

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



**SISTEMA DE VENTAS PARA MICRO Y PEQUEÑAS EMPRESAS,
APLICADO A LA EMPRESA SAN SANTIAGO, ACORA – PUNO -**

2017

TESIS

PRESENTADA POR:

OLIVER GOMEZ CUTIPA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2017

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

SISTEMA DE VENTAS PARA MICRO Y PEQUEÑAS EMPRESAS,
APLICADO A LA EMPRESA SAN SANTIAGO, ACORA - PUNO - 2017

TESIS PRESENTADA POR:

OLIVER GOMEZ CUTIPA

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO



APROBADA POR:

PRESIDENTE

:

M.Sc. Ernesto Nayer Tumi Figueroa

PRIMER MIEMBRO

:

M.Sc. Charles Ignacio Mendoza Mollocondo

SEGUNDO MIEMBRO

:

Mg. Oliver Amadeo Vilca Huayta

DIRECTOR / ASESOR

:

D.Sc. Percy Huata Panca

Área

:

Informática

Tema

:

Sistema de Información

Fecha de Sustentacion

:

15/12/2017

DEDICATORIAS

Dedico este proyecto de tesis a Dios y a mis padres y sobrinos. A Dios porque ha estado conmigo en cada paso que doy, cuidándome y dándome fortaleza para continuar diariamente.

A mis padres Marcelino y Graciela quienes fueron pilares fundamentales en mi vida por sus consejos que han sabido guiarme para culminar mi carrera profesional. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos y familia en general. Quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento.

Depositando su entera confianza en cada reto que se me presento sin dudar ni un solo momento en mi inteligencia y capacidad.

A mis queridos hermanos Alex, Fredy y Fatima, por la comprensión y amor incondicional de hermanos que me brindan.

A mis sobrinos Gyohan, Wilkem, Mahite, Camila y Gian Franco quienes fueron pieza fundamental en lo largo de mi carrera por alegrar mis días y motivarme a ser mejor persona.

A ellos este proyecto, que sin ellos, no hubiese podido ser. Es por ello que soy lo que soy ahora. Los amo con mi vida.

OLIVER GOMEZ CUTIPA

AGRADECIMIENTOS

A Dios, por permitir llegar a este momento tan especial de mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más.

A mis padres y hermanos quienes a lo largo de toda la vida me han apoyado y motivado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades.

A la Universidad Nacional del Altiplano de Puno, a la Facultad de Ingeniería Estadística e Informática por haber compartido sus enseñanzas y hacer realidad mi sueño de ser un profesional para el servicio de la sociedad.

A los miembros de mi jurado de tesis, quienes con sus aportes y correcciones han hecho posible la realización de este trabajo de investigación. A mi presidente de jurado M.Sc. Nayer Tummy Figueroa al M.Sc. Charles Ignacio Mendoza Mollocondo al Mg. Oliver Amadeo Vilca Huayta, así como a mi asesor D.Sc. Percy Huata Panca, con sus correcciones y asesoramientos han hecho posible la culminación de esta tesis.

A los docentes de la Escuela Profesional de Ingeniería Estadística e Informática, por todo el conocimiento esparcido durante los años de mi formación académica.

OLIVER GOMEZ CUTIPA

ÍNDICE GENERAL

RESUMEN	10
ABSTRACT	11
CAPÍTULO I INTRODUCCIÓN	12
1.1. Planteamiento del Problema	13
1.2. Formulación del Problema	15
1.3. Hipotesis de la investigación.....	15
1.4. Justificación de la Investigación.....	15
1.5. Objetivos de la Investigación	16
1.5.1. Objetivo General.....	16
1.5.2. Objetivos Específicos	17
CAPÍTULO II REVISIÓN DE LA LITERATURA	18
2.1. Marco teorico	18
2.1.1. Antecedentes de la investigación	18
2.2. Marco Conceptual.....	21
2.2.1. SISTEMA.....	21
2.2.2. Ingeniería de Software	22
2.2.3. Sistemas de Gestión de Base de Datos	25
2.2.4. Mysql	25
2.2.5. Servidor Http a Apache	26
2.2.6. Netbeans	28

2.2.7. NetBeans Enterprise Pack	29
2.2.8. PHP	29
2.2.9. Python	29
2.2.10. Javascript	30
2.2.11. Mysql Workbench	31
2.2.12. Framework	32
2.2.13. PROGRAMACIÓN MODULAR.....	32
2.2.14. Browser, Explorador, Navegador Web	33
2.2.15. Informática.....	34
2.2.16. Seguridad Informática	34
2.2.17. Metodología Extreme Programing (Xp)	34
2.2.18. Ciclo de las Xp.....	35
2.2.19. Fases de la Metodología Xp.....	36
2.2.20. Software Libre	36
2.2.21. Calidad de Software	38
2.2.22. Métricas de Calidad.....	38
2.2.23. La norma ISO/IEC 9126	38
2.2.24. Tecnología Cliente – Servidor	42
2.2.25. Cliente	43
2.2.26. Servidor	43
2.2.27. Características de un Sistema Cliente – Servidor	44
2.3. Definición de Términos Básicos.....	45

2.3.1. Administración	45
2.3.2. Administración de la Cobranza.....	45
2.3.3. Archivo	46
2.3.4. De Usuarios.....	46
2.3.5. Actualizar Estado Productos	46
2.3.6. Aplicaciones Web.....	47
2.3.7. Control Administrativo	47
2.3.8. Cuenta de Usuarios.....	47
2.3.9. Consultar Base de Datos.....	47
2.3.10. Diseño de un Sistema Informático.....	47
2.3.11. El Control como Fase del Proceso Administrativo.....	48
2.3.12. Implementación de Sistema Informático.....	48
2.3.13. Información.....	48
2.3.14. Proceso de Venta.....	48
2.3.15. Pago.....	49
2.3.16. Reporte.....	49
2.3.17. Recibo de Cobranzas	49
2.3.18. Micro y Pequeña Empresa	49
2.4.Operacionalización de Variables	52
CAPÍTULO III MATERIALES Y MÉTODOS	53
3.1.Localización.....	54

3.2. Población	54
3.3. Muestra.....	54
3.4. Diseño de la Muestra	55
3.5. Método de Recolección de Datos	55
3.5.1. Metodología de Desarrollo Xp (Extreme Programing)	55
3.6. Requerimientos del Sistema	58
3.6.1. Requerimientos Funcionales:	58
3.6.2. Requerimientos no Funcionales:	59
3.6.3. REQUERIMIENTOS TÉCNICOS:	59
3.7. Diseño del Sistema	60
3.8. Ciclo de Vida del Desarrollo del Software	61
3.9. Analisis	61
3.10. Diseño	62
3.11. Implementación.....	62
3.12. Implantación.....	62
CAPÍTULO IV RESULTADOS Y DISCUSIÓN.....	63
4.1. Prueba de Hipótesis Para Diferencia de dos Medias Muestrales	63
4.1.1. Planteamiento de Hipótesis.....	63
4.1.2. Fijar el Nivel de Significancia (α)	64
4.1.3. Regla de Decisión Utilizando.....	64
4.1.4. Prueba Estadística	64

4.1.5. DECISIÓN:	65
4.2. Fases de Desarrollo.....	66
4.2.1. Reglas de Negocio	66
4.3. Diagrama Uml Modelamiento del Sistema.....	67
4.3.1. Diagrama de Casos de Uso	67
4.3.2. Modelo de Secuencia	76
CAPÍTULO V CONCLUSIONES.....	78
CAPÍTULO VI RECOMENDACIONES	79
CAPÍTULO VII REFERENCIAS.....	80
ANEXOS	84

ÍNDICE DE FIGURA

Figura N° 1 Esquema de un Sistema	22
Figura N° 2 Esquema de Base de Datos	24
Figura N° 3 Representación Gráfica de la Programación Modular.....	33
Figura N° 4 Ciclo de la Xp.....	35
Figura N° 5 Fases De La Metodología Xp.....	36
Figura N° 6 Arquitectura Fisica Cliente - Servidor.....	45
Figura N° 7 Ubicación De La Micro Y Pequeña Empresa San Santiago.....	54
Figura N° 8 Formula De Prueba De Hipótesis Para Diferencia De Dos Medias Muestrales.....	65
Figura N° 9 Modelo de Casos de Uso.....	69
Figura N° 10 Modelo de Objetos Gestionar Sistema.....	70
Figura N° 11 Modelo de Objetos Controlar Ventas	71
Figura N° 12 Modelo de Objetivo Controlar Almacén.....	72
Figura N° 13 Modelo de Casos de Uso de Requerimiento Gestionar Sistema	73
Figura N° 14 Modelo de Caso de Uso de Requerimiento Controlar Venta	74
Figura N° 15 Modelo de Casos de Uso de Requerimiento Controlar Almacén	75
Figura N° 16 Modelo de Secuencia: Controlar Producto.....	76
Figura N° 17 Modelo de Secuencia: Generar Reporte Almacen	77

ÍNDICE DE TABLA

Tabla N° 1 Operacionalizacion De Variables	52
Tabla N° 2 Ficha de Evaluación de la Calidad del Producto Estándar	57
Tabla N° 3 Escala Valorativa (Escala De Likert)	58
Tabla N° 4 Cuadro De Decisiones Iso 9126.....	58
Tabla N° 5 Encuesta del Antes y Despues del Sistema.....	64

RESUMEN

El presente trabajo se realizó en el centro poblado de Caritamaya del distrito de Acora, departamento de Puno, la micro y pequeña empresa “San Santiago” no cuenta con un sistema de ventas, ya que el control administrativo se realiza en cuadernos de apunte. el objetivo principal es desarrollar un sistema de ventas, debido a que los municipios distritales están priorizando la creación de microempresas y al mismo tiempo capacitando en la creación de microempresa, por eso es necesario contar con sistemas de ventas. La metodología que se utilizó para el desarrollo del sistema de ventas es metodología de desarrollo XP (extreme programming), La programación extrema es una metodología muy rápida y ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Los resultados del trabajo en relación a la validación del software, se concluye que el sistema de ventas de la micro y pequeña empresa, cumple con los requisitos planteados con un media de 114 Puntos según la escala de calificación del ISO – 9126, con un $T_c = 13.02697303$ rechazando la H_0 y se acepta la H_1 , demostrando así que si existe diferencia entre el antes y el después de haber implantado el sistema de ventas. con un 100% que mejoro el proceso de ventas, control de stock y cubre sus expectativas. El sistema que se implanto en la micro y pequeña empresa San Santiago ofrece una importante y notable satisfacción en el usuario que lo opera.

Palabras clave: Extreme programming, java netbeans, micro y pequeña empresa, sistema de ventas, venta de quesos.

ABSTRACT

The present work was carried out in the town of Caritamaya in the district of Acora, department of Puno, the micro and small business "San Santiago" does not have a sales system, since the administrative control is done in note books. The main objective is to develop a sales system, because the district municipalities are prioritizing the creation of microenterprises and at the same time training in the creation of microenterprises, that is why it is necessary to have sales systems. The methodology that was used for the development of the sales system is XP (extreme program) development methodology, Extreme programming is a very fast and light methodology of software development that is based on simplicity, communication and feedback or reuse of the developed code. The results of the work in relation to software validation, it is concluded that the sales system of the micro and small business, meets the requirements raised with an average of 114 points according to the rating scale of ISO - 9126, with a $T_c = 13.02697303$ rejecting the H_0 and the H_1 is accepted, demonstrating that there is a difference between the before and after having implemented the sales system. with a 100% that improved the sales process, stock control and covers your expectations. The system that was implanted in the micro and small company San Santiago offers an important and notable satisfaction in the user that operates it.

Keywords: Extreme programming, java netbeans, micro and small business, sales system, sale of cheeses.

CAPÍTULO I

INTRODUCCIÓN

Hoy en día las empresas se esfuerzan para tener una mayor participación en el mercado, lo que ha originado el desarrollo de aplicaciones donde se puede registrar el crecimiento de las empresas y las preferencias de los consumidores, estas aplicaciones mejoran el control administrativos mediante un seguimiento preciso de todas las transacciones que se realicen dentro de un negocio en tiempo real proporcionando reportes detallados de ventas que permiten a los administradores ordenar fácilmente la cantidad correcta de productos en el momento adecuado, esto permite al negocio o a las empresas mejorar el servicio al cliente reduciendo el tiempo necesario para terminar una transacción.

Los sistemas de ventas tiene la capacidad de ser personalizados para cumplir con las necesidades específicas de un negocio por ejemplos las organizaciones de ventas al menudeo pueden localizar rápidamente los precios de ventas y costos actuales de todos los productos, y los hoteles pueden vincular fácilmente las cuentas del restaurante con los cargos de una habitación, una

tienda de ropa pueden incluir todos los elementos que ofrece y organizarlos ya sea por marca o por precio.

En años recientes el mercado ha incrementado y se han generado diversidad de negocios en todos los rubros, prácticamente obligando a las empresas a ofrecer una mejor atención no solo limitándose a la venta del producto sino también a considerar mayor comodidad y mejor atención al cliente, esto depende mucho del sistema con el que cuente la empresa, un sistema rápido y sencillo es lo que se busca en la actualidad.

1.1. Planteamiento del Problema

Los sistemas informáticos no estaban incluidos dentro de los temas prioritarios en la agenda del empresario o comerciante y mediano de nuestro país. Solo las grandes empresas tenían el privilegio de informatizar sus circuitos administrativos ya que en ese entonces los costos de desarrollo e implementación de sistemas requerían de inversiones importantes. Esto redundaba en mejoras sustanciales en los procesos con el lógico impacto en los resultados finales de la gestión. Por lo tanto las mayores posibilidades de acceder a nuevas tecnologías cada vez son más competitivas en comparación con el resto.

Hoy, luego de más de veinte años, el panorama ha cambiado. Es mucho más accesible para el pequeño empresario contar con la tecnología que permita reducir la brecha competitiva que lo separa de la gran empresa.

Informatizar la administración de una empresa significa contar con herramientas que nos permitan gestionar inventarios, cuentas corrientes de

clientes, emitir reportes y listados para la liquidación de impuestos y otros. Por tanto nuestro caso en la micro y pequeña empresa San Santiago la Señora Sonia presidenta de la empresa, lleva a todas las personas que compran en forma manual y solo le basta con algunas anotaciones las ventas y el stock de los quesos lo tienen en hojas de apuntes.

Pero también es cierto que cuando necesita tener un reporte de las ventas en el día, el propietario demora en obtener las cuentas haciendo uso de la calculadora, que por cierto no eran exactos, además a medida que la empresa crece ya no es posible almacenar todo, para las diferentes actividades más importantes como: registrar a los clientes, vender, hacer facturas y boletas, registrar productos, el empresario seguramente contara con una base mucho más sólida de información, por lo que es importante la reducción de trabajo operativo que es necesario para obtener la información.

La micro y pequeña empresa San Santiago es una empresa dedicada a la venta de quesos de primera calidad, la misma que comercializa productos a nivel local, regional y nacional, buscando alcanzar la plena aceptación y fidelidad de sus clientes en la región de Puno. Para ello, la empresa necesita llevar un estricto control del stock de los quesos, registro de los clientes, sus reportes.

Actualmente, la micro y pequeña empresa San Santiago, no cuenta con un sistema de ventas, el mismo que por ser limitado y deficiente, no permite concluir satisfactoriamente los procesos, es por ello que la mayoría de estos se vienen realizando en forma manual, el cual implica disponer de

mayor tiempo de llevarlo a cabo. Otro punto a tener en cuenta es que no hay un control estricto de inventario, provocando pérdidas en la empresa.

Todo lo mencionado conlleva a una deficiente gestión de información comercial, disminuyendo su competitividad. Por tanto contar con un sistema informático de ventas para mejorar el proceso de ventas en la micro y pequeña empresa San Santiago significa reducir las tareas de ingreso y registro de las operaciones al mínimo necesario, reducir errores, eliminar la duplicación de tareas, generar listado de clientes y aumentar los controles.

1.2. Formulación del Problema

¿Cómo un sistema de ventas de micro y pequeñas empresas mejorara el proceso de ventas en la empresa San Santiago en la comunidad de caritamaya del distrito de Acora, departamento de Puno, 2017?

1.3. Hipotesis de la investigación

El diseño e implementación de un sistema informático mejorara el proceso de ventas en la micro y pequeña empresa “San Santiago” Del centro poblado de caritamaya, del distrito de Acora, departamento de Puno, 2017

1.4. Justificación de la Investigación

Con el fin de mejorar la productividad y el rendimiento de la micro y pequeña empresa “San Santiago”, es fundamental implementar un sistema de ventas que brinden eficiencia de la gestión de ventas. La creación de

este sistema de ventas para la micro y pequeña empresa “San Santiago” brindo la posibilidad de obtener grandes ventajas, incrementar la capacidad de organización de la empresa, y tomar de esta manera los procesos a una verdadera competitividad, mejorando los reportes de ventas, el control de los productos. Por todo lo descrito la micro y pequeña empresa “San Santiago” convertirá su emprendimiento en una verdadera empresa competitiva insertada en el mercado actual.

El sistema que se creó en la micro y pequeña empresa “San Santiago” ofreciendo una importante y notable satisfacción en los usuarios que lo operaran, debido a su facilidad de uso y su acceso constante y los empleados logrando alcanzar los objetivos planteados por la empresa, por ello, es importante destacar que la creación del sistema de ventas en la micro y pequeña empresa “San Santiago” implica un cambio organizativo, ya que no solo afecta a la administración de la empresa, sino también a sus empleados.

1.5. Objetivos de la Investigación

4.1.1. Objetivo General

Desarrollar un sistema de ventas para micro y pequeñas empresas, aplicado a la empresa San Santiago, acora-puno – 2017.

4.1.2. Objetivos Específicos

- ✓ Diseñar e Implementar un sistema informático de ventas en la micro y pequeña empresa “San Santiago” del centro poblado de Caritamaya, distrito de Acora, departamento de Puno.

- ✓ Comparar los procesos de ventas luego de implementar el sistema informático.

CAPÍTULO II

REVISIÓN DE LA LITERATURA

El desarrollo de un sistema informatizado optimiza el tiempo, mecanizar la labor del usuario, especialmente desarrollado para un análisis rápido y eficaz.

2.1. Marco teorico

4.1.3. Antecedentes de la investigación

Chura (2015) Concluye determinando que los procesos principales de ventas que se realizaron en la tienda Minimarket Jose Carlos de la ciudad de Azángaro tales como son abarrotes con el 50%, los enlatados con un 25% y con 12.5% entre lácteos, frutas y verduras.

Llacchua (2007) Concluye el diseño modular que tiene el sistema facilita la administración y el entendimiento del mismo haciendo más la integración de otros módulos o componentes para su crecimiento con ello también cabe recalcar que el diseño multiplataforma hace que se integre fácilmente a cualquier plataforma de hardware y software. El uso de metodología de desarrollo del sistema sea entendible, sostenible, incremental. Más aun, el uso de un entorno de desarrollo rápido como es netbeans para Java, ha hecho que nuestro desarrollo sea concluido en el tiempo previsto.

Vilema (2007) Llega a las siguientes conclusiones: el diagnóstico y levantamiento de información, como primero, pasos para el desarrollo de sistema, se constituye en elemento crítico para el éxito de proyecto de software, pues allí donde se establecen los problemas actuales carencias en el desarrollo del proceso. Es por ello que debe centrarse gran esfuerzo y tiempo a su realización. El uso de entrevistas personales y/o cuestionarios a los usuarios de la empresa es vital importancia. Pues termina establecer sus necesidades de información e involucrarlos en el desarrollo del proyecto desde el principio.

Vásquez (2008) Reporta que automatizar el proceso de centralización de datos reduce los gastos administrativos y permite obtener información más confiable y oportuna, permitiendo que la toma de decisiones sea más fluida. Los sistemas de información distribuidos reducen la redundancia de tareas durante el control del proceso de comercialización estudiado. El análisis y diseño de diseño mediante la aplicación de las disciplinas del

RUP y UML permite obtener modelos muy detallados y completos de los sistemas que se desarrollan en el mundo real. Aunque implica un esfuerzo mayor que otras metodologías, el RUP mediante la iteración permite un control mayor del ciclo del desarrollo del software.

Romanelli (2009) En el trabajo de investigación titulado “diseño de un sistema de información para la gerencia de ventas de una empresa de mantenimiento y suministro de equipos analíticos de laboratorio, ubicada en puerto Ordaz, Estado de Bolívar”, el objetivo general fue diseñar un sistema de información para la gerencia de ventas de una empresa de mantenimiento y suministro de equipos analíticos de laboratorio, ubicada en puerto Ordaz, Estado de Bolívar, llegando a la conclusión que la Gerencia de ventas maneja un masivo número de información de forma manual lo cual deben ser guardadas y modificadas en ciertas ocasiones, lo que se convierte en un proceso tedioso. Por esta razón se propuso el Diseño del Sistema de Información.

Orozco (2014) Concluye que el sistema desarrollado se constituye en una gran ayuda para la mejor administración des establecimiento comercial sobre todo en aspectos relacionados con el manejo de los inventarios registrados, clasificados y accediendo a la información de una manera ágil, rápida y veraz de una actividad comercial.

Neil (2016) Se logró implementar re reportes con el cual el uso del sistema de matrícula web permite una mayor viabilidad, usabilidad y seguridad en la información, así como el uso de los recursos (hardware.

Software, trabajadores) en el centro educativo. De igual manera, se optimiza la toma de decisiones ya que cuenta con reportes personalizados.

Cuzco, Lozano (2012) El desarrollo del proyecto de tesis y en base al grado de satisfacción del personal de la 1. Municipalidad de cantón Déleg. Podemos decir que se resolvieron las deficiencias con las que contaba la municipalidad en el proceso de cobro e impuesto, evitando la información redundante y consiguiendo que el sistema sobresalga en eficiencia, efectividad y facilidad de uso.

Reátegui (2014) Concluye que si se mantuviese el control de ventas manualmente la situación de los procedimientos serian inadecuados reflejando inconvenientes en el control de ventas, tanto así que los resultados arrojados al implementar un sistema de información web resultan ser de carácter favorable, se automatizara los procesos de tal forma que se empleara la utilización de menos tiempo y menos recursos para obtener mejora en control de ventas

2.2. Marco Conceptual

4.1.4. SISTEMA

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materiales del ambiente y proveen (salida) información, energía o materia.

Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software).

Cada sistema existe dentro de otro más grande, por lo tanto un sistema puede estar formado por subsistemas y partes, y a la vez pueden ser parte de un súper sistema.

Los sistemas tienen límites o fronteras, que los diferencian del ambiente. Este límite puede ser físico (el gabinete de una computadora) o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, el sistema es abierto, de lo contrario, el sistema es cerrado.

Figura N° 1 Esquema de un Sistema



FUENTE: blogspot.pe

4.1.5. Ingeniería de Software

Es la ciencia que ayuda a elaborar los sistemas con el fin de que sea económico, fiable y funcione eficientemente sobre las maquinas reales. La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de 3 elementos clave: métodos, herramientas y procedimientos, estos facilitan al gestor a controlar el proceso de desarrollo

de software y suministra a los que practican dicha ingeniería las bases para construir software de alta calidad. (Kendal, 1997).

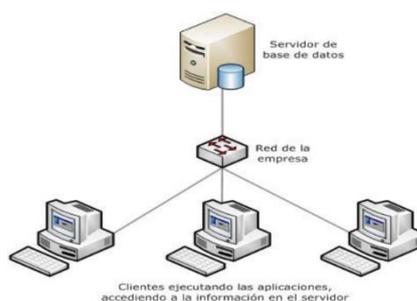
Hardware: los dispositivos electrónicos que proporcionan la capacidad de computación y que proporcionan las funciones del mundo exterior. (Manual avanzado de computación, 2006).

Gente: los individuos que son usuarios y operadores del software y hardware.

Base de Datos: una colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema. (Presman, 2006).

Entre las principales características de los sistemas de base de datos podemos mencionar:

- ✓ Independencia lógica y física de los datos.
- ✓ Redundancia mínima.
- ✓ Acceso concurrente por parte de múltiples usuarios.
- ✓ Integridad de los datos.
- ✓ Consultas complejas optimizadas.
- ✓ Seguridad de acceso y auditoria.
- ✓ Respaldo y recuperación.
- ✓ Acceso a través de lenguajes de programación estándar.

Figura N° 2 Esquema de Base de Datos**FUENTE: es.slideshare.net**

Ventajas de las bases de datos

- ✓ Control sobre la redundancia de datos.
- ✓ Consistencia de datos.
- ✓ Compartición de datos.
- ✓ Mantenimiento de estándares.
- ✓ Mejora en la integridad de datos mejora en la seguridad.
- ✓ Mejora en la accesibilidad a los datos.
- ✓ Mejora en la productividad.
- ✓ Mejora en el mantenimiento.
- ✓ Aumento de la concurrencia.
- ✓ Mejora en los servicios de copias de seguridad.

Tipos de campos

Cada sistema de base de datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- ✓ **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- ✓ **Booleanos:** poseen dos estados: verdadero “Si” y Falso “No”.
- ✓ **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- ✓ **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- ✓ **Alfanumérico:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- ✓ **Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: servir de identificador ya que resultan exclusivos de un registro.

4.1.6. Sistemas de Gestión de Base de Datos

Los sistemas de gestión de base de datos (en inglés Data Base Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta

4.1.7. Mysql

Es un sistema de administración de base de datos relacional (RDBMS) se trata de un programa capaz de almacenar una enorme

cantidad de datos de gran variedad y de distribución para cubrir la necesidad de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos, MySQL compite con sistemas RDBMS propietarios conocidos como Oracle, SQL Server y DB2.

MySQL, es un desarrollo de código abierto, lo que significa que el código fuente está disponible gratuitamente para todo el mundo que lo quiera. Como el código fuente de MySQL está disponible, hay formas disponibles de instalar MySQL. (Castaño, 2005).

4.1.8. Servidor Http a Apache

El servidor HTTP Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular software NCSA, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y “civilizasen” el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor “parcheado”). El servidor Apache se desarrolla dentro del proyecto HTTP Server (http) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, base de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzo su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan solo pueden ser aprovechadas por usuarios locales y no remotamente.

Sin embargo, algunas se pueden accionar remotamente en ciertas disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones LAMP. Junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby). Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM Web Sphere Aplicación Server.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su

escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos. Los programadores de aplicaciones web a veces utilizan una versión local de Apache en orden de pre visualizar y probar código mientras este es desarrollado.

La Free Software Foundation no considera a la licencia Apache como compatible con la versión 2 de la GNU General Public License (GPL), en la cual el software distribuido bajo la GPL sin embargo, la versión 3 de la GPL incluye una provisión (Sección 7e) que le permite ser compatible con licencias que tienen cláusulas de represalia de patentes, incluyendo a la Licencia Apache. El nombre Apache es una marca registrada y puede ser utilizada con el permiso expreso del dueño de la marca (Allison, 2011).

4.1.9. Netbeans

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- ✓ Administración de las interfaces de usuario (ej. menús y barras de herramientas).

- ✓ Administración de las configuraciones del usuario.
- ✓ Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- ✓ Administración de ventanas.
- ✓ Framework basado en asistentes (diálogos paso a paso).

Otra característica son los paquetes que componen a NetBeans (algunos pueden ser descargados individualmente).

4.1.10. NetBeans Enterprise Pack

Provee Soporte para la creación de aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para web services.

4.1.11. PHP

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE.

4.1.12. Python

NetBeans permite crear aplicaciones con Python ya que posee un motor para escribir (resaltando la sintaxis), identificar errores y el debugger. Sin duda alguna, netbeans se ha convertido en un **IDE** apto para la mayoría

de los lenguajes de programación opensource modernos. También se estima que dará para soporte GUI para varias librerías gráficas como son PyQt y GTK. Se espera que salga una versión con motor para soporte para Jython, con acceso a todos las librerías de java e incluyendo soporte para Swing y también para las librerías gráficas de python que ya mencionamos.

Ventajas

- ✓ La plataforma Netbeans puede ser usada para desarrollar cualquier tipo de aplicación.
- ✓ Reutilización del Módulos.
- ✓ Permite el uso de la herramienta Update Center Module.
- ✓ Instalación y actualización simple.
- ✓ Incluye Templates y Wizards.
- ✓ Posee soporte para Php.

Desventajas

- ✓ Poca existencia de pluguins para esta plataforma.
- ✓ hace falta documentación del Rich Client Plataform (RCP).
- ✓ No posee un editor de código HTML.

4.1.13. Javascript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa SunMicrosystems.

4.1.14. Mysql Workbench

MySQL Workbench es un software creado por la empresa Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL.

Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para exportare e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador.

MySQL Workbench puede generar también el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x

4.1.15. Framework

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrirnos. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

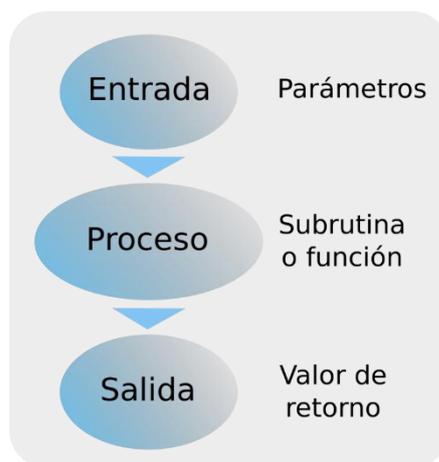
4.1.16. PROGRAMACIÓN MODULAR

Uno de los métodos más conocidos para resolver un problema es dividirlo en problemas más pequeños, llamados subproblemas. De esta manera, en lugar de resolver una tarea compleja y tediosa, resolvemos otras más sencillas y a partir de ellas llegamos a la solución. Esta técnica se usa mucho en programación ya que programar no es más que resolver problemas, y se le suele llamar diseño descendente, metodología del divide y vencerás o programación top-Down.

Es evidente que si esta metodología nos lleva a tratar con subproblemas, entonces también tengamos la necesidad de poder crear y

trabajar con subprogramas para resolverlos. A estos subprogramas se les suele llamar módulos, de ahí viene el nombre de programación modular. En Pascal disponemos de dos tipos de módulos: los procedimientos y las funciones. (Prieto, Ruiz & Cantero, 1989).

Figura N° 3 Representación Gráfica de la Programación Modular



FUENTE: <http://slideplayer.es>

4.1.17. Browser, Explorador, Navegador Web

Aplicación que sirve para acceder a la WWW (todas las páginas web) y “navegar” por ella a través de los enlaces. Generalmente estos programas no solo traen la utilidad de navegar por la WWW, sino que se pueden también administrar correo, grupos de noticias, ingresar al servicio de FTP.4

Actualmente los navegadores más populares son:

- ✓ Google Chrome.
- ✓ Internet Explorer.
- ✓ Mozilla Firefox.

4.1.18. Informática

Se entiende como el resultado de los términos información y automatización. Trata de la concepción. Realización y utilización de los sistemas para procesamiento de información “INFORMÁTICA es la ciencia que estudia el tratamiento automático y racional de la información.” Se dice que el tratamiento es automático por ser maquinas las que realizan los trabajos de captura, proceso y presentación de la información, y se habla de racional por estar todo el proceso definido a través de programas que siguen el razonamiento humano.

4.1.19. Seguridad Informática

La seguridad informática es una disciplina que se relaciona a diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información de un sistema informático y sus usuarios.

Técnicamente es imposible lograr un sistema informático ciento por ciento seguros, pero buenas medidas de seguridad evitan daños y problemas que pueden ocasionar intrusos.

4.1.20. Metodología Extreme Programing (Xp)

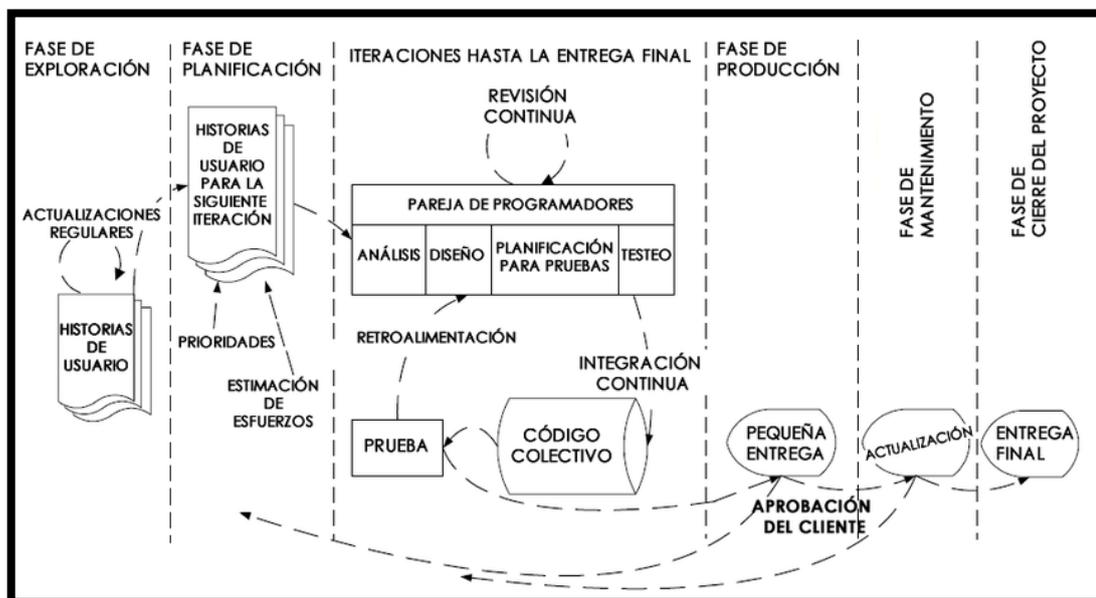
- ✓ Metodología para un ágil desarrollo de software.
- ✓ Programación basada en los deseos del cliente.
- ✓ El equipo lo conforman los jefes de proyecto, desarrolladores y el cliente.
- ✓ Se rige por valores y principios.

Valores de xp

- ✓ **Comunicación:** Crear software requiere de sistemas comunicados.
- ✓ **Simplicidad:** Empezar con lo necesario y requerido y trabajar desde ahí.
- ✓ **Retroalimentación:** Del sistema, del cliente, y del equipo.
- ✓ **Valentía:** Programa para hoy y no para mañana.
- ✓ **Respeto:** El equipo debe trabajar como uno, sin hacer decisiones repentinas.
- ✓ **Codificación:** La parte más importante de XP.
- ✓ **Pruebas:** Nunca se puede estar seguro de algo hasta haberlo probado.
- ✓ **Escuchar:** Escuchar los requisitos del cliente acerca del sistema a crear.
- ✓ **Diseño:** Crear una estructura del diseño para evitar problemas.

4.1.21. Ciclo de las Xp

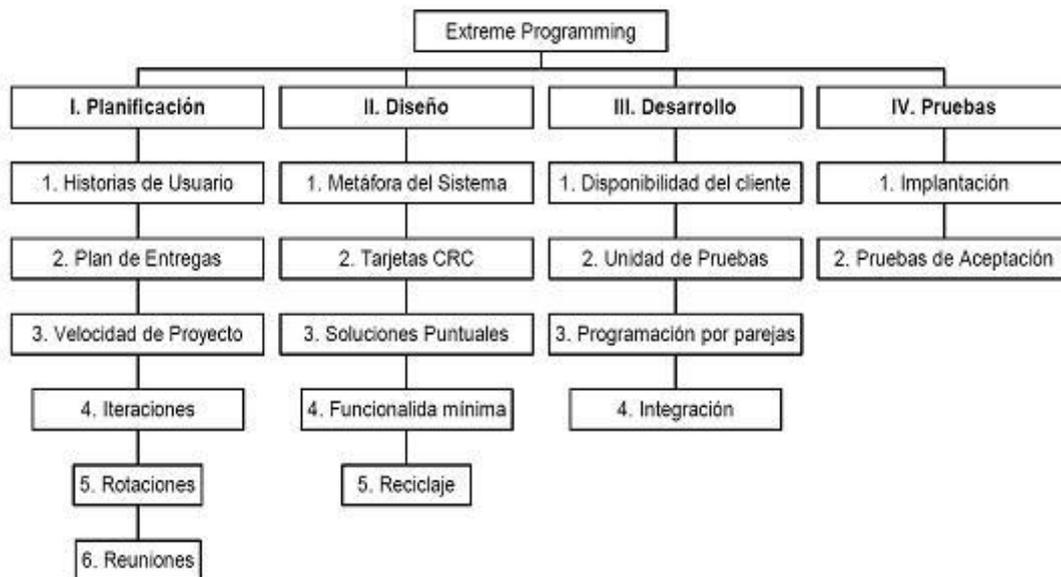
Figura N° 4 Ciclo de la Xp



FUENTE: researchgate.net

4.1.22. Fases de la Metodología Xp

Figura N° 5 Fases De La Metodología Xp



FUENTE: es.slideshare.net

4.1.23. Software Libre

Es el software que una vez adquirido puede ser utilizado libremente por los usuarios; copiarlo, estudiarlo, modificarlo y redistribuirlo. El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios, (no obligatorio) por ende no hay que asociar software libre a software gratuito ya que conservando su carácter de libre puede ser distribuido comercialmente. (Stallman, 2004).

Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- ✓ La libertad de usar el programa, con cualquier propósito (libertad 0).

- ✓ La libertad de estudiar cómo funciona el programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- ✓ La libertad de distribuir copias, con lo que se puede ayudar a las personas más cercanas (libertad 2).
- ✓ La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, se tiene la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no se tiene que pedir o pagar permisos.

Ventajas del software libre en general:

- ✓ Ahorros multimillonarios en la adquisición de licencias.
- ✓ Combate efectivo a la copia ilícita de software.
- ✓ Eliminación de barreras presupuestales.
- ✓ Beneficio social y tecnológico.
- ✓ Los tiempos de desarrollo sobre algo que no exista son menores por la amplia disponibilidad de herramientas y librerías.
- ✓ Las aplicaciones son fácilmente auditadas antes de ser usadas en procesos de misión crítica.

- ✓ Todo el mundo tiene derecho de usarlo sin costo alguno, de acceder a su diseño y aprender de él, y de modificarlo.
- ✓ Es de libre distribución (cualquier persona puede regalarlo, venderlo o restarlo).
- ✓ Tiende a ser muy robusto, debido a que muchas personas tienden a contribuir en su mejora siendo de interés propio.
- ✓ Tiende a ser muy diverso, debido a que las personas que contribuyen tienen necesidades diferentes y esto hace que el software este adaptado a una cantidad más grande de problemas.

4.1.24. Calidad de Software

La calidad del software es una preocupación a la que se dedican muchos esfuerzos, sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios. (Mendoza, 2002).

4.1.25. Métricas de Calidad

Principal objetivo de los ingenieros de software es producir sistemas, aplicaciones o productos de alta calidad.

4.1.26. La norma ISO/IEC 9126

El estándar ISO 9126, ahora englobado en el proyecto SQuaRE para el desarrollo de la norma ISO 25000, establece un modelo de calidad en el

que se recogen las investigaciones durante los últimos 30 años para la caracterización de la calidad del producto software.

Este estándar propone un modelo de calidad que se divide en tres vistas: interior, exterior y en uso.

Estas vistas están compuestas por características, que se dividen en sub características, y que estas a su vez se componen de atributos.

Los atributos obtienen sus valores tras realizar mediciones sobre el software.

Estas mediciones dan como resultado una serie de métricas que se pueden clasificar en tres categorías según sea su naturaleza. (Norma ISO/IEC 9126-1:2001).

- ✓ Métricas básicas, que se obtienen directamente de analizar el código o la ejecución del software.
- ✓ Métricas de agregación, que consisten en la composición de una métrica a partir de un conjunto definido de métricas básicas, generalmente mediante una suma ponderada.
- ✓ Métricas derivadas, que son una función matemática que utiliza como entrada el valor de otras métricas.

El modelo establece diez características, seis que son comunes a las vistas interna y externa y cuatro que son propias de la vista en uso. Las características que definen las vistas internas y externas:

- ✓ Funcionalidad, capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.
- ✓ Fiabilidad, capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.
- ✓ Usabilidad, esfuerzo requerido por el usuario para utilizar el producto satisfactoriamente.
- ✓ Eficiencia, relación entre las prestaciones del software y los requisitos necesarios para su utilización.
- ✓ Mantenibilidad, esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software.
- ✓ Portabilidad, capacidad del software ser transferido de un entorno a otro.

Mientras que las características propias de la vista en uso, se muestran a continuación:

- ✓ Efectividad, capacidad del software de facilitar al usuario alcanzar objetivos con precisión y completitud.
- ✓ Productividad, capacidad del software de permitir a los usuarios gastar la cantidad apropiada de recursos en relación a la efectividad obtenida.
- ✓ Seguridad, capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.

- ✓ Satisfacción, capacidad del software de cumplir con las expectativas de los usuarios en un contexto determinado.

A continuación se detallan las subcaracterísticas correspondientes a la mantenibilidad.

- ✓ Analizabilidad, facilidad para analizar el software en busca de deficiencias e identificar sus componentes y artefactos.
- ✓ Capacidad de cambio, capacidad de permitir cambios en el software.
- ✓ Estabilidad, capacidad de evitar efectos inesperados tras realizar modificaciones en el software.
- ✓ Capacidad de pruebas, capacidad para validar los cambios en el software.
- ✓ Adherencia a las normas, cumplimiento de los estándares y convenciones de mantenibilidad. Hace referencia a todas las anteriores.

Pero si bien el modelo indica que estas subcaracterísticas a su vez se subdividen en atributos, no se especifica cuáles son esos atributos, ya que se entiende que estos son entidades dependientes del producto software y variarían según varié la naturaleza del software analizado: lenguaje, paradigma de programación, complejidad tecnológica.

Documentación: los manuales, los impresos y otra información descriptiva que explica el uso y/o la operación.

Procesamientos: los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

Control: los sistemas trabajan mejor cuando operan dentro de niveles de control tolerables de rendimiento por ejemplo: el sistema de control de un calentador de agua.

2.2.1. Tecnología Cliente – Servidor

La tecnología cliente - servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales. Desde el punto de vista funcional, se puede definir la computación Cliente – Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aun en entornos multiplataforma. En el modelo Cliente – Servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Además como veremos en el modelo de implementación, el concepto es utilizado en forma constante para varias funciones e implementado de distintas formas.

La arquitectura Cliente – Servidor dominan los sistemas basados en computadora: todo existe, desde redes de cajeros automáticos hasta internet, y esto es debido a que el software reside en una computadora – el cliente – solicita servicios y/o datos de otra computadora – servidor.

4.1.27. Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se lo conoce con el término front-end. Este normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir una interfaz gráfica de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos: (Allison, 2011).

- ✓ Administrar la interfaz de usuario.
- ✓ Interactuar con el usuario.
- ✓ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✓ Generar requerimientos de bases de datos.
- ✓ Recibir resultados del servidor.
- ✓ Formatear resultados.

4.1.28. Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrativo por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos: (Allison, 2011).

- ✓ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ✓ Procesar requerimientos de Bases de Datos.
- ✓ Formatear datos para transmitirlos a los clientes.
- ✓ Procesar la lógica de la aplicación y realizar validaciones a nivel de Bases de Datos

4.1.29. Características de un Sistema Cliente – Servidor

Fiabilidad

Probabilidad de que el sistema funcione correctamente durante un intervalo de tiempo.

Seguridad

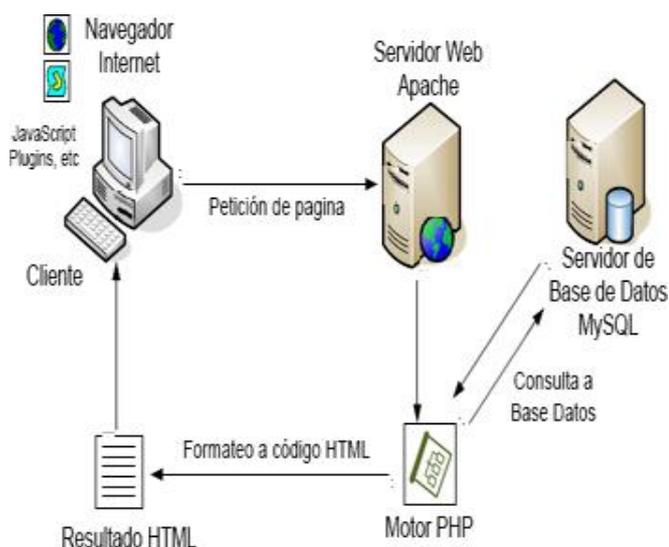
Representa la capacidad de que el sistema no afecte ni sea afectado por su entorno y el de quien lo utiliza.

Disponibilidad

Define la probabilidad de que el sistema esté funcionando en un tiempo determinado.

Protección

Representa la capacidad del sistema para protegerse a sí mismo de intrusiones accidentales o programadas.

Figura N° 6 Arquitectura Física Cliente - Servidor

FUENTE: <http://djesled.blogspot.pe>

2.3. Definición de Términos Básicos

4.1.30. Administración

La administración es la ciencia social y técnica encargada de la planificación, organización, dirección y control de los recursos (humanos, financieros, materiales, tecnológicos, el conocimiento) de una organización, con el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo de los fines perseguidos por la organización. (Bateman – Snell – Scott, 2009).

4.1.31. Administración de la Cobranza

Los diversos métodos de cobros y pagos mediante los cuales una organización mejora la eficiencia de la administración de su efectivo constituyen dos caras de la misma moneda. Producen repercusión conjunta sobre la eficiencia global de la administración del efectivo. La idea general

consiste en hacer efectivas las cuentas por cobrar tan pronto sea posible y diferir lo más que se pueda el pago de las cuentas por pagar, pero 40 manteniendo el mismo tiempo la imagen de crédito de la organización con los proveedores.

4.1.32. Archivo

Un archivo informático es un conjunto de información que se almacena en algún medio de escritura que permita ser leído o accedido por una computadora. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene. Los archivos informáticos se llaman así porque son los equivalentes digitales de los archivos en tarjetas, papel o microfichas del entorno de oficina tradicional. Los archivos informáticos facilitan una manera de organizar los recursos usados para almacenar permanentemente información dentro de un computador.

4.1.33. De Usuarios

Se usa para restringir el acceso a los módulos del sistema de ventas, realizados por el administrador del sistema de ventas de la micro y pequeña empresa San Santiago.

4.1.34. Actualizar Estado Productos

Actualiza automáticamente el estado de los productos.

4.1.35. Aplicaciones Web

Es el servicio final conjuncionado de tecnologías JAVA y el modelamiento en un software sobre un problema que se da solución como software. (Boehm, 1996).

4.1.36. Control Administrativo

El control administrativo es el proceso que permite garantizar que las actividades reales se ajusten a las actividades proyectadas. (Stoner, 1996).

4.1.37. Cuenta de Usuarios

En el contexto de la informática, un usuario es aquel que utiliza un sistema informático. Para que los usuarios puedan obtener seguridad, acceso al sistema, administración de recursos, etc., dichos usuarios deberán identificarse. Para que uno pueda identificarse, el usuario necesita una cuenta (cuenta de usuario) y un usuario, en la mayoría de los casos asociados a una contraseña.

4.1.38. Consultar Base de Datos

Es el proceso mediante el cual se consulta la base de datos de una determinada institución, mediante el cual se obtiene toda la información de la base de datos del sistema consultado. (Boehm, 1996)

4.1.39. Diseño de un Sistema Informático

Sistema de información que basa la parte fundamental de su procesamiento, en el empleo de la computación, como cualquier sistema,

es un conjunto de funciones interrelacionadas, hardware, software y de Recurso Humano. Un sistema informático normal emplea un sistema que usa dispositivos que se usan para programar y almacenar programas y datos. (Boehm, 1996)

4.1.40. El Control como Fase del Proceso Administrativo

El control es un elemento del proceso administrativo que incluye todas las actividades que se emprenden para garantizar que las operaciones reales coincidan con las operaciones planificadas. De tal manera, el control es un elemento clave en la administración.

4.1.41. Implementación de Sistema Informático

Una implementación es la instalación de una aplicación informática, realización o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política.

4.1.42. Información

Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que de un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a la información, primero se debe guardar lógicamente en archivos. (Crovetto, 2005).

4.1.43. Proceso de Venta

Es una secuencia lógica de cuatro pasos que emprende el vendedor para tratar con un comprador potencial y que tiene por objeto producir

alguna reacción deseada en el cliente (usualmente la compra). (DÍEZ DE CASTRO – LANDA, 1996)

4.1.44. Pago

Liquidación de una obligación pecuniaria mediante un desembolso en dinero mediante bienes o servicios aceptados como equivalente del dinero, la satisfacción del deudor o de su nombre al acreedor de una cantidad que se debe, terminado así la obligación entre ambos. (DÍEZ DE CASTRO – LANDA, 1996)

4.1.45. Reporte

Documento de salida originado por un procesamiento de datos y que tiene como destino un medio escrito. (CONSEJERÍA DE INDUSTRIA y COMERCIO 1996)

4.1.46. Recibo de Cobranzas

Es el comprobante de pago que se les otorga a los usuarios.

4.1.47. Micro y Pequeña Empresa

No hay unidad de criterio con respecto a la definición de la Micro y Pequeña Empresa, pues las definiciones que se adoptan varían según sea el tipo de enfoque.

Algunos especialistas destacan la importancia del volumen de ventas, el capital social, el número de personas ocupadas, le valor de la producción o el de los activos para definirla. Otros toman como referencia el criterio

económico – tecnológico (Pequeña Empresa precaria de Subsistencia, Pequeña Empresa Productiva más consolidada y orientada hacia el mercado formal o la pequeña unidad productiva con alta tecnología).

Por otro lado, también existe el criterio de utilizar la densidad de capital para definir los diferentes tamaños de la Micro y Pequeña Empresa. La densidad de capital relaciona el valor de los activos fijos con el número de trabajadores del establecimiento. Mucho se recurre a este indicador para calcular la inversión necesaria para crear puestos de trabajo en la Pequeña Empresa.

Un estudio realizado por la Organización Internacional del Trabajo (OIT) en setenta y cinco países encontró más de cincuenta definiciones distintas sobre Pequeña Empresa. Los criterios utilizados son muy variados, desde considerar la cantidad de trabajadores o el tipo de gestión, el volumen de ventas o los índices de consumo de energía, hasta incluso el nivel tecnológico, por citar los más usuales.

La OIT, en su Informe sobre fomento de las Pequeñas y Medianas Empresas, presentado en la 72ª reunión de la Conferencia Internacional del Trabajo, realizada en Ginebra en 1986, define de manera amplia a las Pequeñas y Medianas Empresas pues considera como tales, tanto a empresas modernas, con no más de cincuenta trabajadores, como a empresas familiares en la cual laboran tres o cuatro de sus miembros, inclusive a los trabajadores autónomos del sector no estructurado de la economía (informales).

La CEPAL define a la microempresa como unidad productora con menos de diez personas ocupadas incluyendo al propietario, sus familiares y a sus trabajadores tanto permanentes como eventuales.

El especialista de la pequeña empresa, Fernando Villarán en su trabajo para el Seminario de "Microempresas y Medio Ambiente" realizado en Lima durante el mes de Julio de 1993, señala, refiriéndose a la micro industria, que está caracterizada por su reducido tamaño , tiene de uno a cuatro personas ocupadas y presenta una escasa densidad de capital equivalente a 600 dólares por puesto de trabajo.

Con respecto a la Pequeña Empresa Industrial, la describe como empresa que tiene un tamaño de 5 a 19 personas ocupadas y una densidad de capital de aproximadamente 3000 Dólares por puesto de trabajo. Su base tecnológica es moderna y la productividad, es mayor que en la microempresa: Es capaz de generar excedentes, acumular y crecer, teniendo entonces, una mayor estabilidad económica que el sector microempresarial.

Como se observará, existen diversos criterios para caracterizar a la Micro y Pequeña Empresa. Sin embargo nuestra legislación, cuando de reglamentarla o promocionarla se trata usualmente ha utilizado, los montos anuales vendidos y/o el número de personas ocupadas o el valor de los activos, para establecer sus límites.

2.4. Operacionalización de Variables

El sistema de ventas mejorara el proceso de ventas de la micro y pequeña empresa San Santiago.

Tabla N° 1 Operacionalizacion De Variables

VARIABLES	INDICADOR	ITEMS	INDICE
VENTAS	REGISTRO DE VENTAS	QUESOS	PRECIO S/. (NUEVOS SOLES)

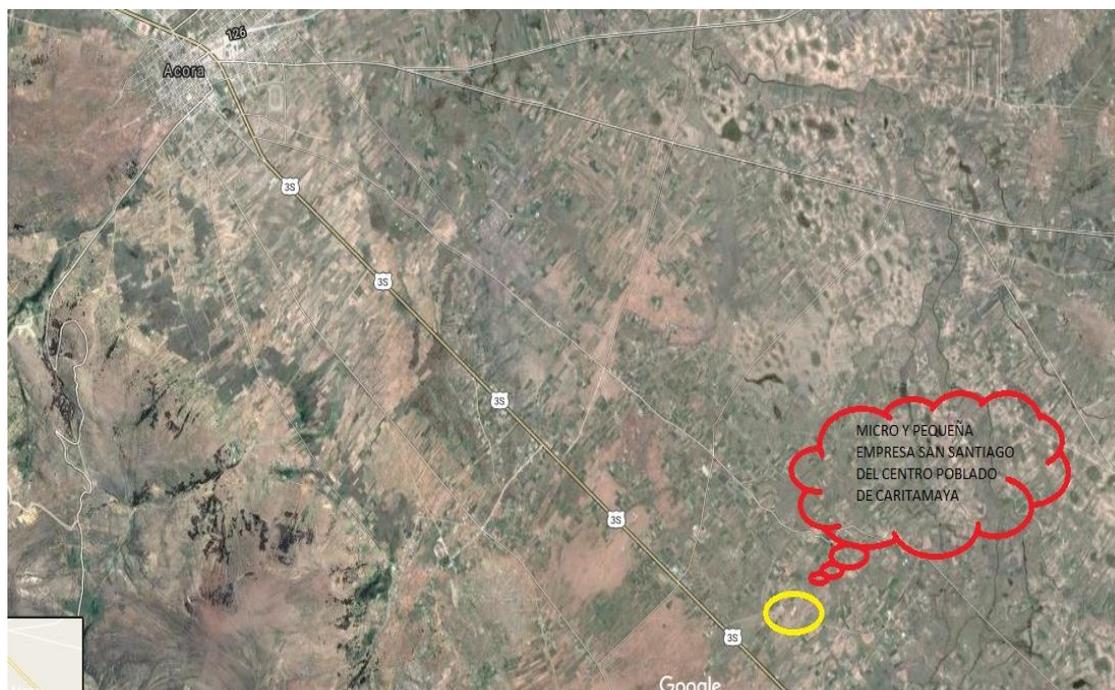
FUENTE: Elaboración propia

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. LOCALIZACIÓN

El presente trabajo de investigación se realizó en el área de Informática de la micro y pequeña empresa de queso “San Santiago”, que se encuentra ubicado en el sector de marquiri, centro poblado de caritamaya, distrito de acora, capital de la Provincia Puno, situada a 3867 msnm en la Meseta del Collao, al Centro - Sur del Lago Titicaca.

Figura N° 7 Ubicación De La Micro Y Pequeña Empresa San Santiago

FUENTE: Google maps. -16.011188, -69.745812

3.2. Población

La población está constituida por los trabajadores de la micro y pequeña empresa San Santiago con un total de 10 trabajadores.

- ✓ 1 Almacenero
- ✓ 1 Administrador
- ✓ 6 Vendedores
- ✓ 2 cajeros

3.3. Muestra

La muestra está conformada por la misma cantidad de personas de la población (10 trabajadores).

3.4. Diseño de la Muestra

La totalidad de la población que trabaja en el área de ventas en la micro y pequeña San Santiago es un número de 10 personas.

3.5. Método de Recolección de Datos

La información para el presente estudio de investigación fue obtenida mediante recopilación directa de la micro y pequeña empresa San Santiago a través de encuestas, entrevistas, observaciones, análisis de contenido y estudio de casos. METODOLOGÍA Y PROCEDIMIENTO

4.1.48. Metodología de Desarrollo Xp (Extreme Programming)

La programación extrema es una metodología muy rápida y ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

3.5.1.1. Planificación

- ✓ Se utiliza historias de usuarios: las necesidades, escritas por los usuarios, con la ayuda de los diseños, que quieren ser satisfechas con el sistema
- ✓ Se crean los planes de entrega, los cuales estiman el tiempo de desarrollo de las historias de usuario.
- ✓ Se llevan a cabo la planificación de iteración: identificar las historias de usuarios que se van a desarrollar en una iteración específica.
- ✓ Se desarrollan reuniones diarias, con el fin de facilitar la comunicación entre el grupo de trabajo y la exposición de los diferentes problemas.

3.5.1.2. Diseño

- ✓ Se escoge una metáfora de sistema, esto para facilitar el manejo consistente de los nombres de las clases y los métodos.
- ✓ Se proponen soluciones a problemas técnicos o de diseño.
- ✓ Se ignoran las funcionalidades extra que podrían incorporarse al proyecto, centrar en lo principal.
- ✓ Se remueve la redundancia, se eliminan las funcionalidades no necesarias y se renuevan los diseños obsoletos.

3.5.1.3. Codificación

- ✓ Se utilizan estándares para escribir el código.
- ✓ Se crean las pruebas antes de empezar a codificar, lo cual hará más sencillas y efectivas las pruebas.
- ✓ Se realiza en equipos de trabajo y luego se llevó a cabo una integración paralela (debido a esta integración no se garantiza la consistencia y la calidad a necesidades de hacer pruebas exhaustivas)
- ✓ Se deja la optimización para el final. Una vez que el código requerido este completo.

3.5.1.4. Pruebas

- ✓ Se crean pruebas de aceptación a partir de las historias de usuario.
- ✓ El cliente es responsable de revisar, tanto las pruebas de aceptación, como los resultados obtenidos al ser estas aplicadas.

Tabla N° 2 Ficha de Evaluación de la Calidad del Producto Estándar

ISO – 9126

INDICADORES	PUNTUACION				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas específicos.					
Seguridad: referido a la capacidad del producto software para proteger la información y los datos					
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad					
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.					
Tolerancia a Fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					
Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones referidas a la fiabilidad.					
3. USUABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					
Facilidad de Aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.					
Atracción: la capacidad del software para atraer al usuario.					
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle					
4. EFICIENCIA					
Comportamiento Temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
Utilización de Recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones					
Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
5. MANTENIBILIDAD					
Analizabilidad: capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
Confiabilidad: capacidad del producto software de permitir implementar una modificación específica. La implementación incluye los cambios en el diseño, el código y la documentación					
Estabilidad: capacidad del producto software de evitar los defectos inesperados de las modificaciones.					
Facilidad de Prueba: capacidad del producto software de permitir validar las partes modificadas.					
Conformidad: capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado					
Facilidad de Instalación: la capacidad del producto software para ser instalado en un ambiente determinado.					
Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					
Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					
SUB TOTALES					
TOTAL					

FUENTE: ficha de evaluación ISO – 9126

Tabla N° 3 Escala Valorativa (Escala De Likert)

INDICADOR CUALITATIVO	VALOR
DEFICIENTE	1
MALO	2
REGULAR	3
BUENO	4
MUY BUENO	5

FUENTE: Escala Valorativa. (Escala de Likert)

Tabla N° 4 Cuadro De Decisiones Iso 9126

CLASIFICACION	INTERVALO	DECISION
A) INACEPTABLE	[27 - 54 >	
B) MINIMAMENTE ACEPTABLE	[54 - 81 >	
C) ACEPTABLE	[81 - 95 >	
D) CUMPLE LOS REQUISITOS	[95 - 122 >	
E) EXCEDE LOS REQUISITOS	[122 - 135]	

FUENTE: Cuadro de Decisiones ISO – 9126

3.6. Requerimientos del Sistema

4.1.49. Requerimientos Funcionales:

Se han definido para el sistema, los siguientes puntos más relevantes que el software debe poder realizar.

R1: Deben existir perfiles para el ingreso al sistema, este le dará mayor seguridad a la información que este maneje.

R2: El sistema debe estar en la capacidad de registrar nuevos productos.

R3: El sistema debe estar en la capacidad de poder editar datos de productos.

R4: Por necesidad debe existir la posibilidad de consultar datos de los productos.

R5: Modulo para generar las fichas de los nuevos productos y clientes.

R6: Modulo para generar recibos y facturas de pago.

R7: El sistema debe contar con modulo para generar reportes de los productos, clientes entre otros.

4.1.50. Requerimientos no Funcionales:

- ✓ Aplicación multiplataforma.
- ✓ Interfaz agradable para fácil entendimiento del software.
- ✓ Disponibilidad del sistema de encontrarse disponible todo los días.
- ✓ Portabilidad estará diseñado en un lenguaje multiplataforma.
- ✓ Mantenimiento y escalabilidad diseñado pensando en el crecimiento del sistema.

4.1.51. REQUERIMIENTOS TÉCNICOS:

Hardware

- ✓ Microprocesador INTEL 3.4Ghz.
- ✓ RAM de 4Gb.
- ✓ Tarjeta de video 128 Mb(16Mb)*

- ✓ Disco duro con espacio disponible de 10 MB (6Mb)*
- ✓ Monitor LED 18"
- ✓ Unidad de almacenamiento (CDROM y/o USB)

Software

- ✓ Plataforma Windows 7,8, 10 Profesional ®
- ✓ Apache
- ✓ Servidor Web: App Server
- ✓ XAMPP
- ✓ Netbeans – java
- ✓ MySql – Workbench
- ✓ jdk-8u25-windows-x64
- ✓ Rational rose 2007 enterprise edition
- ✓ Internet Explore 8.0
- ✓ Google Chrome
- ✓ Mozilla Firefox

3.7. Diseño del Sistema

Para el diseño de la aplicación del equipo de trabajo siguió las recomendaciones de la metodología XP, tratando de evitar las soluciones complejas, y se trabajo es una sola iteración. Sin pensar en los que vendrían más adelante.

Otro aspecto importante en el diseño, fue la constante reestructuración del código. El principal objetivo de la reestructuración fue evitar la duplicación del código poniendo en práctica la normalización de datos simplificando y hacerlo más flexible para facilitar los posteriores cambios. Esto se hizo constantemente en la programación de la aplicación.

3.8. Ciclo de Vida del Desarrollo del Software

Llamado también proceso de desarrollo de software.

El proceso de desarrollo de software está enmarcado por el modelo de desarrollo iterativo incremental, donde cada iteración que presenta mencionado proceso está determinado por el ciclo de vida clásico en sus diferentes etapas:

Análisis, Diseño, Codificación, Pruebas.

3.9. Analisis

Es la primera actividad técnica que se desarrolla como parte de la ingeniería de software orientada a objetos, la cual está compuesta de las siguientes actividades:

- ✓ Descripción del ámbito del problema
- ✓ Elaboración de escenarios y sub. Escenarios.
- ✓ Elaboración de diagramas de casos de uso.
- ✓ Identificación y descripción clases y objetivos.
- ✓ Elaboración de diagrama de interacción.

3.10. Diseño

En esta fase se realiza la solución lógica del sistema que comprende el desarrollo de las siguientes actividades.

- ✓ Elaboración de diagrama de clases.
- ✓ Elaboración del diagrama de componentes.
- ✓ Diseño y descripción de interfaz de componentes.

3.11. Implementación

En esta fase de desarrollo del sistema se traducen las especificaciones detalladas en las etapas del Análisis y el Diseño en los lenguajes o tecnología de programación:

- ✓ Pruebas no convencionales.
- ✓ Pruebas convencionales.
- ✓ Métricas basadas en puntos de función.

3.12. Implantación

Esta fase implica realizar actividades técnicas siguientes:

- ✓ Instalación del sistema
- ✓ Capacitación y puesta en marcha el sistema
- ✓ Elaboración de encuesta para la evaluación del sistema.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Después de describir los resultados antes de implantar el sistema denominado **“SISTEMA DE VENTAS PARA MICRO Y PEQUEÑAS EMPRESAS, APLICADO A LA EMPRESA SAN SANTIAGO, ACORA-PUNO - 2017”**, se discutió los resultados en función a los objetivos planteados.

Realizando prueba de hipótesis para dos medias muestrales y poder así sustentar uno de los objetivos específicos planteado *Comparar los procesos de ventas luego de implementar el sistema informático.*

4.1. Prueba de Hipótesis Para Diferencia de dos Medias Muestrales

4.1.1. Planteamiento de Hipótesis

$$H_0 : \mu A = \mu B$$

H_0 : Promedio Antes del sistema = Promedio Después del sistema

(Hipótesis nula)

H1 : $\mu A \neq \mu B$

H1 : Promedio Antes del sistema \neq Promedio después del sistema

(Hipótesis alterna)

4.1.2. Fijar el Nivel de Significancia (α)

$\alpha = 0.05$ (95 % de nivel de confianza)

4.1.3. Regla de Decisión Utilizando

Si $T_c > T_t$ Se rechaza la H_0 y se Acepta la H_1

4.1.4. Prueba Estadística

Tabla N° 5 Encuesta del Antes y Despues del Sistema

ITEM	ANTES	DESPUES
1	10	1
2	15	1.5
3	5	1
4	8	1
5	10	1
6	15	2
7	5	1
8	10	1
9	10	1
10	5	1
MEDIA	9.3	1.15
VARIANZA	12.41	0.1025

FUENTE: Elaboración Propia

Figura N° 8 Formula De Prueba De Hipótesis Para Diferencia De Dos Medias

Muestrales

$$T_c = \frac{|X_A - X_B|}{\sqrt{\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}}}$$

$$T_c = \frac{|9.3 - 1.15|}{\sqrt{\frac{12.41}{10} + \frac{0.1025}{10}}} = 13.02697303$$

$$T_t = T_{10, \alpha = 0.05} = 2.1009$$

4.1.5. DECISIÓN

Aplicando dicha fórmula se obtuvo un **Tc = 13.02697303**. de tal manera se rechaza la Ho y se acepta la H1, demostrando así que si existe diferencia entre el antes y el después de haber implantado el sistema de ventas en la micro y pequeña empresa San Santiago, para un nivel de significancia de **0,05** y para un nivel de confianza del **95%**.

Para sustentar el otro objetivos específico planteado *Diseñar e Implementar un sistema informático de ventas en la micro y pequeña empresa "San Santiago" del centro poblado de caritamaya, distrito de Acora, departamento de Puno*. Nos basamos en lo siguiente:

4.2. Fases de Desarrollo

4.2.1. Reglas de Negocio

4.2.1.1. Gestionar Sistemas

- ✓ En primer lugar se debe definir los parámetros base para los funcionamientos del sistema así mismo ver “Como, donde y quienes” utilizaran el sistema.
- ✓ La cuenta de usuario será asignado al administrador del sistema y a los empleados que será capacitados.
- ✓ Habrá una persona encargada del mantenimiento de este modulo
- ✓ La contraseña de los usuarios por seguridad tendrá que ser formado por letras (mayúsculas y minúsculas) y números.
- ✓ El administrador definirá la información que es la correcta y velara por la integridad.

4.2.1.2. Controlar Ventas

- ✓ El cliente podrá realizar pago solo con efectivo.
- ✓ Se registran todo los clientes.
- ✓ Los reportes de venta se deben generar al final de cada día.

4.2.1.3. Controlar Almacén

- ✓ Se debe controlar el llenado del Kardex
- ✓ Se debe controlar las fechas de entrada y salida de bienes a almacén

- ✓ Para sacar un producto de almacén debe haber un pedido de bienes
- ✓ Todo producto debe estar estrictamente controlado con la fecha de ingreso.

4.2.1.4. Modelo de Negocio

- ✓ Cliente
- ✓ Proveedor
- ✓ Administrador
- ✓ Vendedor
- ✓ Almacenero

4.2.1.5. Identificación de Casos de Uso del Negocio

- ✓ Gestión de sistemas
- ✓ Control de ventas
- ✓ Control almacén

4.3. Diagrama Uml Modelamiento del Sistema

4.3.1. Diagrama de Casos de Uso

4.3.1.1. Diagrama Casos de Uso Actores del Sistema

Especificaciones casos de uso del negocio

4.3.1.1.1. Gestionar Sistema

✓ Descripción

Se tendrá en cuenta que se contara con un responsable el cual velara por el rendimiento efectivo del sistema. Asimismo, quien brindara soporte a las acciones administrativas propias del sistema.

✓ Objetivo

Almacenar información base

✓ Responsable

Administrador

4.3.1.1.2. Controlar Ventas

✓ Descripción

Es el caso de uso del negocio que permite ingresar y modificar los clientes, es donde brinda el soporte al proceso de ventas en sí que realizan diariamente la entidad. Puesto que con este componente se registran las ventas y reportes de las ventas realizadas

✓ Objetivos

Almacenar

✓ Responsable

Cliente

Administrador y Vendedor

4.3.1.1.3. Controlar Almacén

✓ Descripción

Es el caso de uso del negocio que permite controlar de flujo de entrada y salida de productos de almacén. Así mismo permite controlar el stock de los mismos para así realizar requerimientos de pedidos a proveedores.

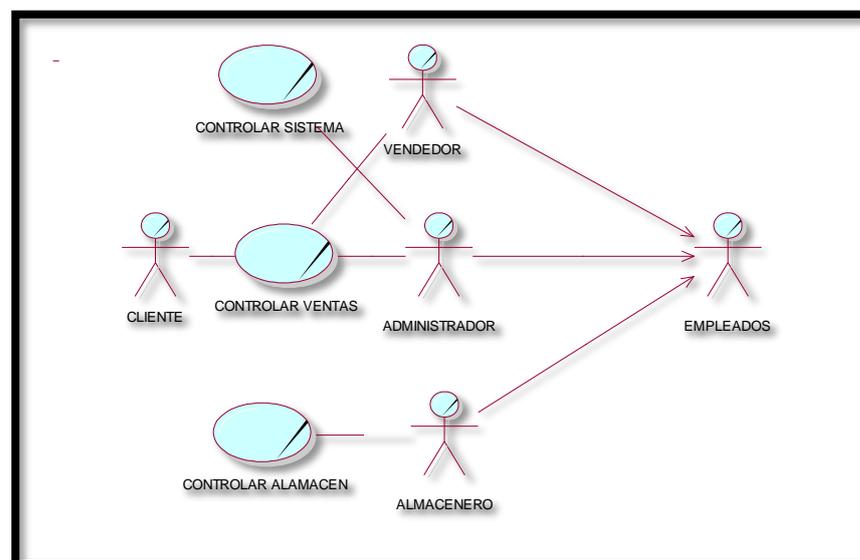
✓ Objetivos

Controlo el stock de los productos

✓ Responsables

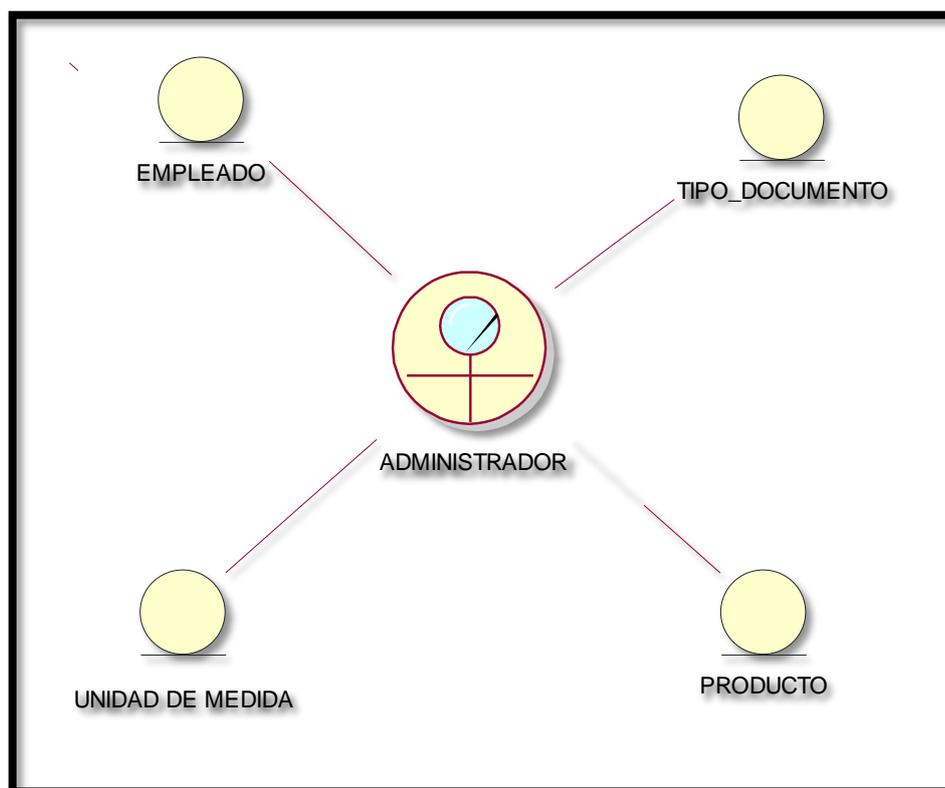
Almacenero

Figura N° 9 Modelo de Casos de Uso



FUENTE: elaboración propia

Figura N° 10 Modelo de Objetos Gestionar Sistema



FUENTE: Elaboración Propia

Se tendrá en cuenta que se contará con un responsable el cual velará por el rendimiento efectivo del sistema. Asimismo, quien brindará soporte a las acciones administrativas propias del sistema.

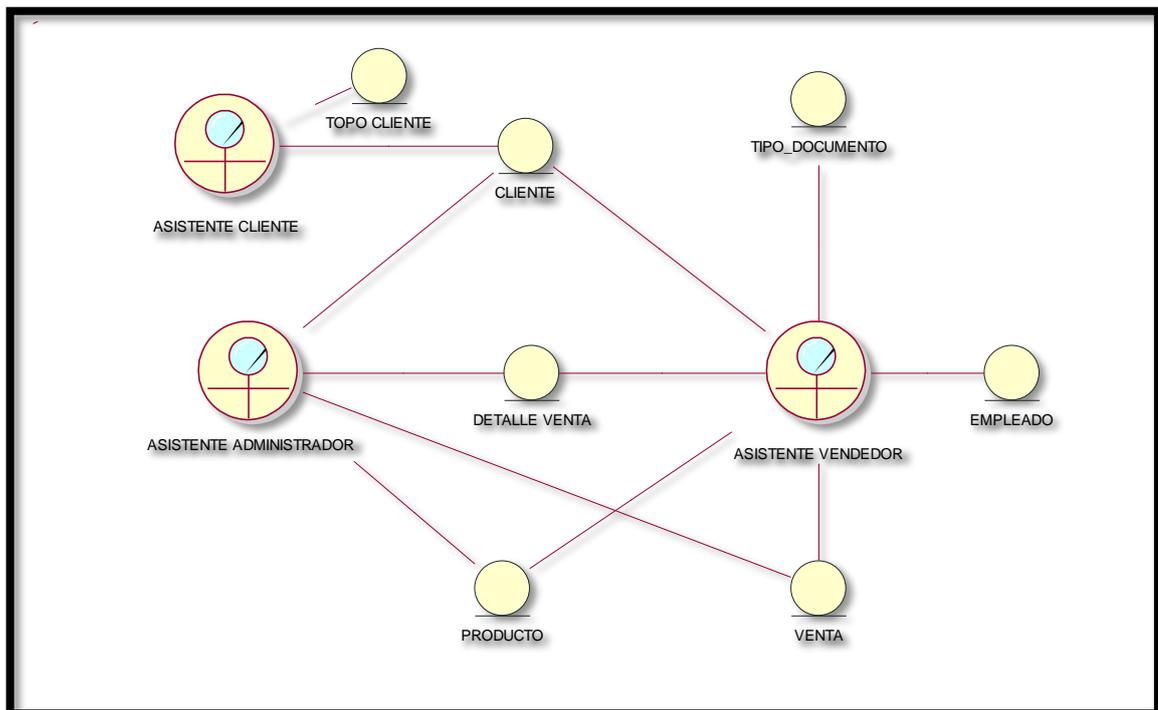
✓ Objetivos

Almacenar información base

✓ Responsables

Administrador

Figura N° 11 Modelo de Objetos Controlar Ventas



FUENTE: Elaboración Propia

Es el caso de uso del negocio que permite ingresar y modificar los clientes, es donde brinda el soporte al proceso de ventas en sí que realizan diariamente la entidad. Puesto que con este componente se registran las ventas y el ingreso de dinero en caja. Además, permitirá realizar reportes de las ventas realizadas.

✓ Objetivos

Almacenar ventas diarias

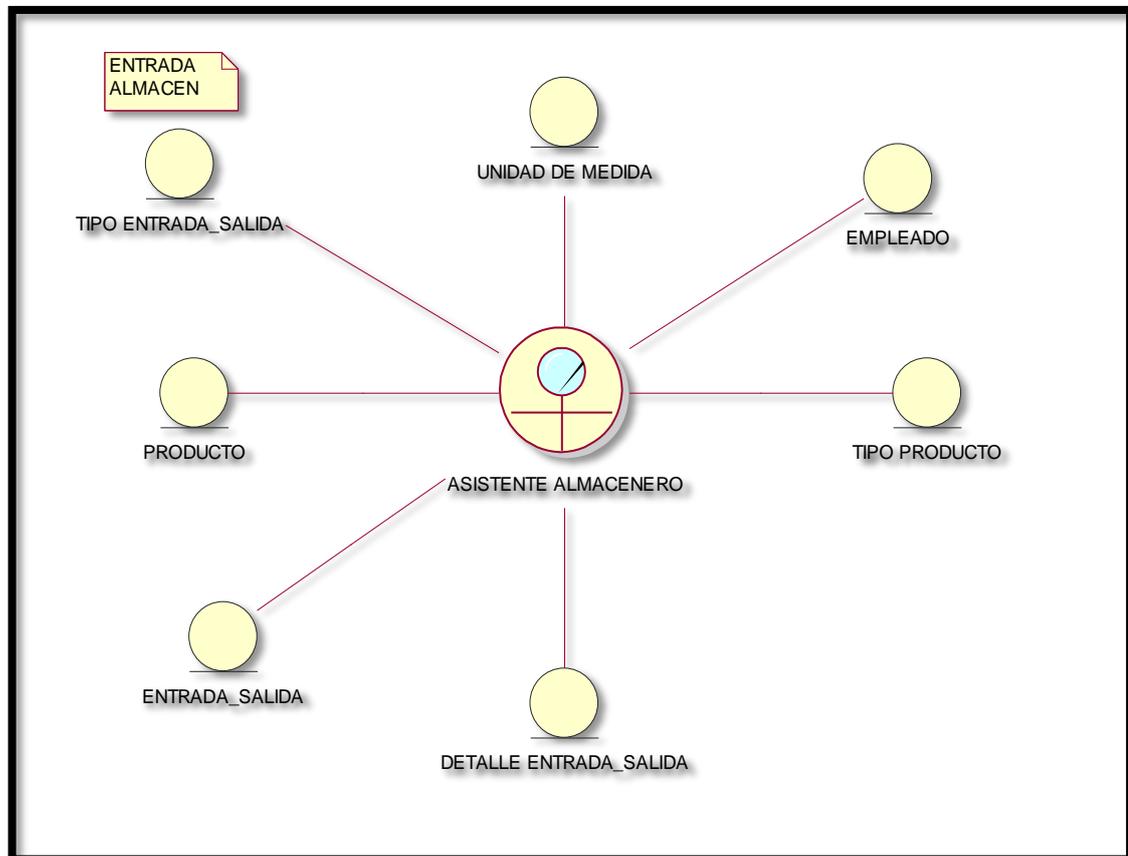
✓ Responsable

Asistente cliente

Asistente administrador

Asistente vendedor.

Figura N° 12 Modelo de Objetivo Controlar Almacén



FUENTE: Elaboración Propia

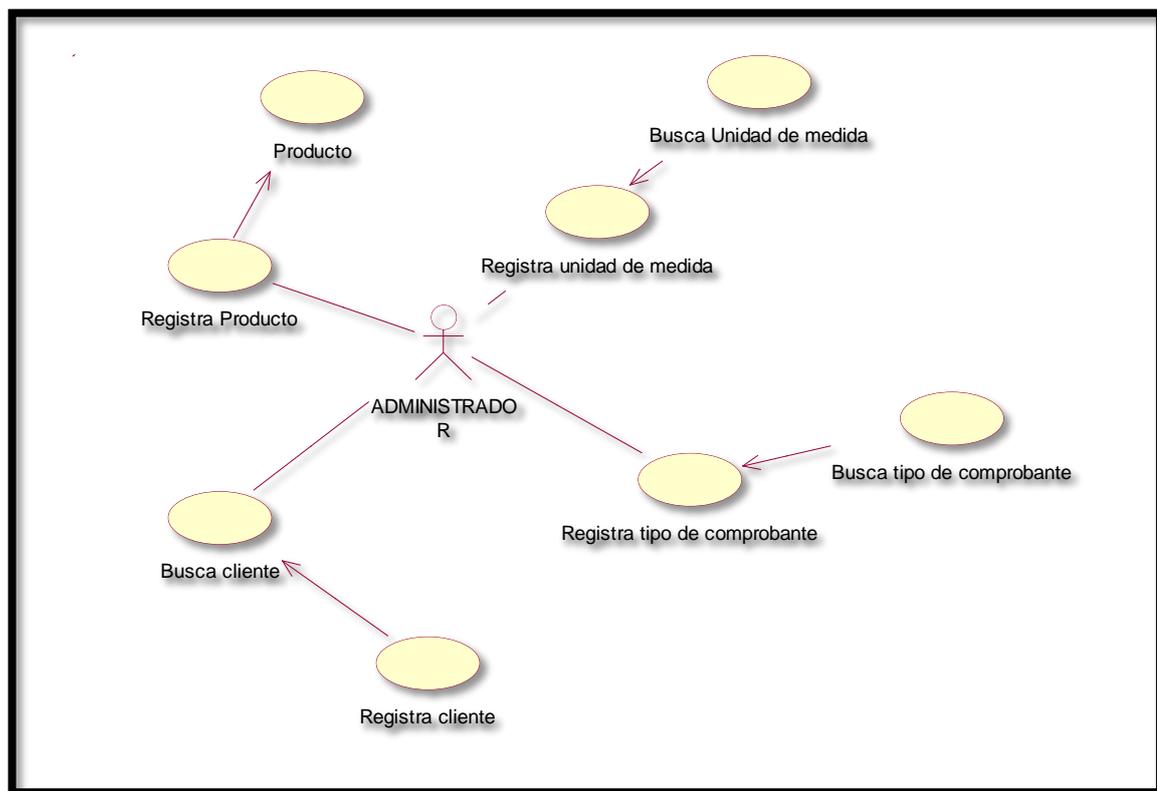
Es el caso de uso del negocio que permite controlar de flujo de entrada y salida de productos de almacén, así mismo permite controlar el stock de los mismos para así realizar requerimientos de pedidos a proveedores.

✓ Objetivos

Controlar el stock de los productos

✓ Responsables

Asistente almacenero

Figura N° 13 Modelo de Casos de Uso de Requerimiento Gestionar Sistema


FUENTE: Elaboración Propia

Se tendrá en Cuenta que se contara con un responsable el cual velara por el rendimiento efectivo del sistema. Asimismo, que brindara soporte a las acciones administrativas propias del sistema.

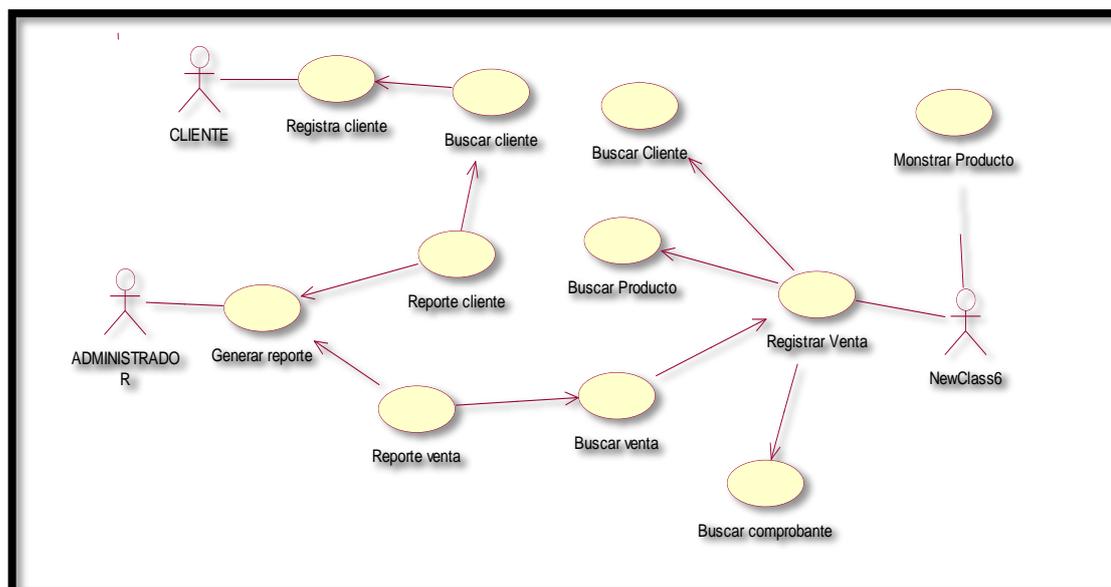
✓ Objetivo

Almacenar información base

✓ Responsables

Administrador

Figura N° 14 Modelo de Caso de Uso de Requerimiento Controlar Venta



FUENTE: Elaboración Propia

Es el caso de uso del negocio que permite ingresar y modificar los clientes, es donde brinda el soporte al proceso de ventas en sí que realizan diariamente la entidad. Puesto que con este componente se registran las ventas y el ingreso de dinero en caja. Además, permitirá realizar reportes de las ventas realizadas.

✓ **Objetivos**

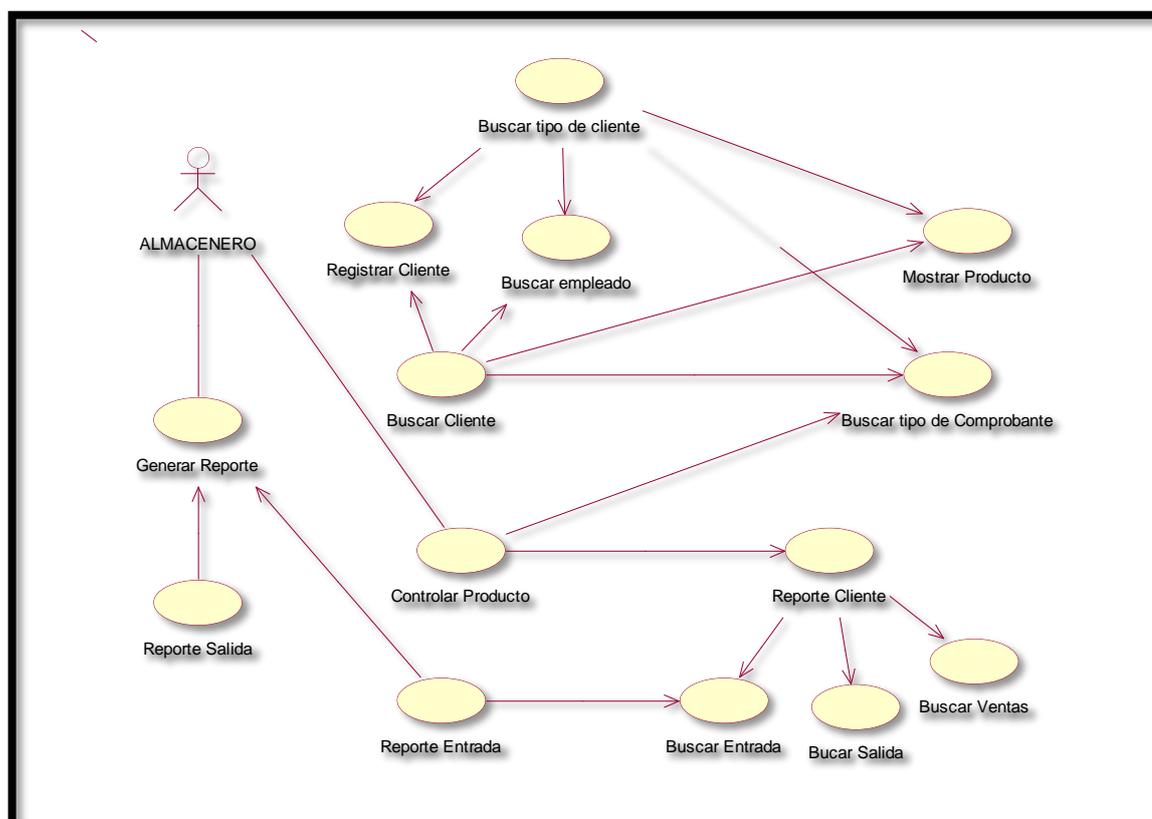
Almacenar ventas diarias

✓ **Responsable**

Cliente

Administrador y Vendedor

Figura N° 15 Modelo de Casos de Uso de Requerimiento Controlar Almacén



FUENTE: Elaboración Propia

Es el caso de uso del negocio que permite controlar el flujo de entrada y salida de producto de almacén, así mismo permite controlar el Stock de los mismos para así realizar requerimientos lácteos.

✓ Objetivo

Controlar el Stock de los productos

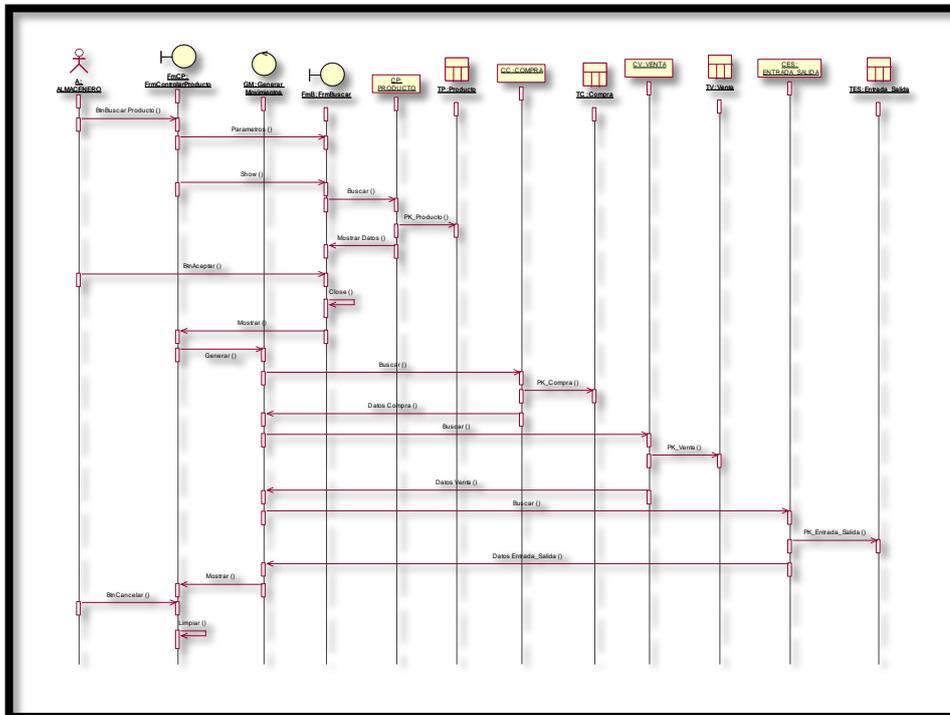
✓ Responsables

Almacenero

4.3.2. Modelo de Secuencia

4.3.2.1. Gestionar Sistema

Figura N° 16 Modelo de Secuencia: Controlar Producto



FUENTE: Elaboración Propia

Es el caso de uso modelo de secuencia del negocio que permite controlar de flujo de entrada y salida de producto de almacén, así mismo permite controlar el stock de los mismos para así realizar requerimientos de pedidos a proveedores.

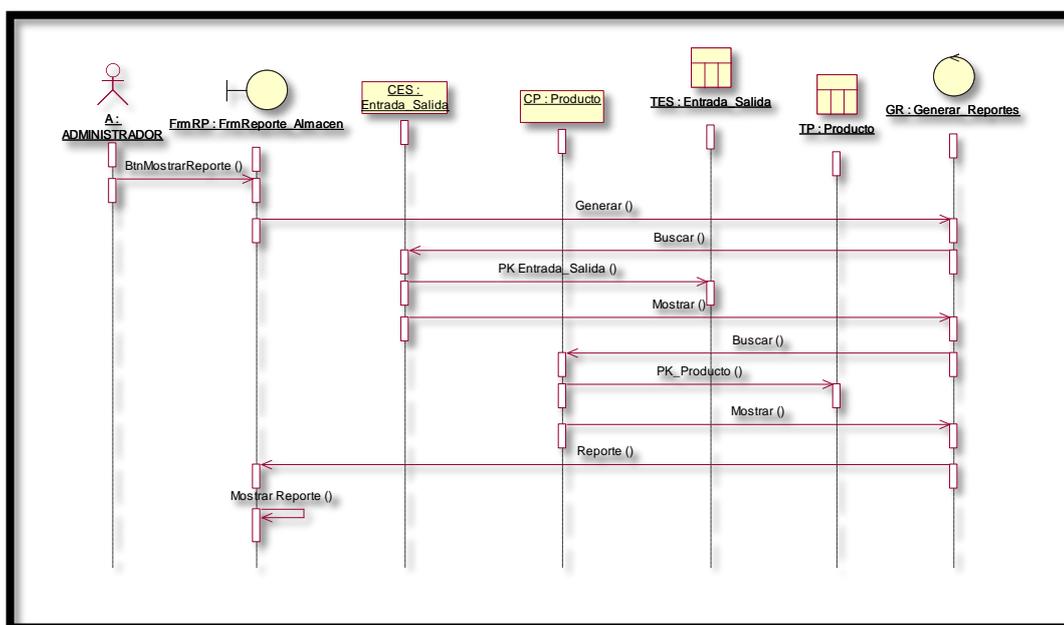
✓ **Objetivos**

Controlar el stock de los productos

✓ **Responsables**

Almacenero

Figura N° 17 Modelo de Secuencia: Generar Reporte Almacen



FUENTE: Elaboración Propia

Es el caso de uso modelo de secuencia del negocio que permite controlar que flujo de entrada y salida de productos de almacén, así mismo permite controlar el Stock de los mismos para así realizar requerimientos de pedidos a proveedores.

✓ Objetivos

Generar reporte de almacén

✓ Responsables

Administrador

CAPÍTULO V

CONCLUSIONES

- ✓ Se implementó el diseño de un sistema informático para facilitar la administración, entendimiento del mismo haciendo más fácil la integración de otros módulos o componentes para que se integre y facilite a cualquier plataforma de hardware y software.
- ✓ Se mejoró **SIGNIFICATIVAMENTE** el proceso de ventas en la micro y pequeña empresa San Santiago, observando el 100% de los encuestados opino que si se mejor el proceso de ventas, además en él se puede observar que el tiempo promedio para brindar información es de 1.15 (un minuto y quince segundos). De la post – test nos muestra el 100% de los encuestados opinaron que el sistema mejoro el control de productos por tanto podemos afirmar que el sistema informático implementado pequeña empresa San Santiago si mejoro con 0.05 de nivel de significancia o con un nivel de confianza de 95%.

CAPÍTULO VI

RECOMENDACIONES

- ✓ Se recomienda tener en cuenta el uso del software como alternativa de desarrollo del sistema.
- ✓ Para que el sistema crezca hasta un nivel gerencia y estratégico. Deberán tener en cuenta en proyectos de desarrollos de módulos de gestión, que estos emitan reportes que sea capaz de hacer ver cómo va el giro del negocio, tendencia y además ayudar a tomar decisiones para más adelante.
- ✓ Realizar una continua actualización de información y preparación en el manejo del sistema, por parte de los usuarios pertenecientes a la empresa.

CAPÍTULO VII

REFERENCIAS

ALEJANDRO, P. (2002). *Base de Datos y UML*. (U. N. Rioja, Ed.)

ALEJANDRO, P. (2003). *Base De Datos Y Uml*. (L. E.-U. RIOJA, Ed.)

ALIAGA ALVARES, S. (2009). *diseño e implementación de una plataforma de tele cobranzas integrado al sistema e-government de una empresa de recaudación tributaria*. tesis de grado publicado, pontificia universidad católica del Perú- lima . lima.

ALLISON. (2011). *La Biblia de MySQL y Apache para Desarrolladores Web y CMS* (IEEE Tras ed.). O'Reilly.

ÁLVAREZ GENDIN, S. (2000). *Teoría y Práctica de lo Contencioso de Procesos de Ventas*. Barcelona - España: Bosch ANDRADE , DAVI Y FALK J. A. (2001). *Eficacia de sistemas de información e percepción de mudanza organizacional: un estudio de caso*. *Revista de Administración contemporánea*, 5(3):53-84

- ARANA, J. (2014). *Desarrollo e implementación de un sistema de gestión de ventas de repuestos automotrices en el almacén de auto respuestos eléctricos marcos en la parroquia pasorja canton Guayaquil, provincia de guayas*. La libertad, Ecuador: UpseBALESTRINI, M. (2006). *Como se Elabora el Proyecto de Investigación* (Quinta ed.). Caracas: Consultores Asociados. BATEMAN, T. S. (2009). *administración, liderazgo y colaboración en un mundo competitivo*. MEXICO: McGraw-Hill.
- BARKLEY, E. (2009.). *Técnicas de Aprendizaje Colaborativo*. Madrid: Ediciones Morata
- BEVAN, N. (2005). *problemas de usabilidad en diseño de sitios web (1998), guías y estándares de usabilidad web*. nationalphysicallaboratory, usabilityservices.
- CHURA SOTOMAYOR, H. (2015). *Sistema de administración de ventas de una micro y pequeña empresa en Azángaro, 2015*
- DUBOIS, P. (2001). *Mysql*. Prentice hall.
- E, K. K. (2007). *Informática de Sistemas* (Última ed.). (Uned, Ed.) Lima, Perú: rama.
- EVERTON, F. C. (2012). *sistema de control de cobranzas para la administración del colegio de contadores públicos de puno - 2012*. tesis de grado universidad nacional del altiplano - puno. puno, puno, Perú. CHAMBERLIN, D. (1996). *Using the new db2: IBMs objetc-related database system*.
- JULIE, K. &. (2005). *Análisis y Diseño de Sistemas*. México: Prentice Hall.

- LÓPEZ, G. B. (2009). *Investigación Estratégica*. Colombia: Artes Gráficas Industriales Ltda.
- LLACCHUA GUTIERRES, M. (2007). *Diseño de un Sistema de Comercialización para el Supermercado Minimarket Tito's*.
- MULLER, M. (2009.). *Fundamento de Administración de los Inventarios*. Bogotá: Grupo Editorial Norma
- ROMANELLI M., R. (2009). *Diseño de un Sistema de Información para la Gerencia de Ventas de una Empresa de Mantenimiento y suministro de Equipos Analíticos de Laboratorio*. Barcelona, Bolívar, España: Universidad de Oriente Núcleo de Anzoátegui.
- SILVA, S. A. (2008). *Diseño y Optimización del Proceso de Gestión y ejecución de la Venta Mayorista para una Empresa Tipo Home Improvement*.
- STALLMAN, R. (2004). *Software libre para una sociedad libre*. Edición, traficantes de sueños. Madrid. pp. 45-47
- VALDERRAMA M., S. (2006). *Pasos para Elaborar Proyectos y Tesis de Investigación Científica*. Lima, Perú: San Marcos.
- VASQUEZ RIOS, D. (2008). *Análisis y Diseño de un Sistema Informático para el Control de los Procesos de Comercialización de la Empresa Grupo Selva Sac. de Tarapoto - Perú*. Tarapoto, Perú.
- VILEMA ESCUDERO, M. D. (2007). *Diseño de un Sistema de Información Comercial para Distribuidora La Familia*.

.WEBGRAFIA

(s.f.). Recuperado el 16 de Enero de 2017, de

http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n

(s.f.). Recuperado el 16 de Febrero de 2015, de <http://json.org/json-es.html>

(s.f.). Recuperado el 20 de Febrero de 2017, de

<http://www.desarrolloweb.com/manuales/manual-jquery.html>

(s.f.). Recuperado el 14 de Febrero de 2017, de

<http://teleformacion.edu.aytolacoruna.es/PASCAL/document/modular.htm>

#intro

(s.f.). Recuperado el 23 de febrero de 2017, de <http://iso2500.com/index.php/iso>

9126, N. I. (2001). *Software engineering - product quality - part 1*.fleitman.

(s.f.). Recuperado el 4 de Enero de 2017, de

<http://www.fleitman.net/articulos/sisInfo.pdf>

(s.f.). Recuperado el 1 de Diciembre de 2017, de

<http://repositorio.upse.edu.ec/bitstream/46000/1585/1/DESARROLLO%20>

[E%20IMPLEMENTACI%C3%93N%20VENTAS%20DE%20REPUESTOS](http://repositorio.upse.edu.ec/bitstream/46000/1585/1/DESARROLLO%20)

[%20AUTOMOTRICES%20EN%20EL%93N%20GUAYAQUIL%2C%20P](http://repositorio.upse.edu.ec/bitstream/46000/1585/1/DESARROLLO%20)

[ROVINCIA%20DEL%20GUAYAS.pdf](http://repositorio.upse.edu.ec/bitstream/46000/1585/1/DESARROLLO%20)

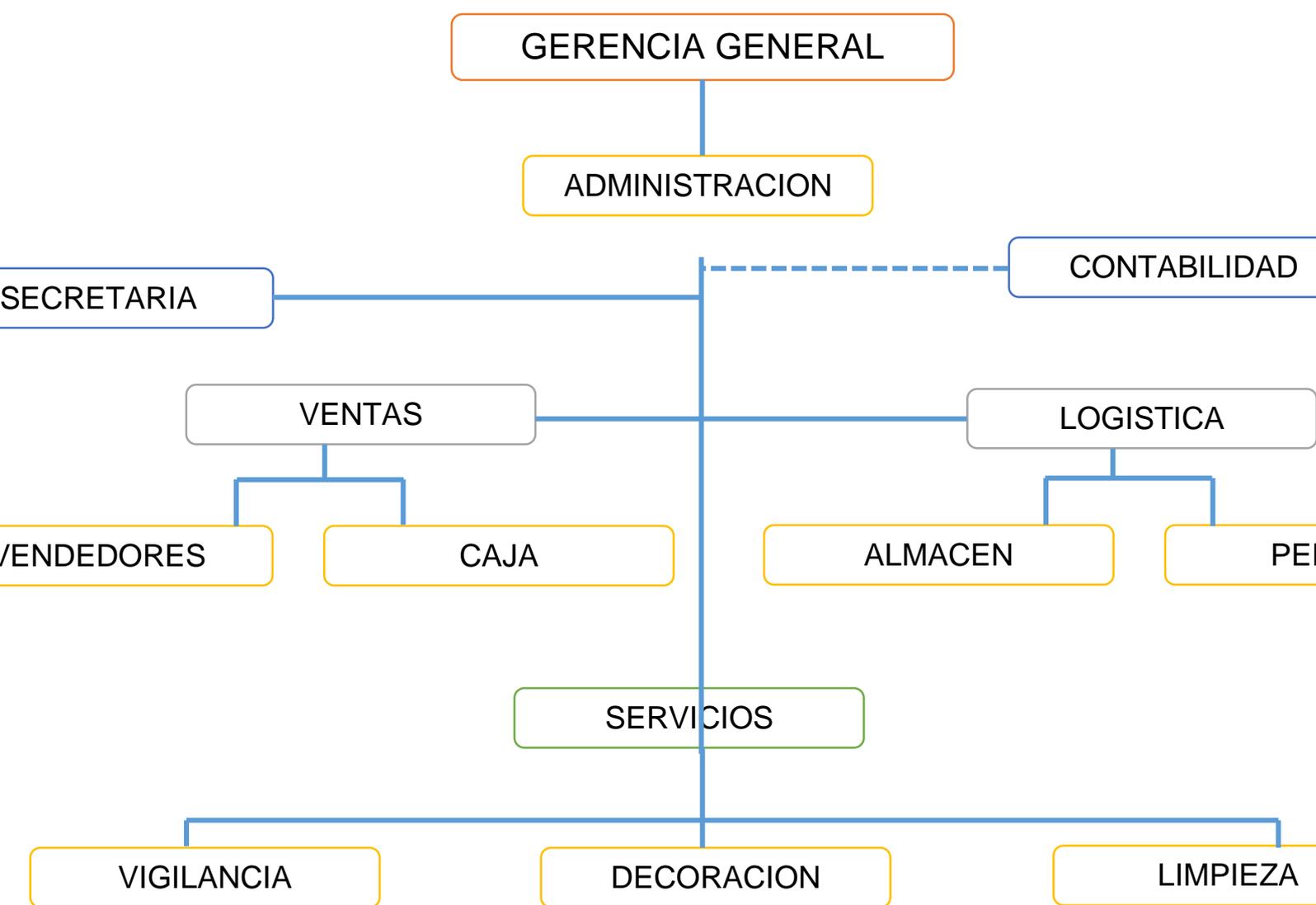
(s.f.). Recuperado el 2 de Enero de 2017, de

<http://www.fleitman.net/articulos/sisInfo.pdf>[http://repositorio.unap.edu.pe/h](http://repositorio.unap.edu.pe/handle/UNAP/1758)

[andle/UNAP/1758](http://repositorio.unap.edu.pe/handle/UNAP/1758)

ANEXOS

ANEXO: Estructura Organica de la Micro y Pequeña Empresa San Santiago del Centro Poblado de Carita



ANEXO: Validación Del Software

Ficha de Evaluación de la Calidad Del Producto Estándar Iso – 9126

INDICADORES	PUNTUACION				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					X
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.				X	
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas específicos.				X	
Seguridad: referido a la capacidad del producto software para proteger la información y los datos					X
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad				X	
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.			X		
Tolerancia a Fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.				X	
Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.				X	
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones referidas a la fiabilidad.				X	
3. USUABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.				X	
Facilidad de Aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.				X	
Atracción: la capacidad del software para atraer al usuario.			X		
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.				X	
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle				X	
4. EFICIENCIA					
Comportamiento Temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.				X	
Utilización de Recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones				X	
Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					X
5. MANTENIBILIDAD					
Analizabilidad: capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.			X		
Confiabilidad: capacidad del producto software de permitir implementar una modificación específica. La implementación incluye los cambios en el diseño, el código y la documentación				X	
Estabilidad: capacidad del producto software de evitar los defectos inesperados de las modificaciones.				X	
Facilidad de Prueba: capacidad del producto software de permitir validar las partes modificadas.				X	
Conformidad: capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					X
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado				X	
Facilidad de Instalación: la capacidad del producto software para ser instalado en un ambiente determinado.				X	
Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.				X	
Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.				X	
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					X
SUB TOTALES			9	7	2
TOTAL				6	5
				110	

Escala Valorativa (Escala De Likert)

INDICADOR CUALITATIVO	VALOR
DEFICIENTE	1
MALO	2
REGULAR	3
BUENO	4
MUY BUENO	5

Cuadro de Decisión Iso 9126

CLASIFICACION	INTERVALO	DECISION
A) INACEPTABLE	[27 - 54 >	
B) MINIMAMENTE ACEPTABLE	[54 - 81 >	
C) ACEPTABLE	[81 - 95 >	
D) CUMPLE LOS REQUISITOS	[95 - 122 >	110
E) EXCEDE LOS REQUISITOS	[122 - 135]	

Ficha de Evaluación de la Calidad del Producto Estándar Iso – 9126

INDICADORES	PUNTUACION				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.				X	
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.				X	
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas específicos.					X
Seguridad: referido a la capacidad del producto software para proteger la información y los datos					X
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad				X	
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.				X	
Tolerancia a Fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					X
Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					X
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones referidas a la fiabilidad.				X	
3. USUABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.				X	
Facilidad de Aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.				X	
Atracción: la capacidad del software para atraer al usuario.				X	
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					X
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle				X	
4. EFICIENCIA					
Comportamiento Temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.				X	
Utilización de Recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones					X
Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					X
5. MANTENIBILIDAD					
Analizabilidad: capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.				X	
Confiabilidad: capacidad del producto software de permitir implementar una modificación específica. La implementación incluye los cambios en el diseño, el código y la documentación				X	
Estabilidad: capacidad del producto software de evitar los defectos inesperados de las modificaciones.				X	
Facilidad de Prueba: capacidad del producto software de permitir validar las partes modificadas.				X	
Conformidad: capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.				X	
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado					X
Facilidad de Instalación: la capacidad del producto software para ser instalado en un ambiente determinado.				X	
Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.				X	
Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					X
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					X
SUB TOTALES				6	5
TOTAL				8	0
	118				

Escala Valorativa (Escala De Likert)

INDICADOR CUALITATIVO	VALOR
DEFICIENTE	1
MALO	2
REGULAR	3
BUENO	4
MUY BUENO	5

Cuadro de Decisión Iso 9126

CLASIFICACION	INTERVALO	DECISION
A) INACEPTABLE	[27 - 54 >	
B) MINIMAMENTE ACEPTABLE	[54 - 81 >	
C) ACEPTABLE	[81 - 95 >	
D) CUMPLE LOS REQUISITOS	[95 - 122 >	118
E) EXCEDE LOS REQUISITOS	[122 - 135]	

Ficha De Evaluación De La Calidad Del Producto Estándar Iso – 9126

INDICADORES	PUNTUACION				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					X
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					X
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas específicos.				X	
Seguridad: referido a la capacidad del producto software para proteger la información y los datos			X		
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad				X	
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.				X	
Tolerancia a Fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					X
Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					X
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones referidas a la fiabilidad.				X	
3. USUABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.				X	
Facilidad de Aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.				X	
Atracción: la capacidad del software para atraer al usuario.			X		
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.				X	
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle				X	
4. EFICIENCIA					
Comportamiento Temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.				X	
Utilización de Recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones					X
Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.				X	
5. MANTENIBILIDAD					
Analizabilidad: capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					X
Confiabilidad: capacidad del producto software de permitir implementar una modificación específica. La implementación incluye los cambios en el diseño, el código y la documentación					X
Estabilidad: capacidad del producto software de evitar los defectos inesperados de las modificaciones.					X
Facilidad de Prueba: capacidad del producto software de permitir validar las partes modificadas.					X
Conformidad: capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.				X	
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado			X		
Facilidad de Instalación: la capacidad del producto software para ser instalado en un ambiente determinado.				X	
Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.				X	
Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.				X	
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					X
SUB TOTALES			9	5 6	50
TOTAL	115				

Escala Valorativa (Escala De Likert)

INDICADOR CUALITATIVO	VALOR
DEFICIENTE	1
MALO	2
REGULAR	3
BUENO	4
MUY BUENO	5

Cuadro de Decisión Iso 9126

CLASIFICACION	INTERVALO	DECISION
A) INACEPTABLE	[27 - 54 >	
B) MINIMAMENTE ACEPTABLE	[54 - 81 >	
C) ACEPTABLE	[81 - 95 >	
D) CUMPLE LOS REQUISITOS	[95 - 122 >	115
E) EXCEDE LOS REQUISITOS	[122 - 135]	

RESULTADOS

Cuadro de decisiones ISO 9126

CLASIFICACION	INTERVALO	DECISION
A) INACEPTABLE	[27 - 54 >	
B) MINIMAMENTE ACEPTABLE	[54 - 81 >	
C) ACEPTABLE	[81 - 95 >	
D) CUMPLE LOS REQUISITOS	[95 - 122 >	114
E) EXCEDE LOS REQUISITOS	[122 - 135]	

CONCLUSIÓN: De acuerdo a los resultados de la validación de software se concluye que el sistema de ventas cumple con una media de 114 puntos según la escala de calificación del ISO – 9126.

ANEXO: Resultados Para la Aplicación del Pre-Test En El Sistema De

Venta

91

Evaluación de la Micro y Pequeña Empresa "San Santiago"

¿sabes que es un Sistema Informático?

Cuadro de frecuencia sobre conocimiento de un Sistema Informático.

		Conteo	Conteo %
P1	NO	1	10,0%
	SI	9	90,0%
	Total	10	

De los 10 encuestados 9 de ellos si tienes conocimiento sobre un sistema informático y 1 de ellos no tiene conocimiento sobre un sistema informático.

¿Cuánto es el ingreso diario en la micro y pequeña empresa "San Santiago"?

Cuadro promedio sobre ingreso diario.

Statistics

P4

N	Valida	10
	Missing	0
Mean		830,00

El promedio de ingreso diario con un sistema manual es de 830.00 soles

Cuadro sobre ingreso diario en la micro y pequeña empresa.

P4

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	500	1	10,0	10,0
	800	6	60,0	70,0
	1000	3	30,0	100,0
	Total	10	100,0	100,0

De los 10 encuestados 6 de ellos opinaron que ingreso diario con un sistema manual es de 800 soles, 3 de ellos 1000 soles y 1 de ellos 500 soles.

¿ha utilizado alguna vez un Sistema de Ventas?

Cuadro sobre si utilizo un Sistema de Ventas anteriormente.

		Conteo	Conteo %
P5	NO	7	70,0%
	SI	3	30,0%
	Total	10	100,0%

¿Cuánto tiempo cree usted que tarda en brindar Información de sus reportes de ventas con un Sistema manual?

Cuadro promedio sobre tiempo en brindar información de sus reportes de ventas con un sistema manual

Statistics

P7

N	Valid	10
	Missing	0
Mean		9,30

El promedio en brindar información con un sistema manual es de 8.30 minutos

Cuadro sobre tiempo en brindar información de sus reportes de ventas con un sistema manual.

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	5	30,0	30,0	30,0
	8	10,0	10,0	40,0
	10	40,0	40,0	80,0
	15	20,0	20,0	100,0
	Total	10	100,0	100,0

De los 10 encuestados 4 de ellos demoraron 10 minutos en brindar información de sus reportes, 3 de ellos 5 minutos, 2 de ellos 15 minutos y 1 que demora 8 minutos.

ANEXO: Evaluación del Sistema de Ventas

¿Cuánto tiempo cree usted que tarda en brindar información de sus reportes de ventas?

Cuadro promedio sobre el tiempo que tarda en brindar información de sus reportes de ventas con el Sistema.

Statistics

Pa

N	Valid	10
	Missing	0
Mean		1,150

El tiempo promedio que demoran en brindar información de sus reportes una vez implantado el sistema es de 1.150 minutos.

Cuadro sobre tiempo que tardan en brindar información de sus reportes con el Sistema.

Pa

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1,0	8	80,0	80,0
	1,5	1	10,0	90,0
	2,0	1	10,0	100,0
	Total	10	100,0	100,0

De los 10 encuestados una vez implantado el sistema de ventas 8 de ellos demoran 1 minuto en dar sus reportes, 1 de ellos demora 2 y uno de ellos demora 1.5 minutos.

¿con el sistema de ventas implantado se mejoró el proceso de ventas?

Cuadro sobre calidad del sistema.

	Conteo	Conteo %
Pc	NO	0
	SI	10
	Total	10

De los 10 encuestados 10 opinaron que si se mejoró el proceso de ventas.

¿con el sistema de ventas instalado mejoro sus reportes de ventas?

Cuadro sobre el mejoramiento de los reportes de venta.

		Conteo	Conteo %
Pd	NO	0	0,0%
	SI	10	100,0%
	Total	10	100,0%

De los 10 encuestados 10 opinaron que si mejoro sus reportes de ventas

¿con el sistema de ventas implementado ayudo a mejorar el control de stock de quesos?

Cuadro sobre el mejoramiento de control de stock de quesos.

		Conteo	Conteo %
Pe	NO	0	0,0%
	SI	10	100,0%
	Total	10	100,0%

¿el sistema de ventas es fácil de utilizar o cubrir sus expectativas?

Cuadro sobre la satisfacción del sistema

		Conteo	Conteo %
Pf	NO	0	0,0%
	SI	10	100,0%
	Total	10	100,0%

¿Cuánto es el ingreso diario en la micro y pequeña empresa “San Santiago”?

Cuadro promedio sobre el ingreso diario con el sistema.

Statistics

Pg

N	Valid	10
	Missing	0
Mean		1450,00

El promedio de ingreso diario una vez implantado el sistema es de 1450.00
nuevo soles

Cuadro sobre el ingreso diario con el Sistema

Pg

	Frequency	Percent	Valid Percent	Cumulative Percent
1000	1	10,0	10,0	10,0
1400	2	20,0	20,0	30,0
Valid 1500	5	50,0	50,0	80,0
1600	2	20,0	20,0	100,0
Total	10	100,0	100,0	

De los 10 encuestados 5 de ellos opinaron que una vez implantado el sistema de ventas el ingreso diario era de 1500, 2 de ellos 1400, 2 de ellos 1600 y 1 de ellos 1000.

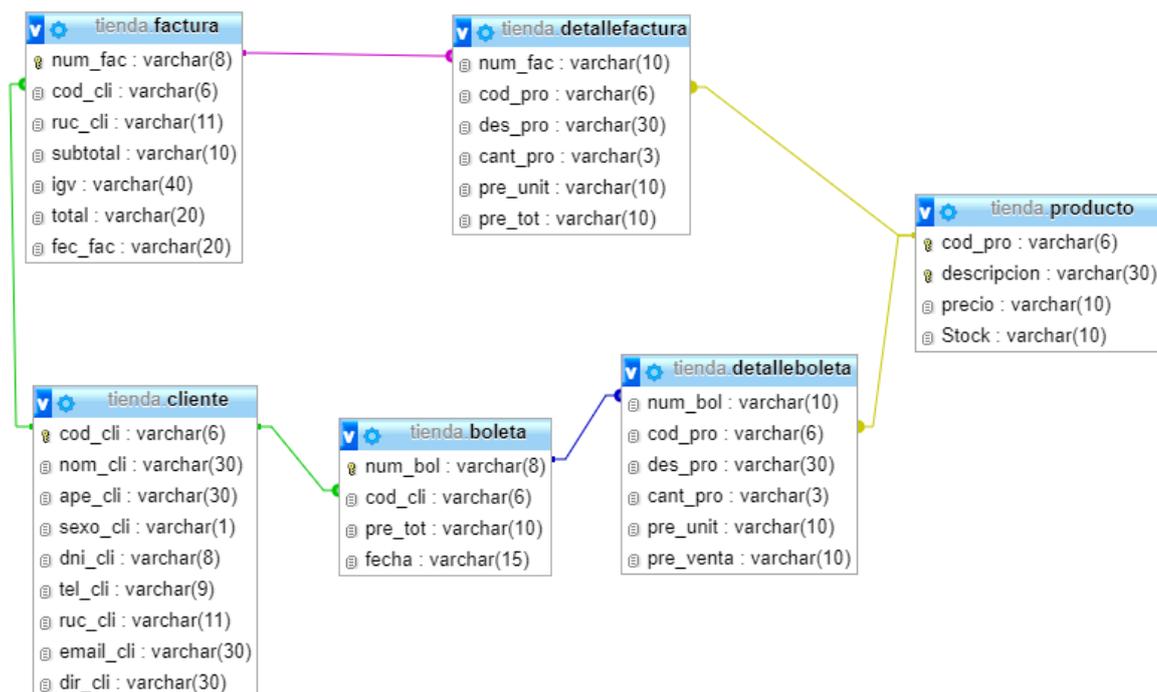
ANEXO: Formulario del Sistema de Ventas de la Micro y Pequeña

Empresa San Santiago.

Formulario del Sistema de Ventas

-  Formulario
 -  Boleta.java
 -  ClientesB.java
 -  ConsultasBoletas1.java
 -  ConsultasClientes.java
 -  ConsultasFacturas1.java
 -  ConsultasProductos.java
 -  DetalleBoleta.java
 -  DetalleFactura.java
 -  Factura.java
 -  GenerarCodigos.java
 -  GenerarNumero.java
 -  IngresoCliente.java
 -  IngresoProductos.java
 -  IngresoSistema.java
 -  Principal.java
 -  Productos.java
 -  ProductosB.java
 -  RecuperarContraseña.java
 -  clientes.java
 -  reportesBoletas.jasper
 -  reportesBoletas.jrxml
 -  reportesClientes.jasper
 -  reportesClientes.jrxml
 -  reportesFacturas.jasper
 -  reportesFacturas.jrxml
 -  reportesProductos.jasper
 -  reportesProductos.jrxml

ANEXO: Diagrama de Base de Datos



ANEXO: Código Fuente

***** SISTEMA DE VENTAS – JAVA NETBEANS *****

FORMULARIO PRINCIPAL

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * Principal.java
 *
 * Created on 05-abr-2017, 20:15:45
 */
package Formulario;

import Formulario.IngresoSistema;
import claseConectar.conectar;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.view.JasperViewer;

/**
 *
 * @author OLIVERPC
 */
public class Principal extends javax.swing.JFrame {

    /** Creates new form Principal */
    public Principal() {
        initComponents();
        this.setLocation(200, 50);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jdpecritorio = new javax.swing.JDesktopPane();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
    }

```

```

jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenuItem3 = new javax.swing.JMenuItem();
jMenuItem4 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
jMenuItem5 = new javax.swing.JMenuItem();
jMenuItem10 = new javax.swing.JMenuItem();
jMenu4 = new javax.swing.JMenu();
jMenuItem7 = new javax.swing.JMenuItem();
jMenuItem8 = new javax.swing.JMenuItem();
jMenuItem9 = new javax.swing.JMenuItem();
jMenuItem11 = new javax.swing.JMenuItem();
jMenu5 = new javax.swing.JMenu();
jMenuItem6 = new javax.swing.JMenuItem();
jMenuItem12 = new javax.swing.JMenuItem();
jMenuItem13 = new javax.swing.JMenuItem();
jMenuItem14 = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Sistema de Ventas");

jdpescritorio.setBackground(new java.awt.Color(25, 153, 153));

jMenu1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagenes/archivo.jpg"))); //
NOI18N
jMenu1.setText("Archivo");
jMenu1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});

jMenuItem1.setText("Cerrar Sesion");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem1);

jMenuItem2.setText("Salir");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem2);

```

```
jMenuBar1.add(jMenu1);

jMenu2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagenes/mantenimiento.jpg"
))); // NOI18N
jMenu2.setText("Mantenimiento");
jMenu2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenu2ActionPerformed(evt);
    }
});

jMenuItem3.setText("Productos");
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem3);

jMenuItem4.setText("Clientes");
jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem4ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem4);

jMenuBar1.add(jMenu2);

jMenu3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagenes/movimiento.JPG")))
; // NOI18N
jMenu3.setText("Movimiento");
jMenu3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenu3ActionPerformed(evt);
    }
});

jMenuItem5.setText("Factura");
jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem5);

jMenuItem10.setText("Boleta");
```

```
jMenuItem10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem10ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem10);

jMenuBar1.add(jMenu3);

jMenu4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagenes/consultas.JPG")));
// NOI18N
jMenu4.setText("Consultas");
jMenu4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenu4ActionPerformed(evt);
    }
});

jMenuItem7.setText("Clientes");
jMenuItem7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem7ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem7);

jMenuItem8.setText("Productos");
jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem8ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem8);

jMenuItem9.setText("Facturas");
jMenuItem9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem9ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem9);

jMenuItem11.setText("Boletas");
jMenuItem11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem11ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem11);
```

```

jMenuBar1.add(jMenu4);

jMenu5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagenes/reportes.JPG"))); //
NOI18N
jMenu5.setText("Reportes");

jMenuItem6.setText("Clientes");
jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem6ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem6);

jMenuItem12.setText("Productos");
jMenuItem12.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem12ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem12);

jMenuItem13.setText("Facturas");
jMenuItem13.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem13ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem13);

jMenuItem14.setText("Boletas");
jMenuItem14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem14ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem14);

jMenuBar1.add(jMenu5);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jdpescritorio,
javax.swing.GroupLayout.DEFAULT_SIZE, 750, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jdpescritorio,
javax.swing.GroupLayout.DEFAULT_SIZE, 503, Short.MAX_VALUE)
        .addGap())
    );

    pack();
} // </editor-fold>

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    IngresoSistema principal= new IngresoSistema();
    principal.setVisible(true);
    principal.pack();
    this.setVisible(false);
}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    this.dispose();
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    IngresoProductos ip= new IngresoProductos();
    jdpescritorio.add(ip);
    ip.show();
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    IngresoCliente cli= new IngresoCliente();
    jdpescritorio.add(cli);
    cli.show();
}

```

```

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    Factura fac= new Factura();
    jdpcritorio.add(fac);
    fac.show();
}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    try {
        conectar cc= new conectar();

        JasperReport
reportes=JasperCompileManager.compileReport("reportesClientes.jrxml");
        JasperPrint print=JasperFillManager.fillReport(reportes,
null,cc.conexion());
        JasperViewer.viewReport(print);

    } catch (Exception e) {
        System.out.printf(e.getMessage());
    }

}

private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    ConsultasClientes clientes= new ConsultasClientes();
    jdpcritorio.add(clientes);
    clientes.show();
}

private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    ConsultasProductos productos= new ConsultasProductos();
    jdpcritorio.add(productos);
    productos.show();
}

private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    ConsultasFacturas1 Facturas= new ConsultasFacturas1();
    jdpcritorio.add(Facturas);
}

```

```
Facturas.show();
}

private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    Boleta bol = new Boleta();
    jdpscriterio.add(bol);
    bol.show();
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}

private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    ConsultasBoletas1 Boletas= new ConsultasBoletas1();
    jdpscriterio.add(Boletas);
    Boletas.show();
}

private void jMenuItem12ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try {
        conectar cc= new conectar();

        JasperReport
reportes=JasperCompileManager.compileReport("reportesProductos.jrxml");
        JasperPrint print=JasperFillManager.fillReport(reportes,
null,cc.conexion());
        JasperViewer.viewReport(print);

    } catch (Exception e) {
        System.out.printf(e.getMessage());
    }
}

private void jMenuItem13ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try {
        conectar cc= new conectar();

        JasperReport
reportes=JasperCompileManager.compileReport("reportesFacturas.jrxml");
        JasperPrint print=JasperFillManager.fillReport(reportes,
null,cc.conexion());
        JasperViewer.viewReport(print);

    } catch (Exception e) {
        System.out.printf(e.getMessage());
    }
}
```

```

    }
}

private void jMenuItem14ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try {
        conectar cc= new conectar();

        } catch (Exception e) {
            System.out.printf(e.getMessage());
        }
    }

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
    }
}

```

```
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new Principal().setVisible(true);
    }
});
}
// Variables declaration - do not modify
private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenu jMenuItem3;
private javax.swing.JMenu jMenuItem4;
private javax.swing.JMenu jMenuItem5;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem10;
private javax.swing.JMenuItem jMenuItem11;
private javax.swing.JMenuItem jMenuItem12;
private javax.swing.JMenuItem jMenuItem13;
private javax.swing.JMenuItem jMenuItem14;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem5;
private javax.swing.JMenuItem jMenuItem6;
private javax.swing.JMenuItem jMenuItem7;
private javax.swing.JMenuItem jMenuItem8;
private javax.swing.JMenuItem jMenuItem9;
public static javax.swing.JDesktopPane jdpcritorio;
// End of variables declaration
}

***** FIN *****
```

ANEXO: Manual de Uso del Sistema de Venta

1. INGRESO AL SISTEMA

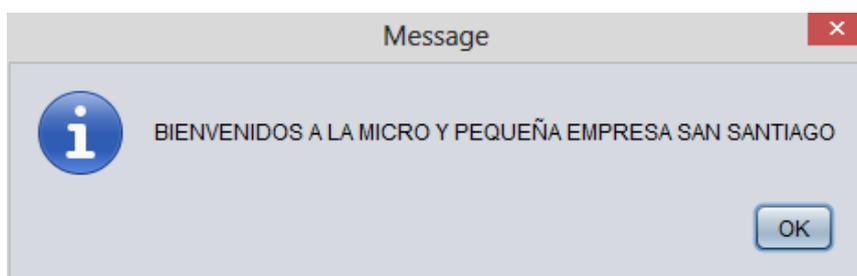


Para ingresar al sistema ejecutamos el programa, el cual nos mostrara un formulario llamado *Acceso al Sistema* el cual nos pedirá unas credenciales usuario y contraseña.

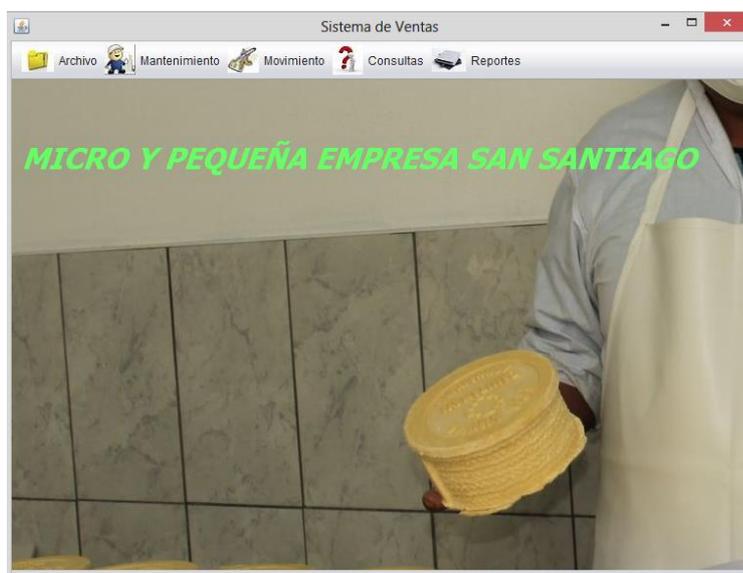


Para este caso utilizaremos las credenciales, usuario: olier , contraseña *****

Damos click en *entrar* y nos dara la bienvenida y damos ok



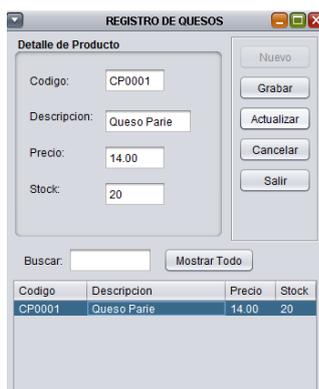
Seguidamente nos mostrara el formulario inicio o ventana principal.



Dentro de ella podemos encontrar varios iconos tales como:

- ✓ Archivo
- ✓ Mantenimiento
- ✓ Movimiento
- ✓ Consultas
- ✓ Resportes

Para registrar un producto damos click en *mantenimiento* -> *producto* en este formulario podemos ingresar un producto, modificar y buscar.



Codigo	Descripcion	Precio	Stock
CP0001	Queso Parle	14.00	20

Para realizar un factura de la misma manera engresamos en *movimiento* -> *factura* en este formulario nos permite exportar el cliente registrado de la misma base de dastos con un click, nos permite modificar y buscar y de la misma manera nos realiza el calculo con igv incluido.



Para realizar un reporte damos click en *reporte* -> aquí escogemos el tipo de reporte que queremos realizar y nos mostrara el reporte en un pdf.

