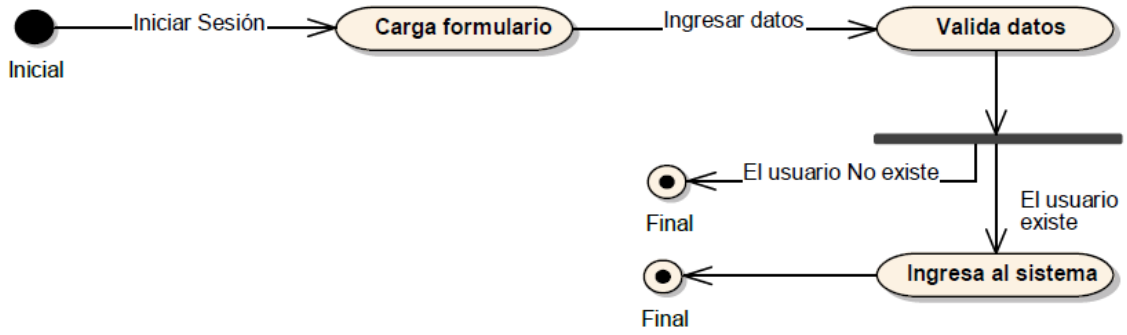


DIAGRAMA DE ESTADO MODIFICAR USUARIO

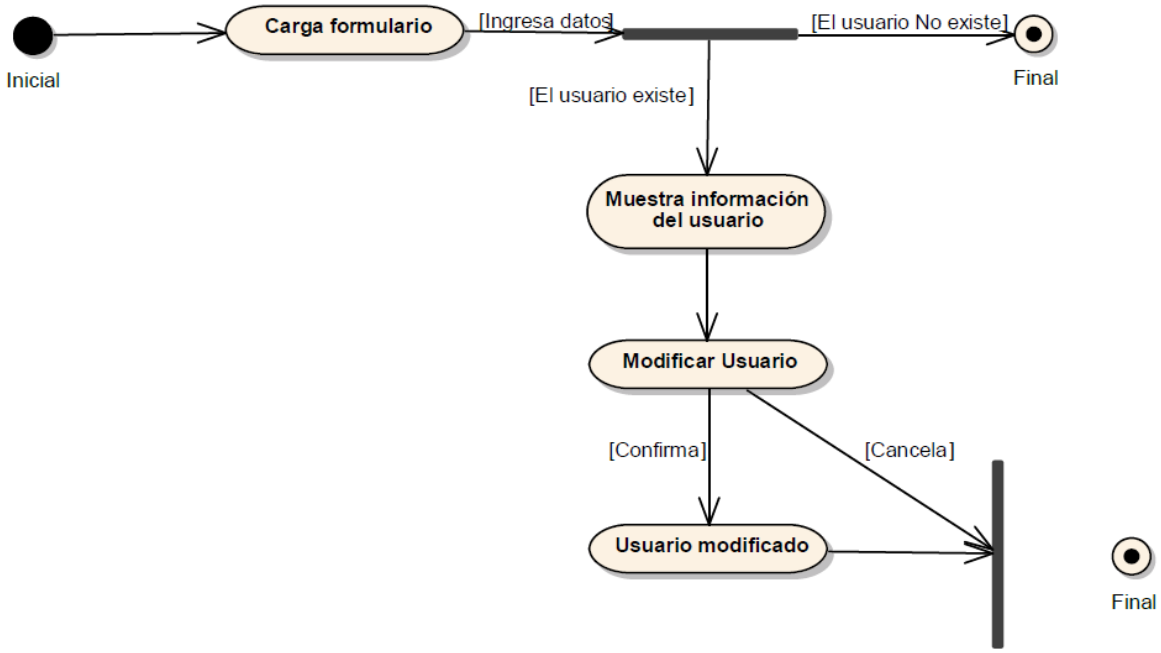
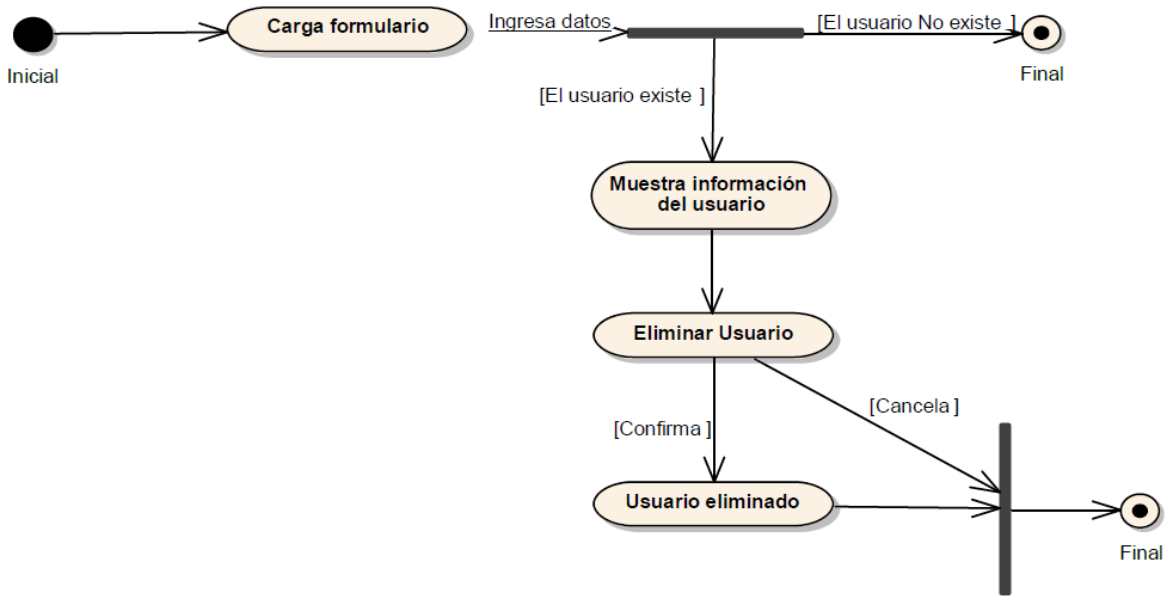



DIAGRAMA DE ESTADO ELIMINAR USUARIO