



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

SISTEMATIZACIÓN DEL PROCESO DE FACTURACIÓN DE ARTÍCULOS DE
LIMPIEZA MEDIANTE UN APLICATIVO WEB PARA LA EMPRESA
PROVEASEO UBICADO EN LA CIUDAD DE QUITO

Proyecto de investigación previo a la obtención del título de Tecnólogo en Análisis
de Sistemas

Autor: Vera Jaramillo Kevin Estuardo

Tutor: Prof. Luis Ríos C.

Quito, diciembre 2017



Declaración de Autoría del Estudiante

Declaro que la investigación es absolutamente original, auténtica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen derechos de autor vigentes. Las ideas, doctrinas, resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Kevin Estuardo Vera Jaramillo

1721067369



Licencia de Uso No Comercial

Yo, Vera Jaramillo Kevin Estuardo portador de la cédula de ciudadanía signada con el No. 1721067369 de conformidad con lo establecido en el Artículo 110 del Código de Economía Social de los Conocimientos, la Creatividad y la Innovación (INGENIOS) que dice: “En el caso de las obras creadas en los centros educativos, universidades, escuelas politécnicas, institutos superiores técnicos, tecnológicos, pedagógicos, de artes y los conservatorios superiores, e institutos públicos de investigación como resultado de su actividad académica o de investigación tales como trabajos de titulación , proyectos de investigación o innovación, artículos académicos, u otros análogos, sin perjuicio de que pueda existir relación de dependencia, la titularidad de los derechos patrimoniales corresponderá a los autores. Sin embargo, el establecimiento tendrá una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra con fines académicos. Sin perjuicio de los derechos reconocidos en el párrafo precedente, el establecimiento podrá realizar un uso comercial de la obra previa autorización a los titulares y notificación a los autores en caso de que se traten de distintas personas. En cuyo caso corresponderá a los autores un porcentaje no inferior al cuarenta por ciento de los beneficios económicos resultantes de esta explotación. El mismo beneficio se aplicará a los autores que hayan transferido sus derechos a instituciones de educación superior o centros educativos.”, otorgo licencia gratuita, intransferible y no exclusiva para el uso no comercial del proyecto denominado Sistematización del proceso de facturación de artículos de limpieza mediante un aplicativo web para la empresa Proveaseo ubicado en la ciudad de Quito con fines académicos al Instituto

Tecnológico Superior Cordillera.

SISTEMATIZACIÓN DEL PROCESO DE FACTURACIÓN DE ARTÍCULOS DE LIMPIEZA MEDIANTE UN APLICATIVO WEB PARA LA EMPRESA PROVEASEO UBICADO EN LA CIUDAD DE QUITO



FIRMA _____

NOMBRE Kevin Estuardo Vera Jaramillo

CEDULA CC 172106736-9

Quito, a los 12 de diciembre de 2017



Agradecimiento

Quiero agradecer a Dios, porque me dio la salud, fuerza y sabiduría para tomar decisiones correctas que me han permitido culminar esta etapa de mi vida.

Quiero agradecer a mi esposa e hija, son la parte más importante en mi vida y su apoyo ha sido incondicional, mi razón de salir adelante.

Agradezco al Instituto Tecnológico Superior Cordillera por su acogida en la Carrera de Análisis de Sistemas, a mis docentes que me supieron formar como profesional con valores éticos y morales.

Gracias.



Dedicatoria

Este trabajo va dedicado hacia mis familiares, en especial a mi suegra, por todas las facilidades y el apoyo que supo brindar a mi familia, sin esperar nada a cambio. Su dedicación y apoyo me permitió continuar mis estudios sin descuidar mis responsabilidades.

Kevin Vera Jaramillo.



Índice General

Contenido	Página
Declaración de Autoría del Estudiante.....	i
Licencia de Uso No Comercial	ii
Agradecimiento.....	iv
Dedicatoria	v
Índice General	vi
Índice de Tablas	x
Índice de Figuras	xii
Índice de Anexos.....	xv
Resumen Ejecutivo	xvi
Abstract	xvii
Introducción	xviii
Capítulo I: Antecedentes	1
1.01 Contexto	1
1.02 Justificación.....	2
1.03 Definición del Problema Central	3
Capítulo II: Análisis de Involucrados	5
2.01 Requerimientos.....	5

SISTEMATIZACIÓN DEL PROCESO DE FACTURACIÓN DE ARTÍCULOS DE LIMPIEZA
MEDIANTE UN APLICATIVO WEB PARA LA EMPRESA PROVEASEO UBICADO EN LA
CIUDAD DE QUITO



2.01.01 Descripción del sistema actual	5
2.01.02 Visión y Alcance	6
2.01.03 Entrevistas	7
2.01.04 Requerimientos.....	8
2.01.05 Descripción detallada de requerimientos	10
2.02 Mapeo de Involucrados	17
2.03 Matriz de Involucrados.....	18
Capítulo III: Problemas y Objetivos	19
3.01 Árbol de Problemas	19
3.02 Árbol de Objetivos	20
3.03 Diagramas de Casos de Uso	21
3.04 Especificación de Casos de Uso	26
3.05 Casos de uso de Realización	31
3.06 Diagrama de Secuencias.....	33
Capítulo IV: Análisis de Alternativas	36
4.0.1 Matriz de Análisis de Alternativas	36
4.0.2 Análisis de la Matriz de Alternativas	37
4.0.3 Matriz de Impacto de Objetivos	37
4.0.4 Análisis de la matriz de impacto de objetivos	37
4.0.5 Estándares para el Diseño de Clases	37
4.0.6 Diagrama de Clases	42



4.0.7 Modelo Físico	42
4.0.8 Diagrama de Estrategia	43
4.0.9 Matriz de marco lógico.....	44
4.0.10 Vistas Arquitectónicas.....	44
4.01.01 Vista Lógica	44
4.01.02 Vista física del diseño de la solución	45
4.01.03 Vista de desarrollo	45
4.01.04 Vista de Procesos	46
Capítulo V: Propuesta	47
5.01 Especificación de estándares de programación	47
5.01.01 Nomenclatura .NET Objetos.....	47
5.01.02 Nomenclatura .NET Clases, métodos y variables.....	47
5.01.03 Nomenclatura Base de Datos	48
5.02 Diseño de interfaces de usuario.....	50
5.03 Especificación de pruebas de unidad.....	57
5.04 Especificación de pruebas de carga.....	62
5.05 Configuración del ambiente mínima/ideal	63
Capítulo VI: Aspectos Administrativos	64
6.01 Recursos	64
6.01.01 Recurso Humano.....	64
6.01.02 Recurso Material	65



6.01.03 Recurso Tecnológico.....	65
6.02 Presupuesto.....	66
6.03 Cronograma.....	66
Capítulo VII: Conclusiones y Recomendaciones.....	67
7.01 Conclusiones.....	67
7.02 Recomendaciones.....	68
ANEXOS.....	69
MANUAL DE INSTALACIÓN.....	74
MANUAL DE USUARIO.....	81
MANUAL TÉCNICO.....	94
BIBLIOGRAFIA.....	169



Índice de Tablas

Tabla 1 <i>Matriz de fuerzas T</i>	3
Tabla 2 <i>Entrevista a Proveaseo</i>	7
Tabla 3 <i>Matriz de requerimientos</i>	8
Tabla 4 <i>Matriz de requerimientos no funcionales</i>	9
Tabla 5 <i>Descripción detallada del requerimiento funcional 001</i>	10
Tabla 6 <i>Descripción detallada del requerimiento funcional 002</i>	11
Tabla 7 <i>Descripción detallada del requerimiento funcional 003</i>	12
Tabla 8 <i>Descripción detallada del Requerimiento Funcional 004</i>	13
Tabla 9 <i>Descripción detallada del Requerimiento Funcional 005</i>	14
Tabla 10 <i>Descripción detallada del requerimiento no funcional 001</i>	15
Tabla 11 <i>Descripción detallada del requerimiento no funcional 002</i>	16
Tabla 12 <i>Especificación Caso de Uso Registrar Usuario</i>	26
Tabla 14 <i>Especificación Caso de Uso Registrar Orden Compra</i>	26
Tabla 15 <i>Especificación Caso de Uso Registrar Producto</i>	27
Tabla 15 <i>Especificación Caso de Uso Registrar Tipo de Producto</i>	28
Tabla 16 <i>Especificación Caso de Uso Registrar Tipo de usuario</i>	28
Tabla 17 <i>Especificación Caso de Uso Registrar Venta</i>	29
Tabla 18 <i>Listado de Cliente</i>	29
Tabla 21 <i>Especificación Caso de Uso Reporte de Orden de Compra</i>	30
Tabla 22 <i>Reporte de Ventas</i>	30
Tabla 23 <i>Matriz de Análisis de Alternativas</i>	36
Tabla 24 <i>Especificación de estándares .NET Objetos</i>	47
Tabla 25 <i>Especificación de estándares.NET Clases</i>	47
Tabla 26 <i>Especificación de estándares BD Tipo de dato</i>	48



Tabla 27 *Especificación de estándares BD Tablas, Campos, Funciones y*

<i>Procedimientos Almacenados</i>	49
Tabla 28 <i>PU001 Ingreso al Sistema</i>	57
Tabla 29 <i>PU002 Registro de Usuarios</i>	57
Tabla 30 <i>PU003 Registro de Productos</i>	58
Tabla 31 <i>PU004 Registro de Marca de producto</i>	58
Tabla 32 <i>PU005 Registro de Categorías de productos</i>	59
Tabla 33 <i>Registro de proveedores</i>	59
Tabla 34 <i>PU007 Generar Orden de Compra</i>	60
Tabla 35 <i>PU008 Generar Factura</i>	60
Tabla 36 <i>Listar Orden de Compra PU009</i>	61
Tabla 37 <i>PU010 Listar Factura</i>	61
Tabla 38 <i>PC001 Facturación</i>	62
Tabla 39 <i>Subida de información</i>	62
Tabla 40 <i>Recurso Humano</i>	64
Tabla 41 <i>Recurso Material</i>	65
Tabla 42 <i>Presupuesto</i>	66



Índice de Figuras

<i>Figura 1 Mapeo de Involucrados</i>	17
<i>Figura 2 Árbol de Problemas</i>	19
<i>Figura 3 Árbol de Objetivos.</i>	20
<i>Figura 4 Caso de Uso General Proceso de Facturación</i>	21
<i>Figura 5 Caso de Uso Registrar Empleado</i>	21
<i>Figura 6 Caso de Uso Registrar Marca de producto</i>	21
<i>Figura 7 Caso de Uso Registrar Orden de Compra</i>	22
<i>Figura 8 Caso de Uso Registrar Pedido</i>	22
<i>Figura 9 Caso de Uso Registrar Producto</i>	23
<i>Figura 10 Caso de Uso Registrar Proveedor</i>	23
<i>Figura 11 Caso de Uso Registrar tipo de producto</i>	23
<i>Figura 12 Caso de Uso Registrar Usuario</i>	24
<i>Figura 13 Caso de Uso Registrar Venta</i>	24
<i>Figura 14 Caso de Uso Listar Cliente</i>	24
<i>Figura 15 Caso de Uso Listar Producto</i>	25
<i>Figura 16 Caso de Uso Listar Proveedor</i>	25
<i>Figura 17 Reporte Orden de Compra</i>	25
<i>Figura 18 Reporte de Venta</i>	25
<i>Figura 20 Caso de Uso de realización Ingreso al aplicativo</i>	31
<i>Figura 21 Caso de Uso de realización Gestión de clientes</i>	31
<i>Figura 22 Caso de uso de realización Generar Venta</i>	32
<i>Figura 23 Caso de uso de realización Emitir Reporte</i>	32
<i>Figura 24 Diagrama de Secuencia Ingreso al aplicativo</i>	33
<i>Figura 25 Diagrama de Secuencia Ingreso de información</i>	33



<i>Figura 26 Diagrama de Secuencias Ingreso de Clientes</i>	34
<i>Figura 27 Diagrama de Secuencias Generar Factura</i>	34
<i>Figura 28 Diagrama de Secuencia Generar Reporte</i>	35
<i>Figura 29 Representación de una clase</i>	38
<i>Figura 30 Representación de la relación de herencia</i>	39
<i>Figura 31 Representación de asociaciones en general</i>	40
<i>Figura 32 Representación de relación de asociación</i>	40
<i>Figura 33 Representación de relación de dependencia</i>	41
<i>Figura 34 Diagrama de Clases</i>	42
<i>Figura 35 Modelo Físico de la Base de Datos</i>	42
<i>Figura 36 Diagrama de Estrategias</i>	43
<i>Figura 37 Vista Lógica de la Base de Datos</i>	44
<i>Figura 38 Vista física del diseño de la solución</i>	45
<i>Figura 39 Vista de Desarrollo.</i>	45
<i>Figura 40 Vista de Procesos del diseño de la solución</i>	46
<i>Figura 41 Ingreso al aplicativo.</i>	50
<i>Figura 42 Pantalla de inicio</i>	51
<i>Figura 43 Mantenimiento Usuarios</i>	51
<i>Figura 44 Mantenimiento Productos</i>	52
<i>Figura 45 Mantenimiento Marcas Productos.</i>	52
<i>Figura 46 Mantenimiento Categorías Productos.</i>	53
<i>Figura 47 Mantenimiento de Proveedores.</i>	53
<i>Figura 48 Generar Orden de Compra.</i>	54
<i>Figura 49 Listado de Órdenes de Compra.</i>	54
<i>Figura 50 Generar Factura</i>	55



<i>Figura 51 Reporte de Stock de Productos</i>	<i>55</i>
<i>Figura 52 Reporte de Facturas.....</i>	<i>56</i>
<i>Figura 53 Diccionario de Datos: Tabla Garantía.....</i>	<i>96</i>
<i>Figura 54 Diccionario de Datos: Tabla Marca_Producto.....</i>	<i>96</i>
<i>Figura 55 Diccionario de Datos: Tabla Orden_Compra.....</i>	<i>96</i>
<i>Figura 56 Diccionario de Datos: Tabla Producto</i>	<i>96</i>
<i>Figura 57 Diccionario de Datos: Tabla Proveedor</i>	<i>97</i>
<i>Figura 58 Diccionario de Datos: Tabla Tipo_Documento</i>	<i>97</i>
<i>Figura 59 Diccionario de Datos: Tabla Tipo_Producto</i>	<i>97</i>
<i>Figura 60 Diccionario de Datos: Tabla Tipo_Usuario.....</i>	<i>97</i>
<i>Figura 61 Diccionario de Datos: Tabla Venta</i>	<i>98</i>
<i>Figura 62 Diccionario de Datos: Tabla Usuario</i>	<i>98</i>
<i>Figura 63 Diccionario de Datos: Tabla Detalle_Orden_Compra</i>	<i>98</i>
<i>Figura 64 Diccionario de Datos: Tabla Detalle_Venta</i>	<i>98</i>
<i>Figura 65 Diccionario de Datos: Primary Key</i>	<i>99</i>
<i>Figura 66 Diccionario de Datos: Foreign Key.....</i>	<i>99</i>



Índice de Anexos

Anexo A.0 1 Matriz de Involucrados	70
Anexo A.0 2 Matriz de Impacto de Objetivos	71
Anexo A.0 3 Matriz de Marco Lógico	72
Anexo A.0 4 Cronograma	73



Resumen Ejecutivo

El presente proyecto tiene como finalidad sistematizar los procesos de facturación de la empresa Proveaseo ubicado en la ciudad de Quito, su actividad principal es proveer de artículos de limpieza hacia pymes y clientes en general, para lo cual es fundamental que sus procesos se automaticen de manera que se pueda cumplir con los requerimientos de los usuarios.

La empresa se encuentra en la necesidad de implementar un aplicativo que controle los procesos de manera automatizada, los procesos manuales que se han venido manejando han llevado a problemáticas como inseguridad, redundancia de información, ordenes, facturas llenadas de manera errónea, falta de control de ingreso y egreso de artículos, entre otros, para este propósito, el desarrollo del software contribuye a una etapa de innovación tecnológica hacia la empresa , implementando un módulo de facturación intuitivo y seguro para el usuario.

El objetivo es cumplir con los requerimientos del cliente de acuerdo a sus necesidades, la sistematización de los procesos beneficia a la empresa en toma de decisiones y aumenta su competitividad con el mercado actual.



Abstract

This Project has been developed to systematize the billing processes for Proveaseo located in the city of Quito, whose main activity is to provide cleaning products to SMEs and customers in general, therefore it's essential that their processes are automated in a way that helps meet the requirements of users.

The company is in need of implementing an application that controls the processes in an automated way, the manual processes that have been handled have led to problems such as insecurity, redundancy of information, orders, invoices filled incorrectly, lack of control income and expenses articles, among others, for this purpose, software development contributes to a stage of technological innovation towards the company, implementing an intuitive and safe billing module for the user.

The objective is to meet the requirements of the client according to their needs; the systematization of the processes benefits the company in decision-making and increases the competitiveness with the current market.



Introducción

En la actualidad, pequeñas y medianas empresas del Ecuador dependen de las tecnologías de información y comunicación (TIC), independientemente del rol del negocio, cada empresa busca la manera de posicionarse en el mercado, los usos de herramientas tecnológicas son utilizados diariamente y por ello la implementación de un sistema informático permite automatizar procesos, generando controles e información que permite a los directivos toma de decisiones y resolución de problemas.

Para la elaboración de un sistema informático no depende solo del conocimiento de la persona encargada, el objetivo es cumplir con las necesidades que la empresa requiere y comprender claramente los procesos que intervienen en el mismo.

En este proyecto se busca sistematizar del proceso de facturación de la empresa Proveaseo, los cuales se han venido desarrollando de manera manual, su problemática radica en: inseguridades, redundancia de información, pérdidas de productos, entre otros. La empresa reconoce la necesidad de implementar un sistema informático que optimice el proceso de facturación, la solución que se plantea permitirá la buena interacción con la información requerida del personal involucrado y directivos de forma organizada, esto con el objetivo de satisfacer las necesidades del cliente.

Capítulo I: Antecedentes

1.01 Contexto

En la actualidad, el cambio de la matriz productiva en el Ecuador implica el desarrollo económico y social a nivel de pequeñas y grandes empresas, con el objetivo de que se pueda generar mayor rentabilidad y sustento a la economía actual impulsando actividades a nivel tecnológico, social y humano, este cambio busca implementar los mecanismos necesarios para prestar mejor calidad de servicio en los diferentes campos de desarrollo, debido al incremento de la competitividad se busca encontrar una propuesta viable para optimizar los diferentes procesos en las organizaciones, es una gran alternativa la implementación de herramientas tecnológicas que permitan a las empresas adaptarse rápidamente a las necesidades del entorno, de tal manera que puedan permanecer vigentes en sus actividades empresariales.

Proveaseo es una empresa ubicada en la ciudad de Quito sector Carcelén, se dedica a la comercialización de productos de aseo, su crecimiento ha sido evidente y por ende demanda mayor control en sus productos, el personal a cargo no tiene un control necesario para evitar errores que perjudicarían económicamente con la pérdida de productos u ventas, con el fin de mantenerse en la competencia en cuanto a comercialización de productos de aseo, tiene la necesidad de implementar un sistema de facturación lo cual permitirá innovar los procesos permitiendo el crecimiento y optimización de tiempo en el desarrollo del comercio y su fortalecimiento para ser más competitivos.



1.02 Justificación

El aplicativo se plantea como propuesta con todos los requerimientos y necesidades del cliente, además de contar con los aspectos técnicos necesarios.

La idea de contar con un software que agilice los procesos y automatice de tal manera que beneficie a los empleados de ProveAseo, realizando un mejor desempeño hacia sus servicios y que la empresa se adapte al uso de las tecnologías de la información, llevando así mejor control, optimizar el tiempo y satisfacer de mejor manera las necesidades de nuestros clientes.

Como beneficios a considerar son:

- Permitirá un acceso completo a la información específica y exclusiva de cada cliente.
- El sistema permitirá llevar a cabo el registro de todas las ventas hechas.
- Mejora el proceso contable de la empresa.
- Controlara los ingresos y egresos para poder contabilizar las ganancias de la empresa.
- Reducirá costos en suministros de oficina
- Agilizará el proceso de atención al cliente
- Utilizará tecnología de punta para controlar los procesos de gestión de la empresa.

1.03 Definición del Problema Central

Tabla 1

Matriz de fuerzas T

ANÁLISIS DE FUERZAS

Situación empeorada	Situación actual				Situación Mejorada
Perdida de información	Inexistencia de un proceso de facturación automatizado				Información integra confiable y con procesos ágiles
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Generación de facturas de manera correcta	2	4	4	2	Errores al ingresar datos manualmente
Automatización y control de los artículos en Stock	1	4	3	4	Falta de conocimiento para ordenar correctamente los archivos
Reducir tiempos de emisión de facturas	3	5	4	3	Falta de eficiencia al momento de facturar
Control de registros documentos y reportes	1	4	3	1	Al finalizar la jornada no se tiene reporte de ventas.

En la matriz entre los problemas identificados y con más importancia, se ha considerado lo siguiente:

- Falta de un sistema que controle y maneje la información de cada cliente en ventas y mantenimiento.
- Pérdida de tiempo al llevar a cabo los procesos de registro de cada cliente.
- Pérdida de tiempo para renovar el stock de productos.



-Falta de una interfaz que facilite el trabajo del usuario.

En base a estos problemas se propone:

Desarrollar un sistema de control de registro de datos y facturación.

-Optimizar el proceso de distribución de nuestros productos a todos nuestros clientes

-Implementar una interfaz que facilite el trabajo de todos los usuarios.

Capítulo II: Análisis de Involucrados

2.01 Requerimientos

2.01.01 Descripción del sistema actual

Proveaseo es una empresa que se dedica a la venta de artículos de limpieza, para cumplir con la regla del negocio, los artículos son adquiridos a través de los proveedores, sus procesos principales son: pedidos a proveedores, ventas e inventario. Los artículos son adquiridos presencialmente en una distribuidora mediante una compra, esta factura es emitida a nombre de la empresa, este proceso se realiza cuando los artículos no son suficientes para cubrir algún pedido de cliente o para mantener un stock, estos documentos son archivados de forma manual en una bitácora, el problema radica cuando al final de cada mes se contabiliza estos documentos, en muchas ocasiones no coincide con el stock actual, adicional que no posee ningún tipo de seguridad en cuanto a la integridad de los mismos y, esto permite que pueda existir daños u alteraciones y además de no contar con un respaldo de contingencia en casos de amenazas; como empresa legal conformada, Proveaseo realiza la emisión de facturas a sus clientes por las ventas realizadas de manera manual, el cliente solicita vía telefónica los productos que requiere, el personal emite la factura y no es registrado de forma correcta, generando inconvenientes al momento de la entrega y pérdida de tiempo, por lo cual el proceso no es atendido de una manera automática, la necesidad de contar con un aplicativo que permita realizar este tipo de procesos de manera rápida y segura, evitando así la falta de organización y errores que puedan existir.



2.01.02 Visión y Alcance

Visión

Desarrollar un sistema de facturación que brinde a la empresa ProveAseo mayor agilidad en su proceso actual de facturación, facilitando así el acceso a la información de manera confiable y segura.

Alcance

Se pretende solucionar una problemática que radica en cuanto a mejorar el servicio al cliente y la calidad de procesos actuales permitiendo llevar una mayor organización que permita una correcta toma de decisiones por parte de gerencia, adicional se pretende impulsar a las pequeñas empresas tomen intereses en lo que sistematización de procesos se refiere.



2.01.03 Entrevistas

Tabla 2

Entrevista a Proveaseo

**Entrevista: Propietario del
negocio**

Elaborado por: Kevin Vera

Identificador: 001

PREGUNTAS	OBJETIVOS	ANÁLISIS POSTERIOR
¿Qué dificultades presenta en el proceso de facturación?	Determinar las falencias que presentan los procesos actuales	Ejecutar validaciones para evitar ingresos de manera errónea.
¿Conoce a detalle los procesos que realizan los usuarios?	Determinar la existencia de control en el proceso de facturación	Los usuarios no presentan un manual de procesos adecuado.
¿Tiene información completa acerca de las transacciones realizadas?	Identificar los procesos de seguimiento hacia las transacciones	Los seguimientos no se encuentran automatizados y son requeridos en contacto con el usuario
¿La información que solicita de sus usuarios es confiable y a tiempo real?	Determinar si el usuario opta por migrar a nuevas tecnologías	Determinar si el usuario es adaptable hacia una automatización de procesos
¿Los reportes que realiza llevan de manera concisa los datos?	Determinar la integridad de los datos en el proceso de facturación	Guardar información clasificada y evitar la utilización de editores.



2.01.04 Requerimientos

Tabla 3

Matriz de requerimientos

MATRIZ DE REQUERIMIENTOS						
Identificador	Descripción	Fuentes	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS FUNCIONALES						
RF001	El sistema deberá permitir el ingreso al sistema solo por personal autorizado	Gerente	Alta	Sistema	Valido	General
RF002	El sistema deberá permitir el ingreso de información de productos en el sistema	Gerente, empleados	Alta	Sistema	Valido	Empleados
RF003	El sistema deberá tener un inventario actualizado de los productos	Empleados	Alta	Sistema	Valido	General
RF004	El sistema deberá permitir la emisión de facturas y órdenes de compra	Empleados	Alta	Sistema	Valido	General
RF005	El sistema deberá generar reportes de productos y facturas.	Empleados	Alta	Sistema	Valido	General



Tabla 4

Matriz de requerimientos no funcionales

MATRIZ DE REQUERIMIENTOS NO FUNCIONALES						
Identificador	Descripción	Fuentes	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS NO FUNCIONALES						
RNF001	El sistema deberá ser funcional en cualquier tipo de navegador	Gerente	Alta	Sistema	En producción	General
RNF002	El sistema necesita tener un SO Windows	Empleados	Alta	Sistema	En producción	General



2.01.05 Descripción detallada de requerimientos

Tabla 5

Descripción detallada del requerimiento funcional 001

Descripción del requerimiento	Estado	Valido
El sistema deberá permitir el ingreso al sistema solo por personal autorizado		
Creado por	Vera Kevin	Actualizado por Vera Kevin
Fecha Creación	01/07/2017	Fecha de actualización 14/07/2017
Identificador	RF001	
Tipo de requerimiento	Crítico	Tipo de requerimiento Funcional
Datos de entrada	Nombre de Usuario, Password	
Descripción	Realizar una interfaz donde permita el acceso al sistema únicamente a los usuarios registrados	
Datos de salida	Presentación del sistema de acuerdo al rol especificado	
Resultados esperados	Obtener parámetros de ingreso al sistema	
Origen	Gerencia	
Dirigido a	Gerencia, Empleados	
Prioridad	5	
Requerimientos asociados	Ninguno	
ESPECIFICACION		
Precondiciones	1. Para acceder al sistema los usuarios deben estar registrados por un administrador 2. Una vez realizado el registro podrá acceder al sistema a los diferentes módulos.	
Pos condiciones	El ingreso correcto permitirá la manipulación del módulo presentado	
Criterios de Aceptación	Permite el ingreso el usuario	



Tabla 6

Descripción detallada del requerimiento funcional 002

Descripción del requerimiento El sistema deberá permitir el ingreso de información de productos en el sistema		Estado	Valido
Creado por	Vera Kevin	Actualizado por	Vera Kevin
Fecha Creación	01/07/2017	Fecha de actualización	14/07/2017
Identificador	RF002		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Kardex de cada producto		
Descripción	Realizar una interfaz donde permita ingresar los productos en venta para emisión de la factura		
Datos de salida	Mensaje sobre producto creado con éxito		
Resultados esperados	Controlar el ingreso de productos que se encuentran en stock		
Origen	Gerencia		
Dirigido a	Empleados		
Prioridad	5		
Requerimientos asociados	RF001		
ESPECIFICACION			
Precondiciones	<ol style="list-style-type: none"> 1. Para acceder al ingreso y edición de artículos los usuarios deben estar logueados 2. Una vez realizado el ingreso o edición de artículos estos se colocaran si están activo/inactivo 		
Pos condiciones	El ingreso de artículos debe coincidir con la compra realizadas al proveedor		
Criterios de Aceptación	Permite el ingreso u modificación de artículos		



Tabla 7

Descripción detallada del requerimiento funcional 003

Descripción del requerimiento	El sistema deberá tener un inventario actualizado de los productos	Estado	Valido
Creado por	Vera Kevin	Actualizado por	Vera Kevin
Fecha Creación	01/07/2017	Fecha de actualización	14/07/2017
Identificador	RF003		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Stock producto, cantidad , precio		
Descripción	Realizar una interfaz donde permita visualizar el inventario existente para realizar compras		
Datos de salida	Inventario actualizado correctamente		
Resultados esperados	Verificar el inventario inicial con el inventario final de los movimientos de los artículos		
Origen	Gerencia		
Dirigido a	Empleados		
Prioridad	5		
Requerimientos asociados	RF001		
ESPECIFICACION			
Precondiciones	1. Para acceder al inventario inicial e inventario final debe tener accesos de administrador		
	2. Debe coincidir el stock del inventario final al del inventario inicial		
Pos condiciones	Una vez realizado el inventario los artículos no puede modificarse		
Criterios de Aceptación	Permite verificar el inventario		



Tabla 8

Descripción detallada del Requerimiento Funcional 004

Descripción del requerimiento	El sistema deberá permitir la emisión de facturas y órdenes de compra		Estado	Valido
Creado por	Vera Kevin	Actualizado por	Vera Kevin	
Fecha Creación	01/07/2017	Fecha de actualización	14/07/2017	
Identificador	RF004			
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional	
Datos de entrada	Datos cliente, Código producto, cantidad, precio, descuento			
Descripción	Realizar una interfaz para la generación de nuevas facturas solicitadas por el cliente.			
Datos de salida	Factura emitida			
Resultados esperados	Agilidad en el proceso de facturación			
Origen	Gerencia			
Dirigido a	Empleados			
Prioridad	5			
Requerimientos asociados	RF001			
ESPECIFICACION				
Precondiciones	1. Para la cabecera de la factura deben estar ingresados los datos del cliente			
Pos condiciones				
Criterios de Aceptación	Permite al usuario rapidez en el proceso de facturación			



Tabla 9

Descripción detallada del Requerimiento Funcional 005

Descripción del requerimiento El sistema deberá generar reportes de productos y facturas.		Estado	Valido
Creado por	Vera Kevin	Actualizado por	Vera Kevin
Fecha Creación	01/07/2017	Fecha de actualización	14/07/2017
Identificador	RF005		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Todas las facturas emitidas		
Descripción	Desarrollar una interfaz para la generación de reportes de ventas y facturas		
Datos de salida	Reporte digital u impreso		
Resultados esperados	Mayor eficiencia en la entrega de la factura y sobre todo integra en cuanto a manipulación		
Origen	Gerencia		
Dirigido a	Empleados		
Prioridad	5		
Requerimientos asociados	RF001-RF004		
ESPECIFICACION			
Precondiciones	1. Para ejecutar este requerimiento deben existir ventas realizadas		
Pos condiciones			



Tabla 10

Descripción detallada del requerimiento no funcional 001

Descripción del requerimiento El sistema deberá ser funcional en cualquier tipo de navegador	Estado	En producción
Creado por	Vera Kevin	Vera Kevin
Fecha Creación	01/07/2017	14/07/2017
	Actualizado por	
	Fecha de actualización	
Identificador	RNF001	
Tipo de requerimiento	Medio	Tipo de requerimiento Funcional
Datos de entrada		Ninguno
Descripción	El usuario podrá abrir en los aplicativos de mayor uso , Chrome Mozilla u Internet Explorer	
Datos de salida	Uso del sistema en cualquier navegador	
Resultados esperados	Facilidad de acceso al sistema	
Origen	Empleados	
Dirigido a	Administrador	
Prioridad	1	
Requerimientos asociados		
ESPECIFICACION		
Precondiciones	1. Para ejecutar este requerimiento el aplicativo debe ser probado en los navegadores mencionados	
Pos condiciones	Evidenciar funcionamiento o corregir las fallas	
Criterios de Aceptación	Mejor agilidad y acceso al usuario	



Tabla 11

Descripción detallada del requerimiento no funcional 002

Descripción del requerimiento El sistema necesita tener un SO Windows		Estado	En producción
Creado por	Vera Kevin	Actualizado por	Vera Kevin
Fecha Creación	01/07/2017	Fecha de actualización	14/07/2017
Identificador	RNF002		
Tipo de requerimiento	No Critico	Tipo de requerimiento	No funcional
Datos de entrada	Ninguno		
Descripción	El aplicativo debe funcionar en un sistema operativo Windows		
Datos de salida	Ninguno		
Resultados esperados	El aplicativo debe funcionar al 100% en un SO Windows		
Origen	Empleados		
Dirigido a	Empleados		
Prioridad	5		
Requerimientos asociados	RF001-RF004		
ESPECIFICACION			
Precondiciones	1. Para ejecutar este requerimiento el aplicativo debe ser compatible con SO Windows 2. Las terminales deberán estar conectadas en red para acceder al aplicativo		
Pos condiciones	El aplicativo podrá ser utilizado en cualquier pc si así se lo requiere		
Criterios de Aceptación	El aplicativo será compatible con SO Windows e instalada en 2 computadores por red		

2.02 Mapeo de Involucrados

El mapeo de involucrados ayuda a identificar a aquellas personas u organizaciones interesadas en el proyecto, como tal la entidad principal es la empresa Proveaseo

El Área de Facturación interviene como principal involucrado durante en el desarrollo de la aplicación, debido a que utilizara los procesos de producción y así enviara la información para realizar las entregas al cliente de manera confiable.

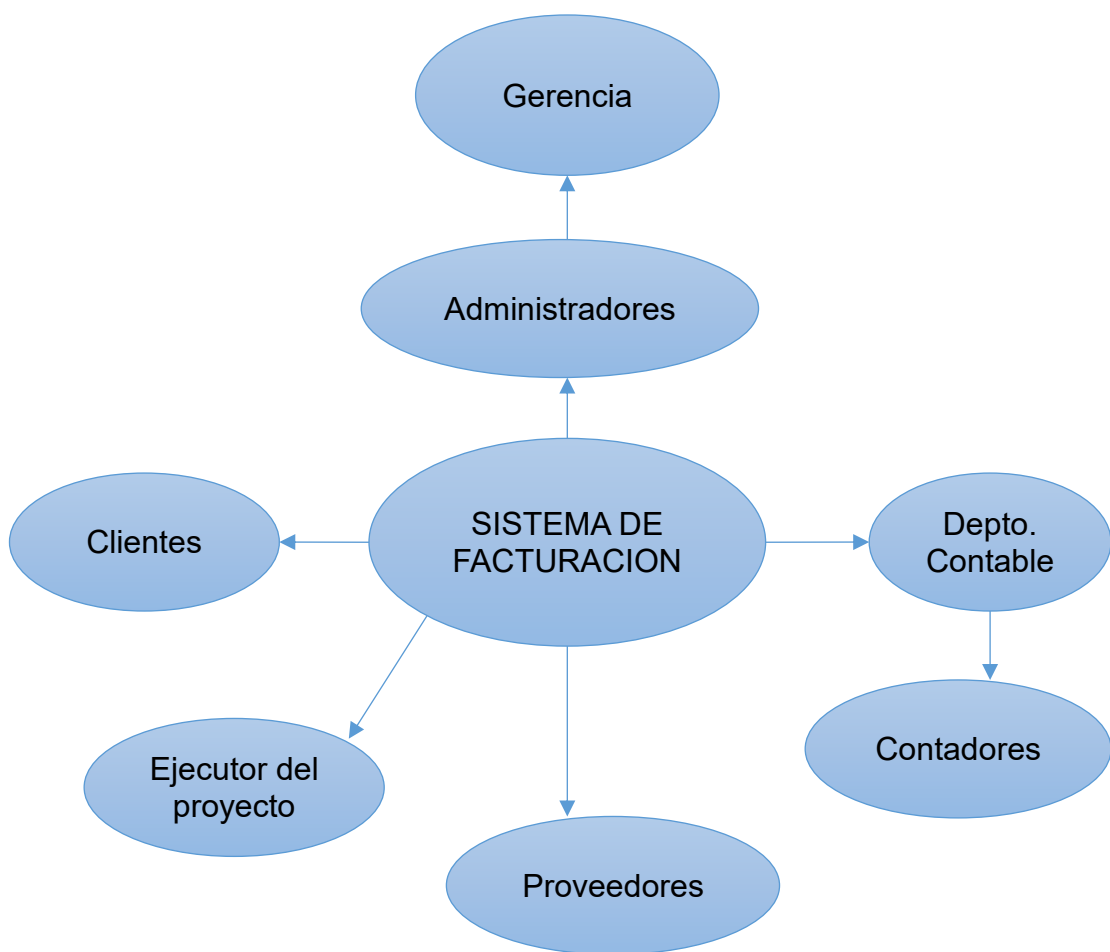


Figura 1 Mapeo de Involucrados



2.03 Matriz de Involucrados

Según Juan Pablo Pájaro en su publicación sobre análisis de involucrados en la web “El análisis de involucrados consiste en identificar los diferentes intereses, capacidades y necesidades de los grupos afectados por el proyecto de inversión. Para luego usar dichas diferencias en la definición de problemas, análisis de objetivos y selección de alternativas.”

Como particularidad se desarrolla una matriz de involucrados donde se redacta el Interés de cada uno de ellos y el problema percibido en la actualidad de la empresa Proveaseo. (Ver Anexo A1)

Capítulo III: Problemas y Objetivos

3.01 Árbol de Problemas

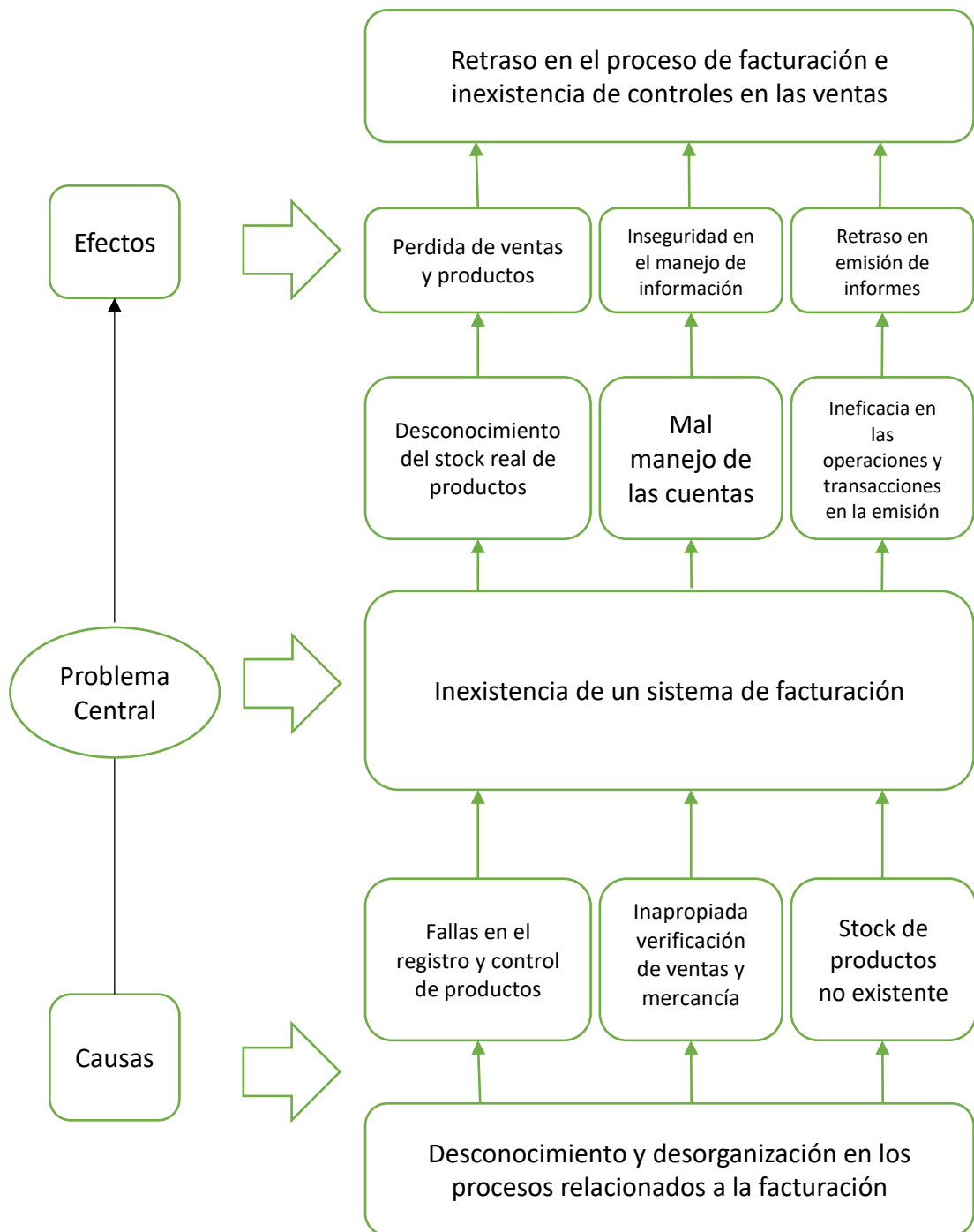


Figura 2 Árbol de Problemas

3.02 Árbol de Objetivos

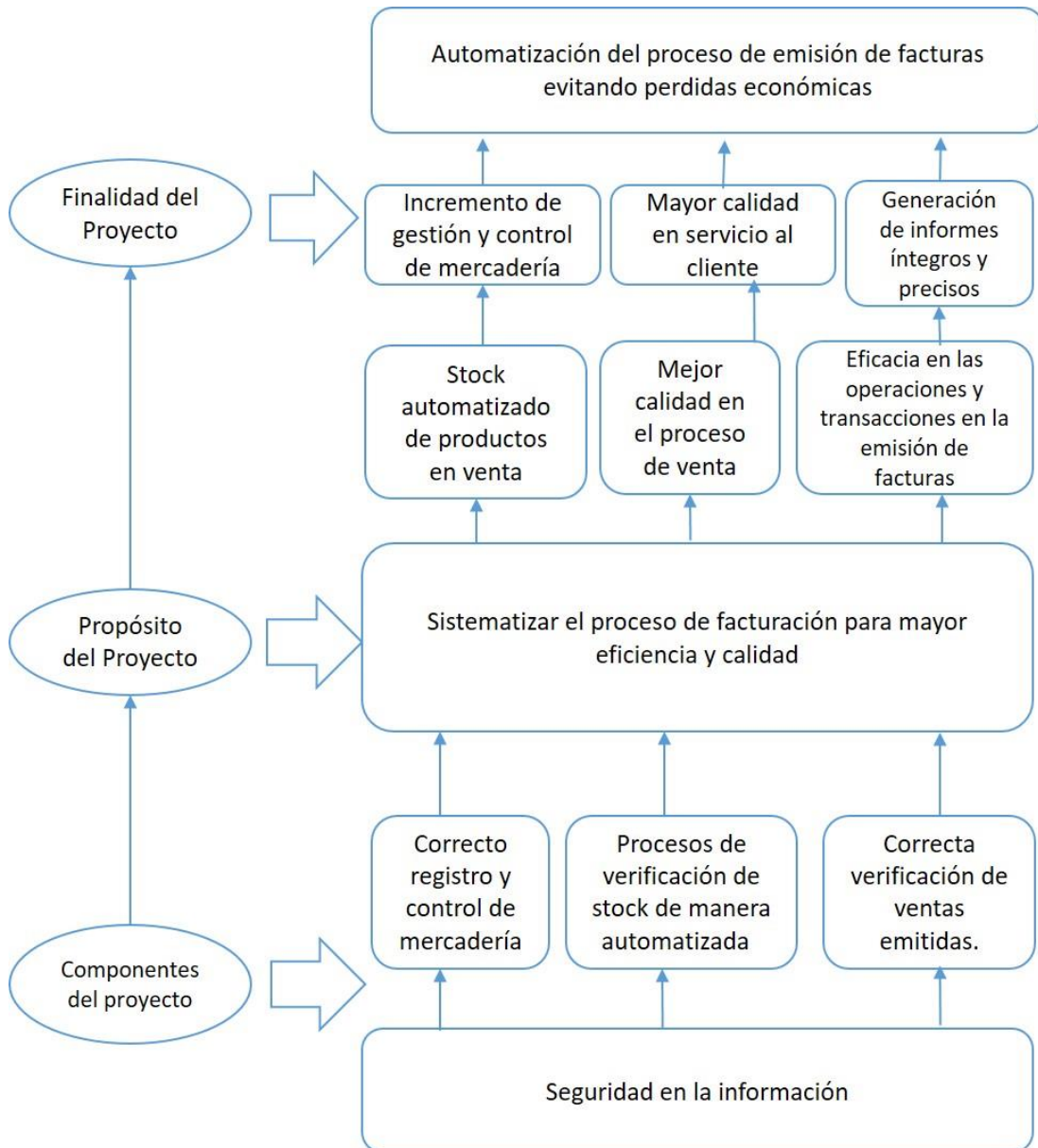


Figura 3 Árbol de Objetivos.

En la figura se muestra los principales objetivos para mejorar el proceso de facturación.

3.03 Diagramas de Casos de Uso

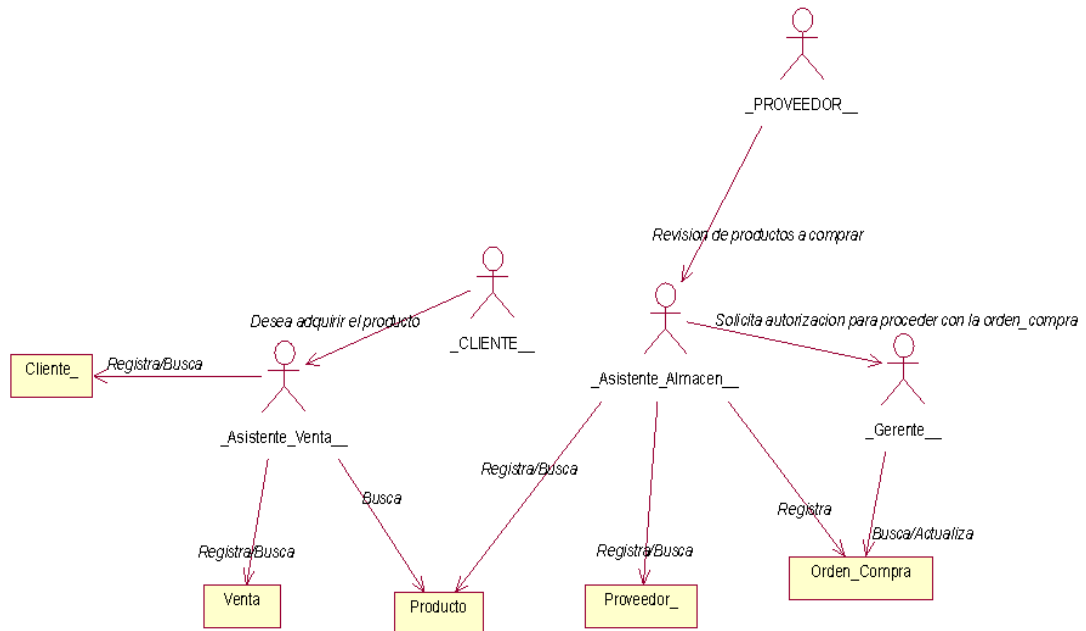


Figura 4 Caso de Uso General Proceso de Facturación

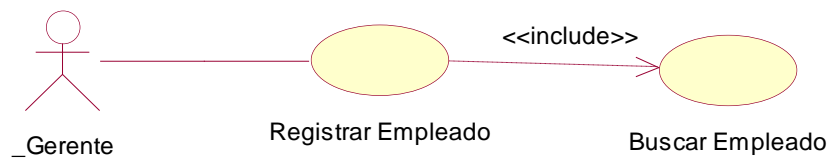


Figura 5 Caso de Uso Registrar Empleado

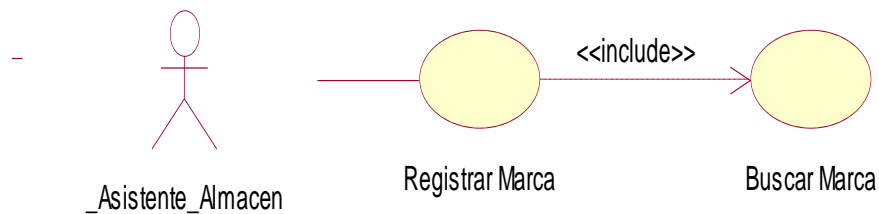


Figura 6 Caso de Uso Registrar Marca de producto

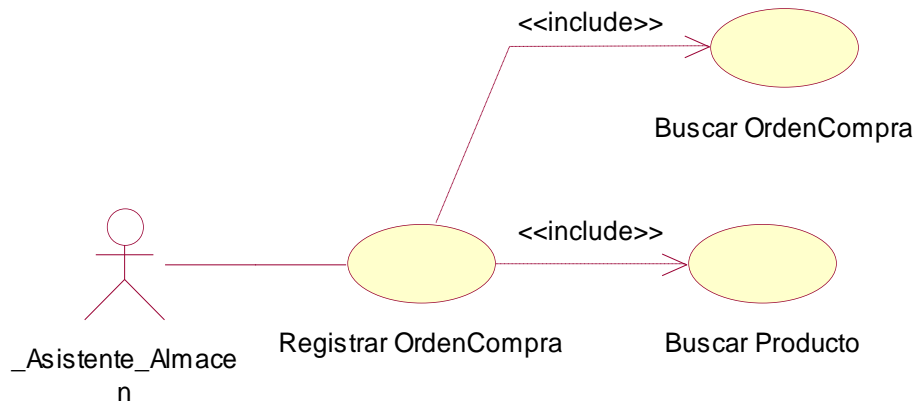


Figura 7 Caso de Uso Registrar Orden de Compra

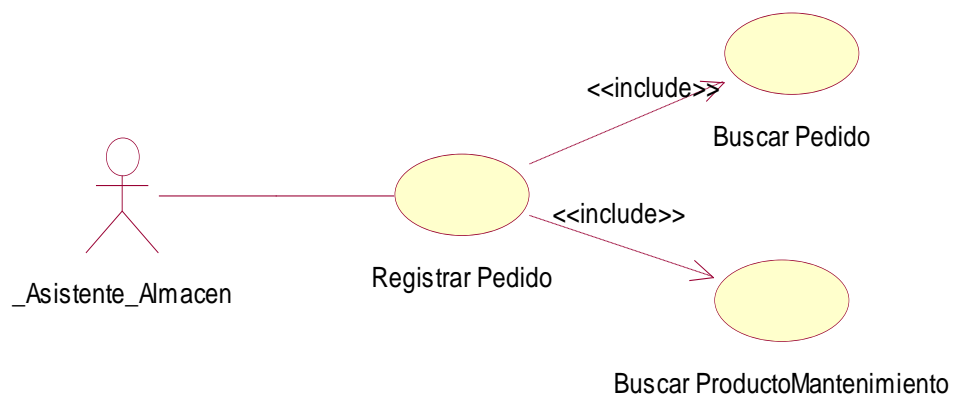


Figura 8 Caso de Uso Registrar Pedido

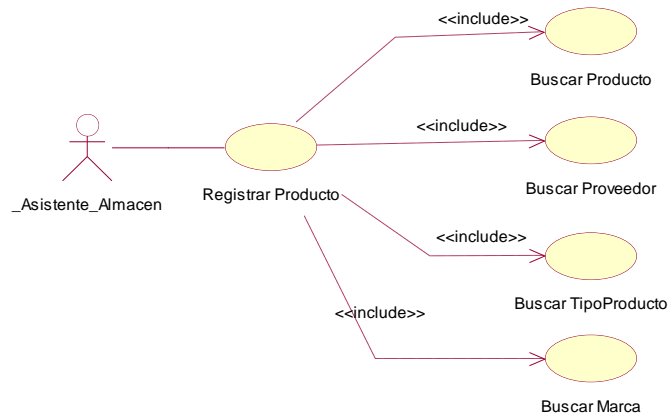


Figura 9 Caso de Uso Registrar Producto

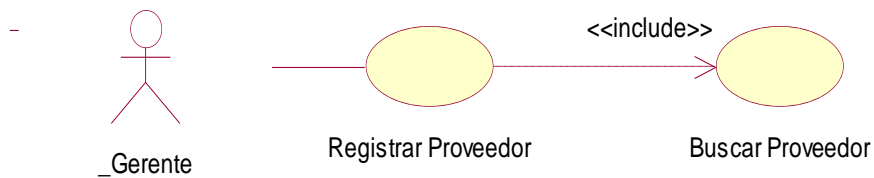


Figura 10 Caso de Uso Registrar Proveedor

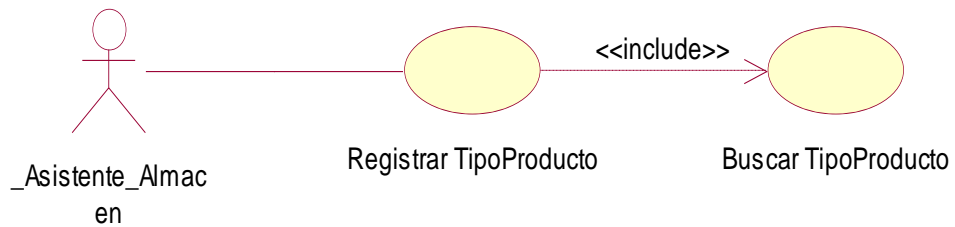


Figura 11 Caso de Uso Registrar tipo de producto

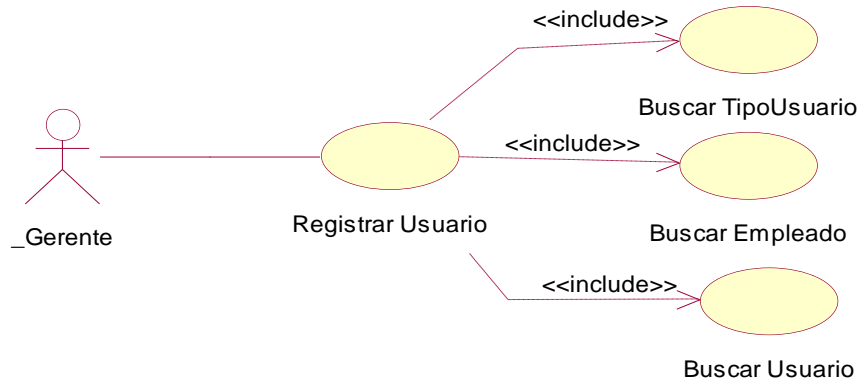


Figura 12 Caso de Uso Registrar Usuario

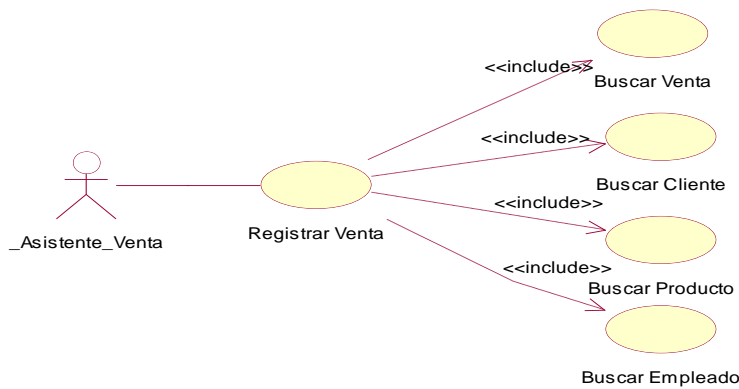


Figura 13 Caso de Uso Registrar Venta

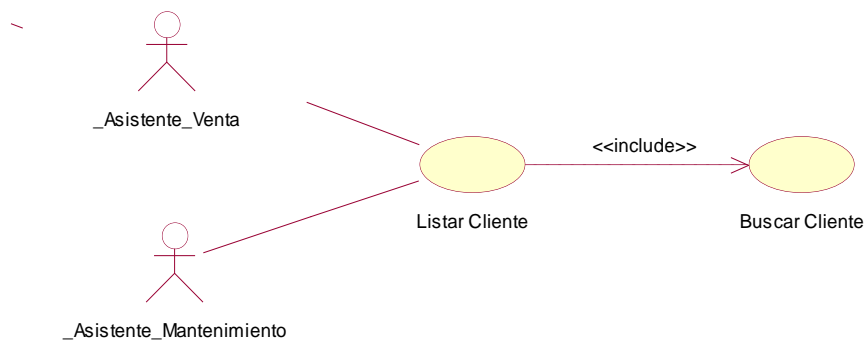


Figura 14 Caso de Uso Listar Cliente

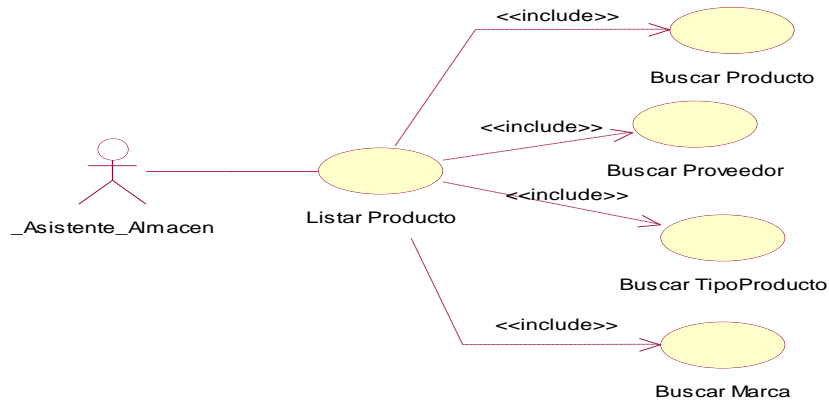


Figura 15 Caso de Uso Listar Producto

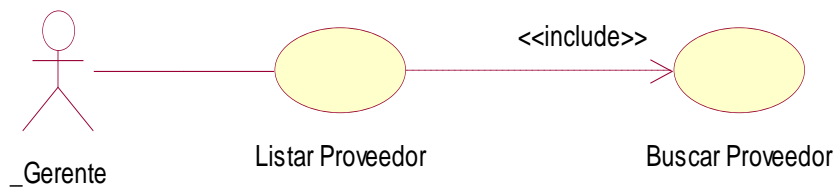


Figura 16 Caso de Uso Listar Proveedor

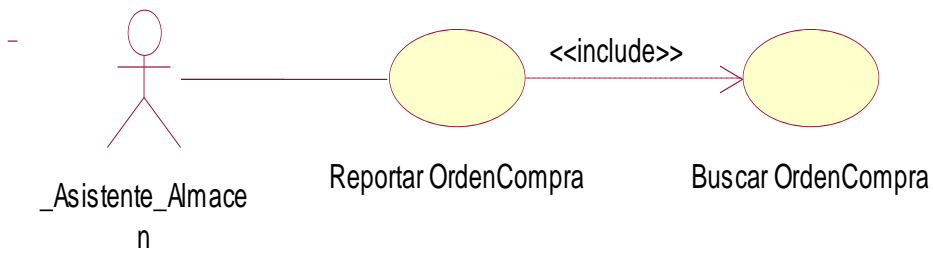


Figura 17 Reporte Orden de Compra

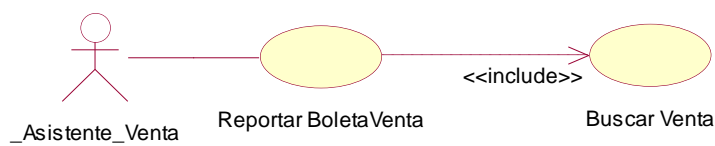


Figura 18 Reporte de Venta



3.04 Especificación de Casos de Uso

Tabla 12

Especificación Caso de Uso Registrar Usuario

RegEmp	Registrar Usuario	
Descripción	El Sistema deberá permitir al Administrador registrar nuevos empleados que trabajaran en la empresa. Según se describe en el siguiente caso de uso:	
Precondición		
Secuencia	Paso	Acción
Normal	1	Buscar Empleado
	2	Registrar Empleado
Post condición		
Salidas Pantalla	Usuario registrado	

Tabla 13

Especificación Caso de Uso Registrar Orden Compra

RegOrdComp	Registrar OrdenCompra	
Descripción	El Gerente verifica los pedidos pendientes y realiza una orden de compra.	
Precondición	Registrar pedido.	
Secuencia	Paso	Acción
Normal	1	Buscar pedido.
	2	Buscar orden de compra.
	3	Registrar orden de compra.
Post condición		
Excepciones	Paso	Acción
	E-1	Entrada Inválida. El usuario ha ingresado un dato no valido.
	E-2	Datos no válidos, el sistema no puede grabar el registro.
	E-3	La Información no puede ser recuperada.
Salidas pantalla	Orden de compra registrado	



Tabla 14

Especificación Caso de Uso Registrar Producto

RegProd	RegistrarProducto	
Descripción	El asistente de almacén se encarga de registrar los nuevos productos para su posterior venta	
Precondición	Registrar tipo producto. Registrar proveedor. Registrar marca	
Secuencia	Paso	Acción
Normal	1.	Buscar proveedor
	2.	Buscar tipo producto
	3.	Buscar Marca
	4.	Registrar producto.
Post condición		
Excepciones	Paso	Acción
	E-1	Entrada Inválida. El usuario ha ingresado un dato no valido.
	E-2	Datos no válidos, el sistema no puede grabar el registro.
	E-3	La Información no puede ser recuperada.
Salidas pantalla	Producto Registrado	

Tabla 15

Especificación Caso de Uso Registrar Tipo de Producto

RegTipoProd	Registrar TipoProducto
Descripción	El asistente de almacén procede a registrar el tipo de mantenimiento
Precondición	Buscar tipo de producto
Secuencia	Paso Acción
Normal	1 Buscar tipo de producto. 2 Registrar tipo de producto.
Post condición	Tipo de producto registrado.
Excepciones	Paso Acción
	E-1 Entrada Inválida. El usuario ha ingresado un dato no valido.
	E-2 Datos no válidos, el sistema no puede grabar el registro.
	E-3 La Información no puede ser recuperada.
Salidas pantalla	Tipo de Producto registrado

Tabla 16

Especificación Caso de Uso Registrar Tipo de usuario

RegTipoUsu	RegistrarTipoUsuario
Descripción	El Gerente procede a registrar el tipo de usuario
Precondición	Buscar tipo de usuario.
Secuencia	Paso Acción
Normal	1 Buscar tipo de usuario. 2 Registrar tipo de usuario.
Post condición	
Excepciones	Paso Acción
	E-1 Entrada Inválida. El usuario ha ingresado un dato no valido.
	E-2 Datos no válidos, el sistema no puede grabar el registro.
	E-3 La información no puede ser recuperada.
Salida Pantalla	Tipo de usuario registrado



Tabla 17

Especificación Caso de Uso Registrar Venta

Regven	Registrar Venta	
Descripción	El asistente de venta procede a registrar la venta	
Precondición	Registrar empleado. Registrar cliente. Registrar producto.	
Secuencia Normal	Paso	Acción
	1	Buscar empleado.
	2	Buscar cliente.
	3	Buscar producto.
	4.	Registrar venta.
Post condición		
Excepciones	Paso	Acción
	E-1	Entrada Inválida. El usuario ha ingresado un dato no valido.
	E-2	Datos no válidos, el sistema no puede grabar el registro.
	E-3	La información no puede ser recuperada.
Salida Pantalla	Venta registrada	

Tabla 18

Listado de Cliente

LisCli	Listado Cliente	
Descripción	El Sistema deberá permitir a la secretaria realizar un listado de todos los clientes registrados en forma detallada.	
Precondición	Registrar cliente.	
Secuencia Normal	Paso	Acción
	2	Buscar cliente.
	3	Listar clientes.
Post condición		
Salida Pantalla	Listado de clientes	



Tabla 19

Especificación Caso de Uso Reporte de Orden de Compra

RepOrdCom	Reporte de Orden de Compra	
Descripción	El Sistema deberá permitir al Administrador realizar un reporte de todas las compras en forma detallada o especificando una condición.	
Precondición	Orden de Compra registrado	
Secuencia	Paso	Acción
Normal	1	Buscar pedido.
	3	Buscar orden de compra.
	2	Reporte de orden de compra
Post condición		
Salida Pantalla	Reporte de orden de compra	

Tabla 20

Reporte de Ventas

RegBolVen	Reporte Ventas	
Descripción	El Sistema deberá permitir al asistente de venta realizar un reporte de todas las ventas de los clientes naturales.	
Precondición	Registrar Venta.	
Secuencia	Paso	Acción
Normal	1	Buscar venta.
	3	Reportar BoletaVenta.
Post condición		
Salida Pantalla	Reporte de venta realizado	

3.05 Casos de uso de Realización

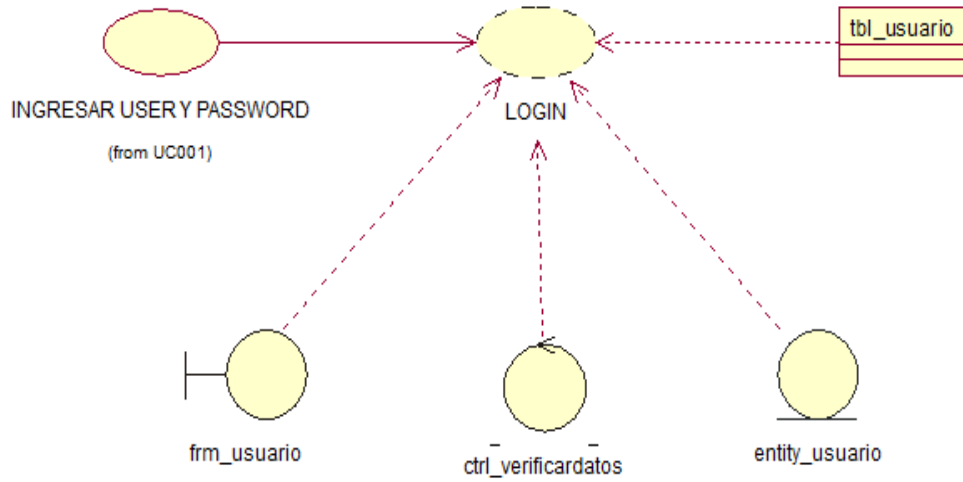


Figura 19 Caso de Uso de realización Ingreso al aplicativo

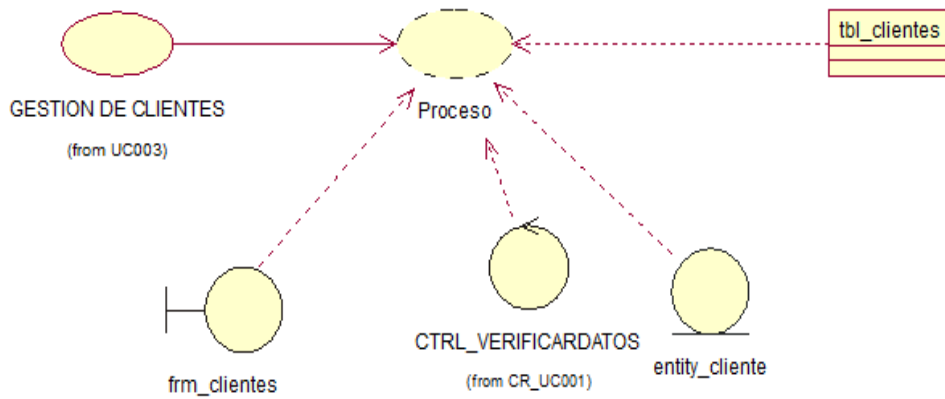


Figura 20 Caso de Uso de realización Gestión de clientes

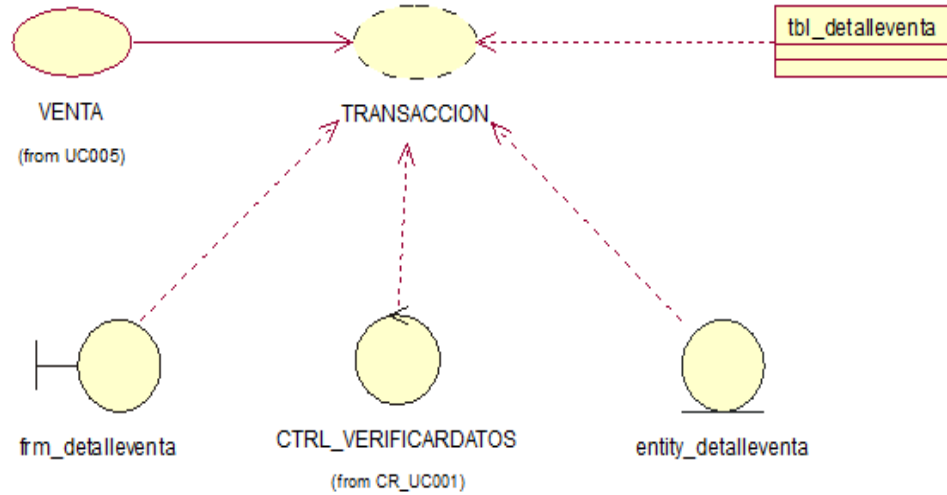


Figura 21 Caso de uso de realización Generar Venta

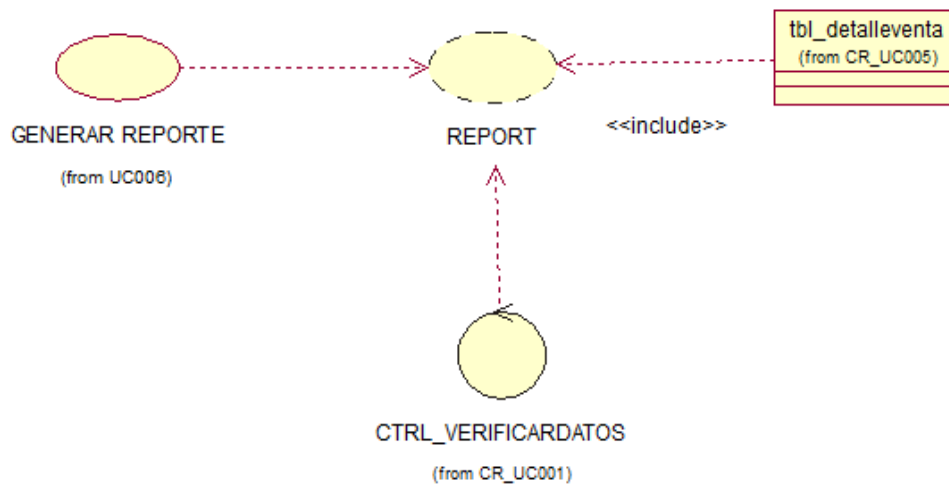


Figura 22 Caso de uso de realización Emitir Reporte

3.06 Diagrama de Secuencias

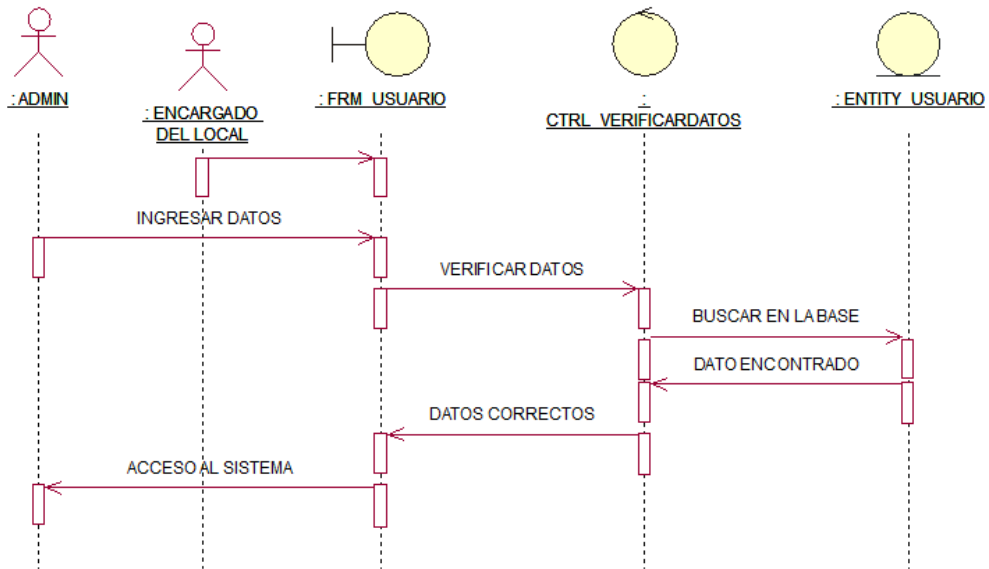


Figura 23 Diagrama de Secuencia Ingreso al aplicativo

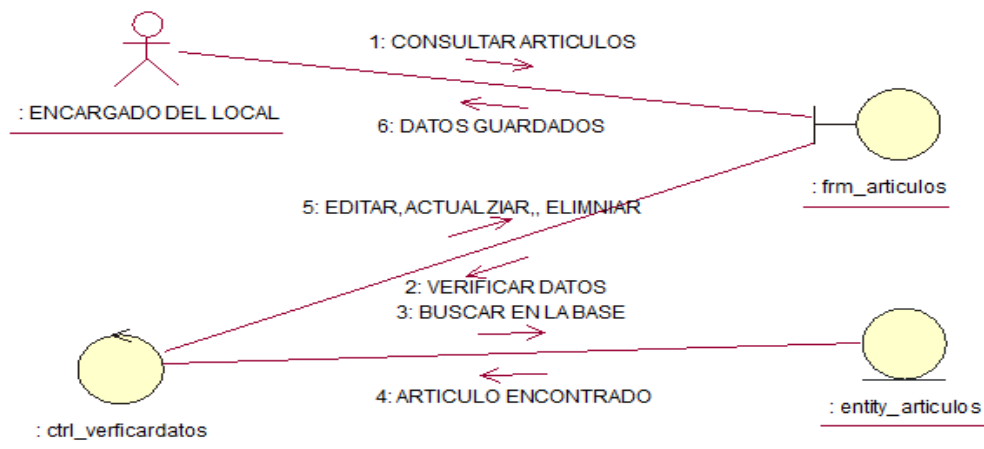


Figura 24 Diagrama de Secuencia Ingreso de información

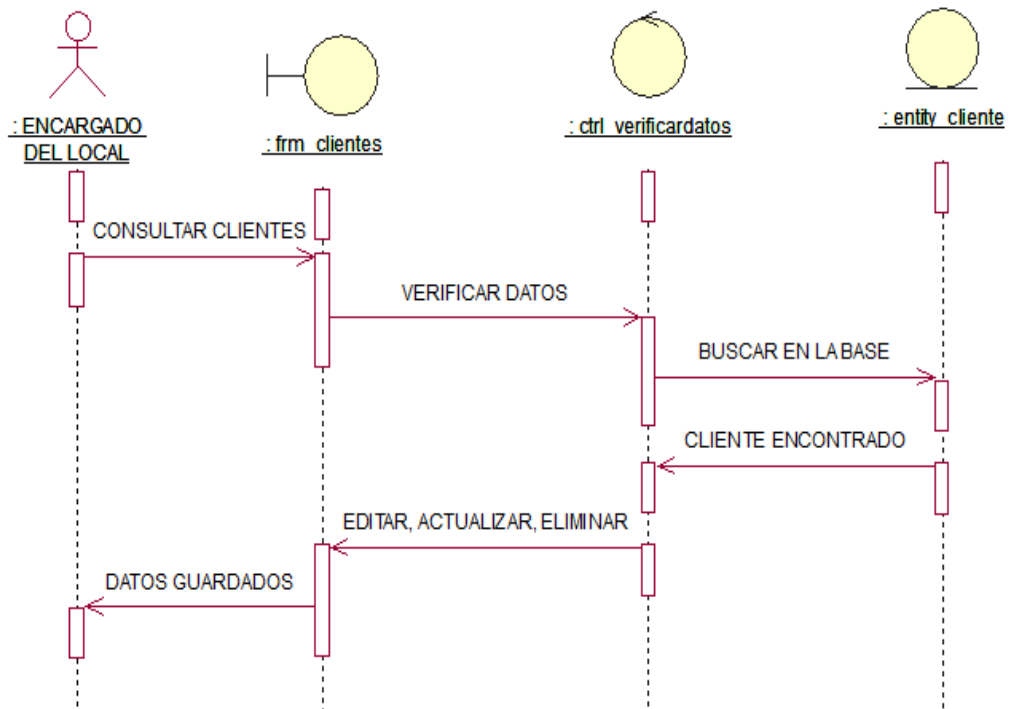


Figura 25 Diagrama de Secuencias Ingreso de Clientes

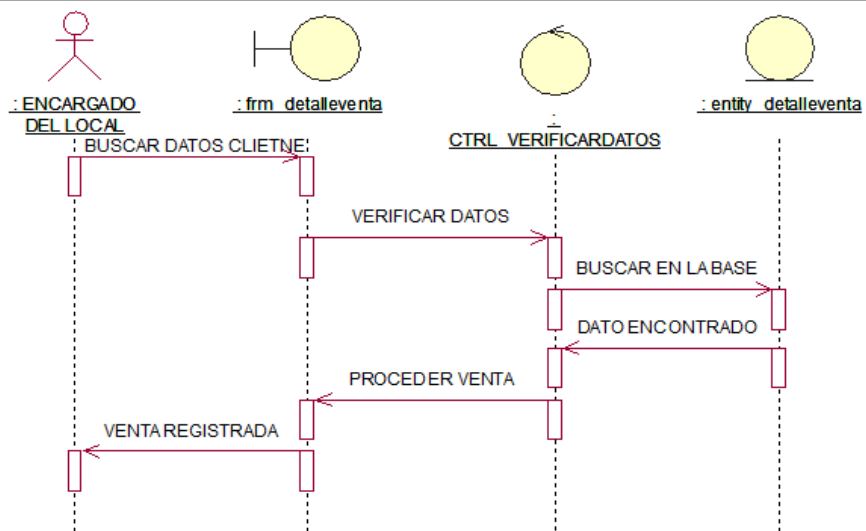


Figura 26 Diagrama de Secuencias Generar Factura

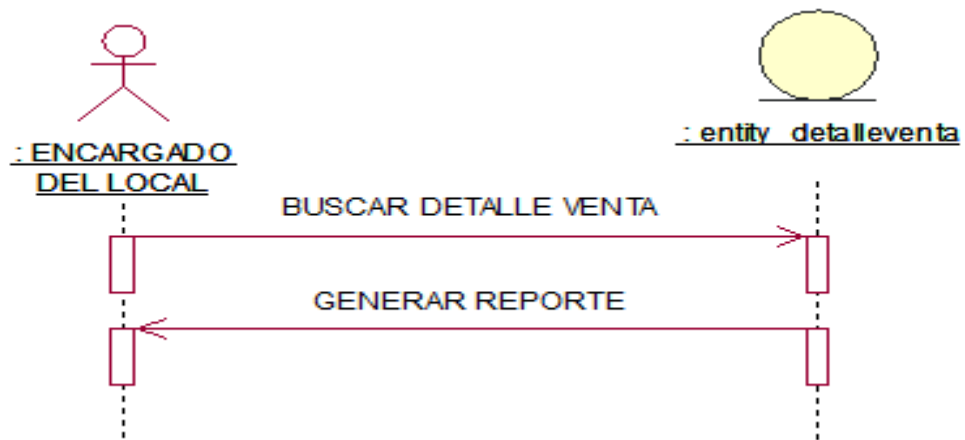


Figura 27 Diagrama de Secuencia Generar Reporte

Capítulo IV: Análisis de Alternativas

4.0.1 Matriz de Análisis de Alternativas

Tabla 21

Matriz de Análisis de Alternativas

Objetivos	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Controlar las ventas adecuadamente con registro de ingresos	4	4	4	3	3	18	Alta
Crear controles para el inventario de productos	4	4	3	4	3	18	Alta
Mejorar tiempos de respuesta en el proceso de facturación	5	4	3	4	4	20	Alta
Entrega de reportes de manera optima	4	5	3	4	3	19	Alta
Mejorar servicio al cliente	4	4	3	3	4	18	Alta
Total	21	21	16	18	17	93	Alta



4.0.2 Análisis de la Matriz de Alternativas

De esta matriz se puede analizar que tenemos los objetivos que referentes que ayudarán a llegar al propósito que es la automatización de procesos, tienen un valor alto, ya que con la implementación del sistema se logra entender que mejorará los procesos requeridos.

4.0.3 Matriz de Impacto de Objetivos

Ver Anexo A02

4.0.4 Análisis de la matriz de impacto de objetivos

En la matriz de impacto de los objetivos se puede determinar que existe factibilidad, relevancia y sostenibilidad para el impacto que incide en el desarrollo del proyecto.

4.0.5 Estándares para el Diseño de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de acontecimiento.

- Clase: atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Elementos

Clase

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones:

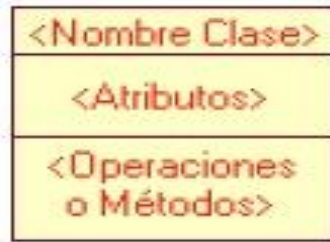


Figura 28 Representación de una clase


Superior: Contiene el nombre de la Clase


Intermedio: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).


Inferior: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

Atributos y Métodos:

Los atributos o métodos de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

Public (+, ): Indica que el atributo o el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

Private (-, ): Indica que el atributo o el método sólo será accesible desde dentro de la clase (sólo sus métodos pueden acceder).

Protected (#, ): Indica que el atributo o el método no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven (ver herencia).

Relaciones entre Clases:

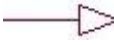
Ahora ya definido el concepto de Clase, es necesario explicar cómo se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

uno o muchos: 1...* (1..n)

0 o muchos: 0...* (0..n)

Número fijo: m (m denota el número).

Herencia (Especialización/Generalización): 

Indica que una subclase hereda los métodos y atributos especificados por una súper Clase, por ende, la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la súper Clase (public y protected).

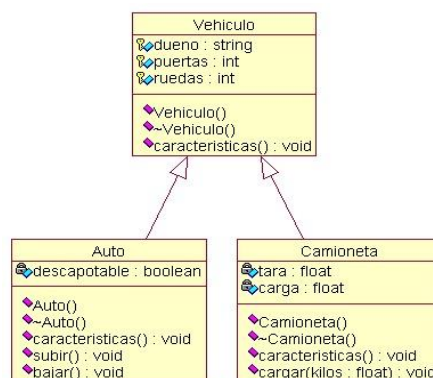



Figura 29 Representación de la relación de herencia

En la figura se especifica que Auto y Camión heredan de Vehículo, es decir, Auto posee las Características de Vehículo (Precio, VelMax, etc.) además posee algo particular que es Descapotable, en cambio Camión también hereda las características de Vehículo (Precio, VelMax, etc.) pero posee como particularidad propia Acoplado, Tara y Carga.

Agregación: 

Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

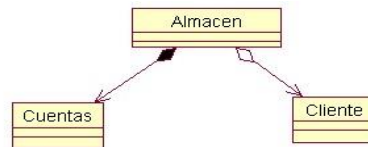




Figura 30 Representación de asociaciones en general

Asociación: 

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre sí.



Figura 31 Representación de relación de asociación

Dependencia o Instanciación (uso): 

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). (Lizana,2015)



Figura 32 Representación de relación de dependencia

4.0.8 Diagrama de Estrategia

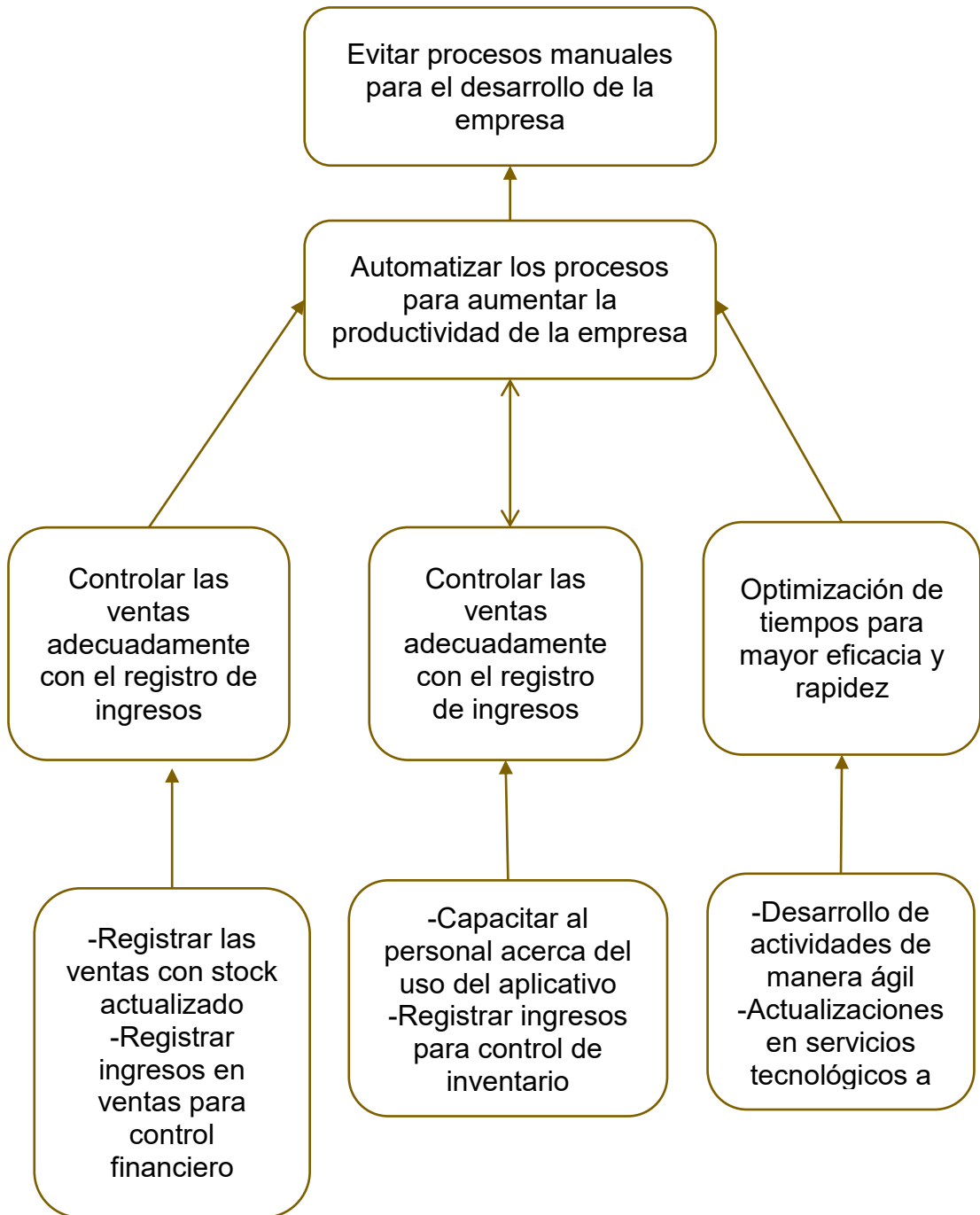


Figura 35 Diagrama de Estrategias

4.01.02 Vista física del diseño de la solución

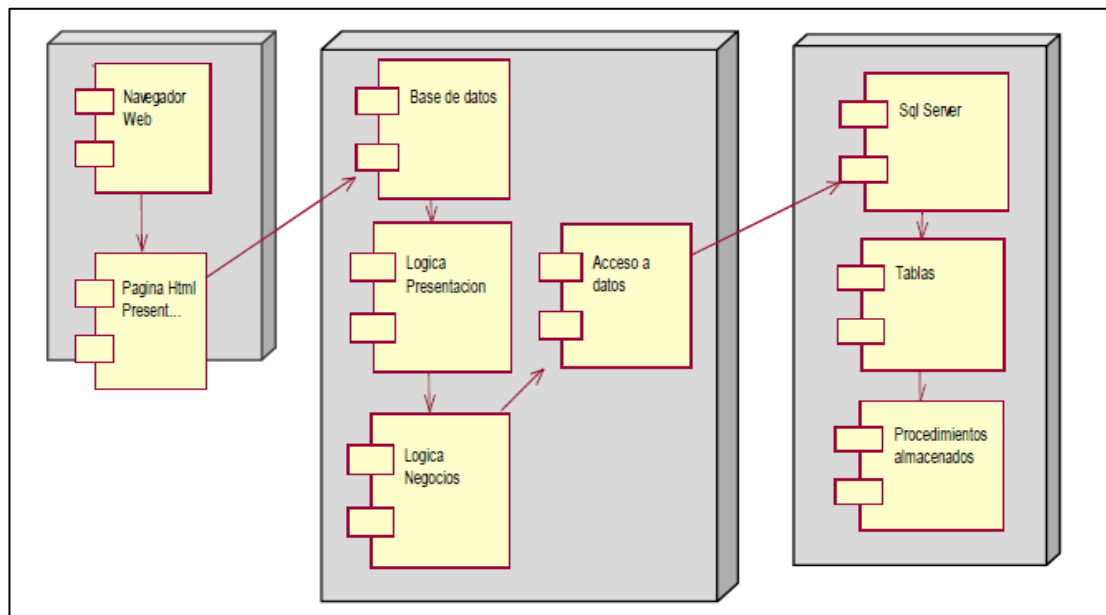


Figura 37 Vista física del diseño de la solución

4.01.03 Vista de desarrollo

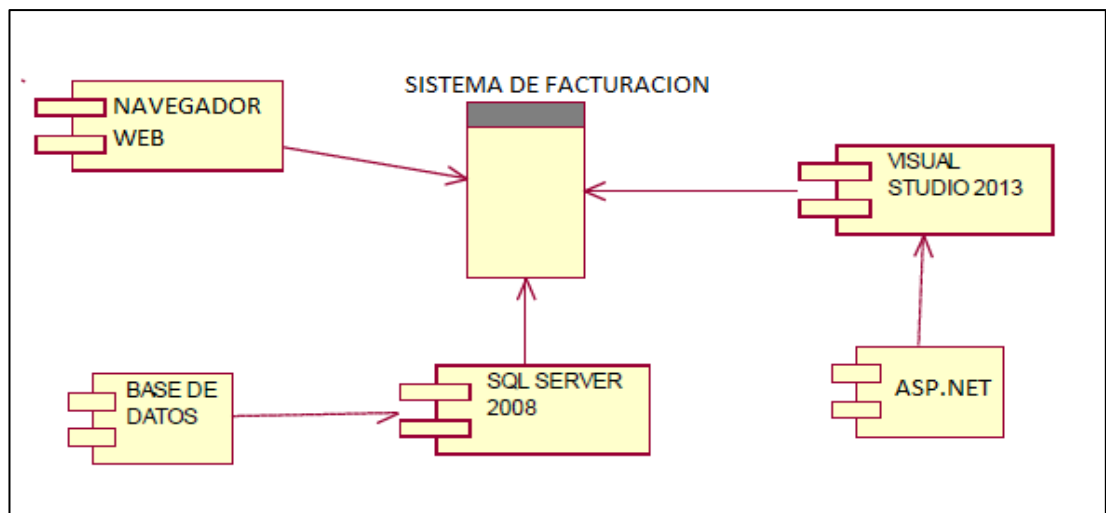


Figura 38 Vista de Desarrollo.

Esta figura muestra la vista de Desarrollo de la solución

4.01.04 Vista de Procesos

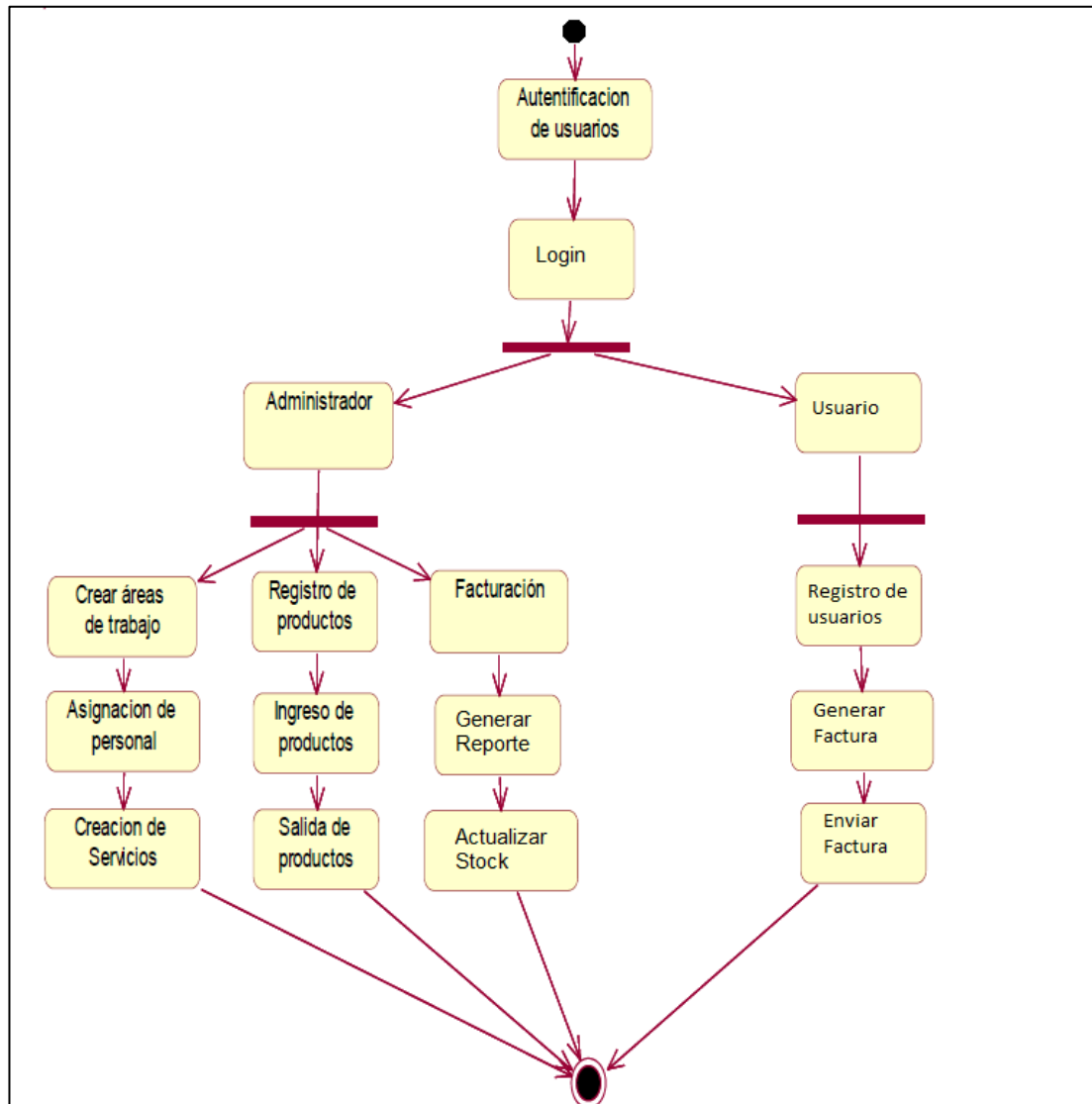


Figura 39 Vista de Procesos del diseño de la solución

Capítulo V: Propuesta

5.01 Especificación de estándares de programación

Para la programación en la fase de desarrollo se debe manejar una nomenclatura la cual permita un estándar evitando así redundancia de datos.

5.01.01 Nomenclatura .NET Objetos

Tabla 22

Especificación de estándares .NET Objetos

Tipo de Objeto	Nomenclatura	Ejemplo
Formulario	frm	frmUsuarios
RadioButton	rb	rbactivo
CheckBox	chk	chktipo
GridView	gv	gvClientes
DropDownList	ddl	ddlmodelo
Button	btn	btningresar
Image	img	imgLogo
Label	lbl	lblverificar
TextBox	tb	tbapellido

5.01.02 Nomenclatura .NET Clases, métodos y variables

Tabla 23

Especificación de estándares .NET Clases

Tipo de objeto	Nomenclatura	Ejemplo
Clases	Cls_ + "nombre clase"	Public class Cls_conexion
Métodos	"nombre método" ()	cerrarconexion()
Variables	"fac "+"tipo de dato"	fac_nombre



5.01.03 Nomenclatura Base de Datos

Tabla 24

Especificación de estándares BD Tipo de dato

Tipo de dato	Uso
INT	Para campos que requieran valores numéricos enteros y claves primarias.
NVARCHAR	Para campos que requieran textos y números
DECIMAL	Para campos que requieran valores decimales
DATE	Para campos de fecha
DATETIME	Para campos de fecha y hora
CHAR(1)	Para campos con ingreso de un solo carácter



Tabla 25

Especificación de estándares BD Tablas, Campos, Funciones y Procedimientos

Almacenados

Tipo	Especificación	Ejemplo
TABLAS	Para nombrar a la tabla se inicia con el prefijo "FAC" seguido de guion bajo y finalmente la descripción de la tabla , todo en mayúsculas	FAC_CLIENTES
CAMPOS	Para nombrar a los campos se inicia con las 3 iniciales de la descripción de la tabla seguido de guion bajo y finalmente la descripción del campo	CLI_NOMBRE
CLAVES PRIMARIAS Y FORANEAS	La descripción para las claves primarias será CLAVE	CLI_CLAVE
FUNCIONES	Para nombrar funciones se inicia con el prefijo "FN" seguido de guion bajo y la descripción de la función	FN_CEDULA
PROCEDIMIENTOS ALMACENADOS	Para nombrar procedimientos se inicia con el prefijo "PA" seguido de guion bajo y la descripción de la función	PA_CAPTURARTIPO

5.02 Diseño de interfaces de usuario

The screenshot shows a login form titled "Productos de Limpieza ProveAseo". The title "ProveAseo" is prominently displayed in large, stylized letters, with "Prove" in blue and "Aseo" in yellow. Below the title, there are two input fields: "Nombre de Usuario" and "Password". Both fields have a placeholder text "Ingrese nombre de usuario". Below the password field is a blue button labeled "INGRESAR".

Figura 40 Ingreso al aplicativo.

La figura muestra el ingreso del aplicativo donde se ingresa nombre de usuario y contraseña proporcionados por administrador, al dar clic en el botón "Ingresar" le proporciona acceso a la interface principal, en caso de no estar registrado se despliega un mensaje de error.



Figura 41 Pantalla de inicio

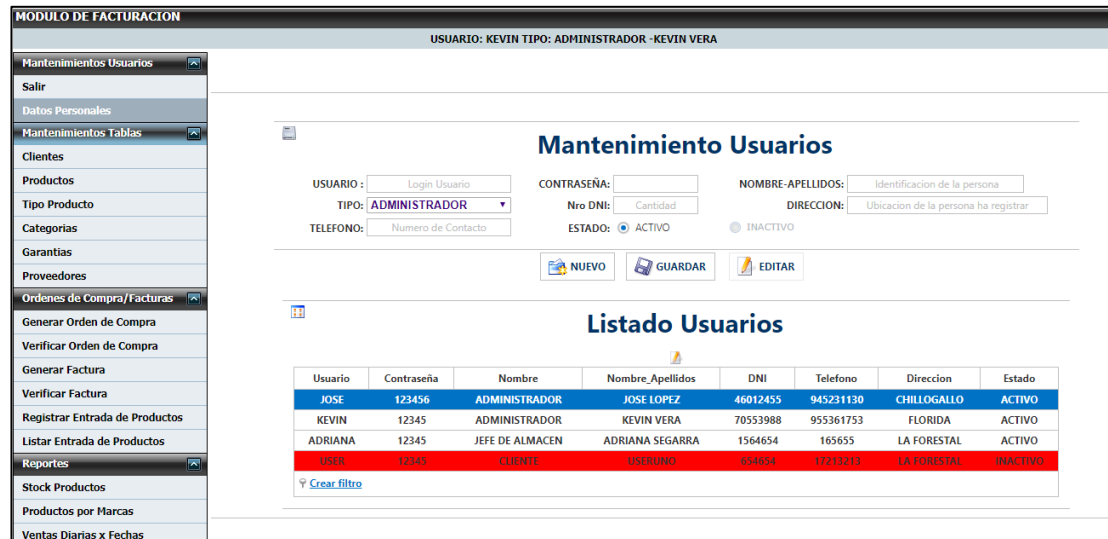


Figura 42 Mantenimiento Usuarios

En la figura se observa la interfaz para crear nuevos usuarios, editar usuarios, activar u inactivar usuarios u modificar contraseñas.

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

MANTENIMIENTO DE PRODUCTOS DE ALMACEN

INGRESO DE DATOS

NOMBRE: MARCA: DESCRIPCION:

TIPO: STOCK: PRECIO COMPRA: PRECIO VENTA:

ESTADO: ACTIVO INACTIVO

LISTADO - BUSQUEDA DE DATOS

NOMBRE	TIPO	MARCA	PRECIO_COMPRA	PRECIO_VENTA	STOCK	DESCRIPCION	ESTADO
JABON LIQUIDO	JABONES	JHONSON BABY	125,00	129,00	231	JHONSON BABY 250ML	ACTIVO
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA	50,00	60,00	-8	PAPEL HIGIENICO *6	ACTIVO
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT	28,00	35,00	52	PAPEL HIGIENICO * 12	ACTIVO
SUMANS	JABONES	DESENGRASANTE	52,00	62,00	-11		ACTIVO
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE	15,00	25,00	10	DESINFECTANTE PISOS 250ML FRAGANCIA LIMON	ACTIVO

Figura 43 Mantenimiento Productos

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

MANTENIMIENTO MARCA DEL PRODUCTO

INGRESO DE DATOS

NOMBRE:

ESTADO: ACTIVO INACTIVO

LISTADO - BUSQUEDA DE DATOS

NOMBRE	ESTADO
JHONSON BABY	1
SOFT	1
FAMILIA	1
DESENGRASANTE	1
AXION	1

[Crear filtro](#)

Figura 44 Mantenimiento Marcas Productos.

En la figura se puede observar la interfaz para creación de marcas de productos, editar, activar u inactivar marcas.

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantias

Proveedores

Órdenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Productos por Marcas

Ventas Diarias x Fechas

MANTENIMIENTO TIPO DE PRODUCTO

INGRESO DE DATOS

NOMBRE:

ESTADO: ACTIVO INACTIVO

LISTADO - BUSQUEDA DE DATOS

NOMBRE	ESTADO
AMBIENTALES	1
JABONES	1
PAPEL HIGIENICO	1
LIMPIEZA PISOS	1

[Crear filtro](#)

Figura 45 Mantenimiento Categorías Productos.

En la figura se puede observar la interfaz para creación de categorías de productos, editar, activar u inactivar categorías.

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantias

Proveedores

Órdenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Productos por Marcas

Ventas Diarias x Fechas

MANTENIMIENTO DE PROVEEDORES

INGRESO DE DATOS

NOMBRE EMPRESA: RUC: DIRECCION:

NOMBRE CONTACTO: TELEFONO: EMAIL:

ESTADO: ACTIVO INACTIVO

CUENTA BANCARIA:

LISTADO PROVEEDORES

NOMBRE_COMPANIA	NOMBRE_CONTACTO	RUC	BANCARIO	TELEFONO	DIRECCION	EMAIL	ESTADO
ROUCE DATA	ROGER REYES	45487884545	xxx - 5454	954545454	CHIMBOTE		ACTIVO
ECO GRANT	JUAN PERES	54845454545	BCC-551545454	943074990	CHIMBOTE		ACTIVO
PARTS KIL	ANA BENITES	48454845454	BCP-54545454545	943958455	CASMA		ACTIVO
AUGART	FELIPE PERES	84478464248	BCP-45454545454	859593933	CHIMBOTE		INACTIVO

[Crear filtro](#)

Figura 46 Mantenimiento de Proveedores.

En la figura se puede observar la interfaz para creación de proveedores, editar, activar u inactivar proveedores.

REGISTRO DE ORDEN DE COMPRA

INGRESO LISTADO

Orden de Compra

AUTORIZADO:

PROVEEDOR: KEVIN VERA - USUARIO: KEVIN

DESCRIPCION:

ESTADO: ENVIADO ACTIVADO ELIMINADO

NUEVO GUARDAR EDITAR

Arrastre una columna aquí...

MONTO TOTAL:

Buscar Producto

NOMBRE	TIPO	MARCA
JABON LIQUIDO	JABONES	JHONSON BABY
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT
SUMANS	JABONES	DESENGRASANTE
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE

[Crear filtro](#)

Figura 47 Generar Orden de Compra.

En la figura se puede observar la interfaz para la generación de orden de compra a proveedores.

REGISTRO DE ORDEN DE COMPRA

INGRESO LISTADO

Buscar Orden de Compra

Num_ORDEN_COMPRA	FECHA	MONTO_TOTAL	NOMBRE_COMPANIA	RUC	ESTADO
C0001	02/04/2017	500,00	PYDACO	45487884545	ENVIADO
C0002	06/05/2017	520,00	SUPERUNO	54845454545	ENVIADO
C0003	06/05/2017	250,00	SUPERUNO	54845454545	ENVIADO
C0004	07/09/2017	260,00	SUPERUNO	54845454545	ENVIADO
C0005	07/09/2017	2080,00	PYDACO	45487884545	ENVIADO
C0006	08/09/2017	3750,00	SUPERUNO	54845454545	ENVIADO
C0007	08/09/2017	500,00	SUPERUNO	54845454545	ENVIADO

[Crear filtro](#)

Figura 48 Listado de Órdenes de Compra.

En la figura se puede observar el listado de las órdenes de compra realizadas para control y actualización de stock.

MODULO DE FACTURACION
USUARIO: KEVIN TIPO: ADMINISTRADOR -KEVIN VERA

Nueva Factura

AUTORIZADO: KEVIN VERA -USUARIO: KEVIN

NOMBRES: Datos personales del comprador DNI/RUC: Numero Dni / Ruc

Arrastre una columna aquí...

Sin datos para mostrar

MONTO TOTAL:

Buscar Producto

NOMBRE	TIPO	MARCA
JABON LIQUIDO	JABONES	JHONSON BABY
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT
SUMANS	JABONES	DESENGRASANTE
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE

[Crear filtro](#)

Figura 49 Generar Factura

En la figura se puede observar la interfaz para generar una nueva factura.

MODULO DE FACTURACION
USUARIO: KEVIN TIPO: ADMINISTRADOR -KEVIN VERA

REPORTE DE PRODUCTOS X TIPO DE PRODUCTO

TIPO PRODUCTO MARCA PRODUCTO

1 of 1 Find | Next

PROVEASEO
Productos de Limpieza **REPORTE DE PRODUCTOS POR GRUPOS DE TIPO DE PRODUCTO**

TIPO PRODUCTO	NOMBRE	MARCA	PRECIO COMPRA	PRECIO VENTA	STOCK
JABONES	JABON LIQUIDO	JHONSON BABY	125.00 \$	129.00 \$	231
	JABON PARA MANOS	AXION	52.00 \$	62.00 \$	-11
LIMPIEZA PISOS	DESINFECTANTE PISOS 250ML	DESENGRASANTE	15.00 \$	25.00 \$	10
PAPEL HIGIENICO	PAPEL HIGIENICO *6	FAMILIA	50.00 \$	60.00 \$	-8
	PAPEL HIGIENICO * 12	SOFT	28.00 \$	35.00 \$	52

Figura 50 Reporte de Stock de Productos



En la figura se puede observar el reporte de todos los productos que se encuentran en stock.

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios
Salir
Datos Personales
Mantenimientos Tablas
Clientes
Productos
Tipo Producto
Categorías
Garantías
Proveedores
Ordenes de Compra/Facturas
Generar Orden de Compra
Verificar Orden de Compra
Generar Factura
Verificar Factura
Registrar Entrada de Productos
Listar Entrada de Productos
Reportes
Stock Productos
Productos por Marcas
Ventas Diarias x Fechas

REPORTE DE VENTAS DEL DIA

BUSCAR FACTURA

Numero Venta	FECHA	Monto Total	DOCUMENTO	NOMBRES	Dni Ruc
V0001	26/05/2017	65,00	FACTURA	JULIO SALAS	70553988
V0002	27/05/2017	162,00	FACTURA	MARIA VEGA	54854554
V0003	13/06/2017	225,00	FACTURA	luiosd	32548978
V0004	15/06/2017	71,00	FACTURA	juan	65487594
V0005	07/09/2017	327,00	FACTURA	LOURDES	172106444
V0006	07/09/2017	2147,00	FACTURA	JORGE VIVANCO	17654654
V0007	07/09/2017	535,00	FACTURA	JULIAN	1321321
V0008	08/09/2017	2515,00	FACTURA	JULIAN BENITEZ	1721067369
V0009	08/09/2017	615,00	FACTURA	JUAN	16865654
V0010	08/09/2017	615,00	FACTURA	LUIS ANDRADE	17546546

[Crear filtro](#)

Figura 51 Reporte de Facturas

En la figura se puede observar el reporte de facturas generadas.



5.03 Especificación de pruebas de unidad

Tabla 26

PU001 Ingreso al Sistema

Identificador de la prueba	PU001
Método a probar	Ingreso al Sistema
Objetivo de la Prueba:	Determinar si el ingreso cumple con estándares de seguridad
Datos de entrada	Pantalla de Login del sistema Situación 1: Ingreso de usuario y contraseña correcto Situación 2: Ingreso de usuario y contraseña incorrecto
Resultado Esperado	Situación 1: Acceso autorizado presentación del sistema Situación 2: Mensaje indicando: LOGIN INCORRECTO
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 27

PU002 Registro de Usuarios

Identificador de la prueba	PU002
Método a probar	Registro de Usuarios
Objetivo de la Prueba:	Verificar si el sistema realiza el correcto ingreso de nuevos usuarios al sistema, adicional realizar la edición u corrección de datos de los mismos.
Datos de entrada	Hacer clic en la opción "Datos personales" del Menú Situación 1: Ingreso de nombre de usuario, seleccionar tipo de usuario, ingresar número de teléfono, contraseña, DNI, Nombres, apellidos, dirección y estado activo/inactivo. Situación 2: Seleccionar del listado el usuario que desea modificar y editar de acuerdo a criterio del administrador
Resultado Esperado	Situación 1: Al dar clic en el botón "Nuevo", verificar en el listado de usuarios si fue ingresado correctamente con los datos anteriormente descritos. Situación 2: Al dar clic en el botón "Editar", verificar en el listado de usuarios si fue editado correctamente el producto.
Comentarios	La prueba de unidad funciona correctamente al 100%



Tabla 28

PU003 Registro de Productos

Identificador de la prueba	PU003
Método a probar	Registro de Productos
Objetivo de la Prueba:	Verificar si el sistema realiza el correcto ingreso de nuevos productos al sistema, adicional realizar la edición u corrección de datos de los mismos.
Datos de entrada	Situación 1: Ingreso de nombre de producto, descripción, Precio Compra, Precio Venta y seleccionar marca y categoría. Situación 2: Seleccionar del listado el producto que desea modificar y editar de acuerdo a criterio del administrador
Resultados esperados	Situación 1: Al dar clic en el botón "Nuevo", verificar en el listado de usuarios si fue ingresado correctamente con los datos anteriormente descritos. Situación 2: Al dar clic en el botón "Editar", verificar en el listado de usuarios si fue editado correctamente el producto.
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 29

PU004 Registro de Marca de producto

Identificador de la prueba	PU004
Método a probar	Registro de Marca de producto
Objetivo de la Prueba:	Verificar si el sistema realiza el correcto ingreso de marcas de productos al sistema, adicional realizar la edición u corrección de datos de los mismos.
Datos de entrada	Situación 1: Ingreso de nombre de marca y seleccionar estado activo/inactivo. Situación 2: Seleccionar del listado la marca de producto que desea modificar y editar de acuerdo a criterio del administrador
Resultados esperados	Situación 1: Al dar clic en el botón "Nuevo", verificar en la marca de productos si fue ingresado correctamente con los datos anteriormente descritos. Situación 2: Al dar clic en el botón "Editar", verificar en el listado de usuarios si fue editado correctamente la marca del producto
Comentarios	La prueba de unidad funciona correctamente al 100%



Tabla 30

PU005 Registro de Categorías de productos

Identificador de la prueba	PU005
Método a probar	Registro de categorías de productos
Objetivo de la Prueba:	Verificar si el sistema realiza el correcto ingreso de nuevas categorías de productos al sistema, adicional realizar la edición u corrección de datos de los mismos.
Datos de entrada	Situación 1: Ingreso de nombre de categoría y seleccionar estado activo/inactivo. Situación 2: Seleccionar del listado la categoría que desea modificar y editar de acuerdo a criterio del administrador
Resultados esperados	Situación 1: Al dar clic en el botón "Nuevo", verificar en el listado de categorías si fue ingresado correctamente con los datos anteriormente descritos. Situación 2: Al dar clic en el botón "Editar", verificar en el listado de categorías si fue editado correctamente el producto.
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 31

Registro de proveedores

Identificador de la prueba	PU006
Método a probar	Registro de Proveedores
Objetivo de la Prueba:	Verificar si el sistema realiza el correcto ingreso de nuevos proveedores de productos al sistema, adicional realizar la edición u corrección de datos de los mismos.
Datos de entrada	Situación 1: Ingreso de nombre de empresa, nombre de contacto, RUC, teléfono, dirección, email y cuenta bancaria Situación 2: Seleccionar del listado la categoría que desea modificar y editar de acuerdo a criterio del administrador
Resultados esperados	Situación 1: Al dar clic en el botón "Nuevo", verificar en el listado de proveedores si fue ingresado correctamente con los datos anteriormente descritos. Situación 2: Al dar clic en el botón "Editar", verificar en el listado de proveedores si fue editado correctamente el producto.
Comentarios	La prueba de unidad funciona correctamente al 100%



Tabla 32

PU007 Generar Orden de Compra

Identificador de la prueba	PU007
Método a probar	Generar Orden de Compra
Objetivo de la Prueba:	Verificar si el sistema realiza la correcta generación de una orden de compra.
Datos de entrada	Situación 1: Buscar el nombre del proveedor, Colocar una descripción, indicar estado (enviado/activado), seleccionar del listado de productos los productos para la compra. Situación 2: Dar clic en el botón "Nuevo" Situación 3 : Dar clic en el botón "Editar"
Resultados esperados	Situación 1: Al dar clic en el botón "Guardar", inmediatamente aparece el reporte de la orden de compra con los ítems solicitados y el monto a cancelar Situación 2: Permitir la generación de una nueva orden de compra Situación 3: Permitir la edición de los productos de la orden de compra
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 33

PU008 Generar Factura

Identificador de la prueba	PU008
Método a probar	Generar factura
Objetivo de la Prueba:	Verificar si el sistema realiza la correcta generación de una factura.
Datos de entrada	Situación 1: Digitar el nombre del cliente, cedula o RUC, dirección, teléfono, seleccionar los productos para la factura y hacer clic en el botón "Guardar" Situación 2: Dar clic en el botón "Nuevo" Situación 3 : Dar clic en el botón "Editar"
Resultados esperados	Situación 1: Al dar clic en el botón "Guardar", inmediatamente aparece el reporte de la factura con los ítems solicitados y el monto a cancelar Situación 2: Permitir la generación de una nueva factura Situación 3: Permitir la edición de los productos de la factura
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 34*Listar Orden de Compra PU009*

Identificador de la prueba	PU009
Método a probar	Listar Orden de Compra
Objetivo de la Prueba:	Verificar si el sistema permite visualizar las órdenes de compra generados
Datos de entrada	Hacer clic en la opción "Listar Orden de Compra" del menú Visualizar la orden de compra verificando si los datos fueron ingresados correctamente
Resultados esperados	Imprimir la orden de compra seleccionada Descargar la orden de compra en formato PDF
Comentarios	La prueba de unidad funciona correctamente al 100%

Tabla 35*PU010 Listar Factura*

Identificador de la prueba	PU010
Método a probar	Listar Factura
Objetivo de la Prueba:	Verificar si el sistema permite visualizar las facturas generadas.
Datos de entrada	Hacer clic en la opción "Listar Factura" del menú Visualizar la orden de compra verificando si los datos fueron ingresados correctamente
Resultados esperados	Imprimir la factura seleccionada Descargar la factura en formato PDF
Comentarios	La prueba de unidad funciona correctamente al 100%



5.04 Especificación de pruebas de carga

Tabla 36

PC001 Facturación

Identificador de la prueba	PC001
Tipo de Prueba	Facturación al mismo tiempo
Objetivo de la prueba	Validar el tiempo de respuesta de las diferentes peticiones de varios usuarios
Descripción	De acuerdo al programa "JMETER" se realizó la prueba indicando los siguientes resultados. Numero de hilos: 10 Periodo de tiempo: 6 segundos Cantidad de bucle: 1
Resultados Esperados	
Determinar si los tiempos de transacción son eficientes.	
Comentarios	Los tiempos de transacción tiene un porcentaje valido

Tabla 37

Subida de información

Identificador de la prueba	PC002
Tipo de Prueba	Subida masiva de información
Objetivo de la prueba	Determinar el tiempo de respuesta de la subida masiva de información
Descripción	De acuerdo al programa "JMETER" se realizó la prueba indicando los siguientes resultados. Numero de hilos: 10 Periodo de tiempo: 18 segundos Cantidad de bucle: 3
Resultados Esperados	
Determinar la carga en el sistema con el proceso de subida masiva de información	
Comentarios	El sistema no tuvo inconvenientes en realizar la carga de información



5.05 Configuración del ambiente mínima/ideal

El aplicativo se encuentra diseñado bajo arquitectura cliente-servidor por tanto como características recomendadas en cuanto a hardware para un ambiente recomendado se puede considerar lo siguiente:

Servidor

- Windows Server 2008 u superior
- SQL Server 2008 u 2012
- Procesador de 3,1 GHZ de 64 bits
- Memoria (RAM) de 16GB
- Disco duro de 200GB

Cliente

- Procesador DualCore de 1,4GHZ o superior
- Memoria (RAM) 4GB
- Disco Duro de 160 GB
- Conexión a red con acceso a correo electrónico

Como ambiente ideal se puede considerar lo siguiente:

- Asignación de roles de trabajo a cada uno de los usuarios
- Capacitación para el correcto uso del sistema
- Buen gestor de base de datos que mantenga la integridad de los mismos
- Equipos que se desempeñen eficazmente
- Red correctamente estructurada



Capítulo VI: Aspectos Administrativos

6.01 Recursos

En la fase de planificación de un proyecto es importante destacar los recursos a utilizar para determinar con información precisa el objetivo con el cual vamos a utilizar. Se considera el recurso humano, material y tecnológico.

6.01.01 Recurso Humano

Tabla 38

Recurso Humano

Humano	Nombre	Actividad	Responsabilidad
Promotor del proyecto	Vera	Desarrollo del	Proponer, Analizar, diseñar e
	Jaramillo	proyecto	implementar la sistematización del
	Kevin		proceso de facturación de la
	Estuardo		empresa PROVEASEO.
Tutor	Prof. Luis	Guiar el	Hacer un seguimiento al estudiante
	Ríos C.	desarrollo del	corrigiendo y guiando la
		proyecto	documentación del proyecto.
Lector		Revisar el	Que la documentación del
		desarrollo del	proyecto cumpla con las normas
		proyecto	establecidas.



6.01.02 Recurso Material

Tabla 39

Recurso Material

RECURSO MATERIAL			
Material	Cantidad	Valor Unitario	Valor Total
Computador	1	1	900
Hojas Papel Bond	450	0,03	13,50
Esfero	2	0,35	0,70
Lápiz	1	0,40	0,40
Borrador	1	0,10	0,10
Folder	1	0,35	0,35

6.01.03 Recurso Tecnológico

- Computador Portátil
- Sistema Operativo Windows 10
- Visual Studio 2013
- Crystal Reports
- Base de datos SQL Server 2012
- Gestor SQL Server Management Studio
- Sybase Power Designer
- Rational Rose
- Paquete de Office
- Conexión a Internet



6.02 Presupuesto

Tabla 40

Presupuesto

RECURSO MATERIAL			
Material	Cantidad	Valor Unitario	Valor Total
Computador	1	1	900
Hojas Papel Bond	450	0,03	13,5
Esfero	2	0,35	0,7
Lápiz	1	0,4	0,4
Borrador	1	0,1	0,1
Folder	1	0,35	0,35
Total Materiales			915,05
SERVICIOS			
Proceso de titulación	6 meses	804	804
Internet	6 meses	25	150
Transporte	6 meses	50	300
Alimentación	6 meses	12	72
Semestre	Sexto Nivel	650,25	650,25
Total Servicios			1976,25
Costo Total			2891,3

6.03 Cronograma

(Ver Anexo A.04)



Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones

- La sistematización del proceso de facturación ha permitido a la empresa PROVEASEO reducir tiempos considerables, optimizando los procesos y manejando información validada.
- La automatización de los procesos de compra y venta de productos, permite un control adecuado y disminuye pérdidas tanto en productos como en ventas
- La generación de reportes permitirá tomar decisiones a la gerencia, gracias a la información brindada por el aplicativo a la cual puede acceder en cualquier momento.
- El ambiente de trabajo será más óptimo ya que se evitará confrontaciones en cuanto a pérdidas de facturas, compras o stock de productos.
- El manejo adecuado de las herramientas tecnológicas permitió cumplir con los requerimientos solicitados, los usuarios podrán desempeñarse con mayor agilidad y transparencia.
- Las interfaces mantienen aspectos interactivos los cuales permite mayor fluidez y desenvolvimiento del mismo.
- La decisión de la empresa PROVEASEO en implementar un aplicativo le permitirá tener un crecimiento a nivel de pymes.



7.02 Recomendaciones

- El usuario designado como administrador debe tener el conocimiento adecuado para que los procesos fluyan y el sistema mantenga su correcto funcionamiento
- Entregar manuales de políticas a los usuarios en cuanto al uso del software
- Capacitar a los diferentes usuarios de todas las prestaciones que ofrece el software
- El sistema ha sido desarrollado para brindar mayores implementaciones a futuro, por lo cual se podrá añadir módulos que permitan sistematizar otros procesos
- Antes de realizar órdenes de compra siempre se debe verificar el stock actual y luego confirmar con existencia física. Evitará errores en conteos al finalizar cada mes.



ANEXOS



Anexo A.0 1 Matriz de Involucrados

Actores involucrados	Interés sobre el problema central	Problemas percibidos	Recursos, mandatos y capacidades	Intereses sobre el proyecto	Conflictos potenciales
Clientes	Mejorar procesos de entrega y recepción de facturas	Mala entrega de facturas	Recursos humanos, técnicos y tecnológicos	Eficiente control en la entrega y recepción de facturas	Falta de conocimiento en recursos informáticos
Administrador	Reducir tiempos y manejo de usuarios.	Falta de información al instante y pérdida de documentos	Recursos humanos Recursos tecnológicos	Agilidad en emisión de reportes a gerencias, reducción de tiempo	Mal ingreso de información por parte de los empleados.
Gerencia	Conocer los movimientos y estado financiero de sus productos.	Retraso en la generación de reporte para tomas de decisiones	Recurso Financiero Recurso Humano Recurso Tecnológico	Obtener eficiencia en cuanto a reportes con información integra	Entrega de documentos de forma manual.
Contadores	Seguridad en el manejo de la información Automatizar los procesos contables	Redundancia de información	Recurso humano	Apoyar a la necesidad de la empresa en mejorar los procesos	Capacitación en cuanto al uso y manejo del aplicativo
Proveedores	Agilidad en el proceso de emisión de documentos	Demora al generar los documentos	Recurso Humano Recurso Tecnológico	Agilidad para el pago de facturas	Desconocimiento de los sistemas informáticos
Ejecutor del proyecto	Solucionar los problemas planteados por los usuarios, sistematizando el proceso de facturación	Falta de seguridades Redundancia de información Inadecuado manejo de datos Proceso de facturación manual	Recurso Humano	Satisfacer las necesidades de los usuarios manejando procesos sistematizados	Tiempo para la ejecución del sistema y su culminación



Anexo A.0 2 Matriz de Impacto de Objetivos

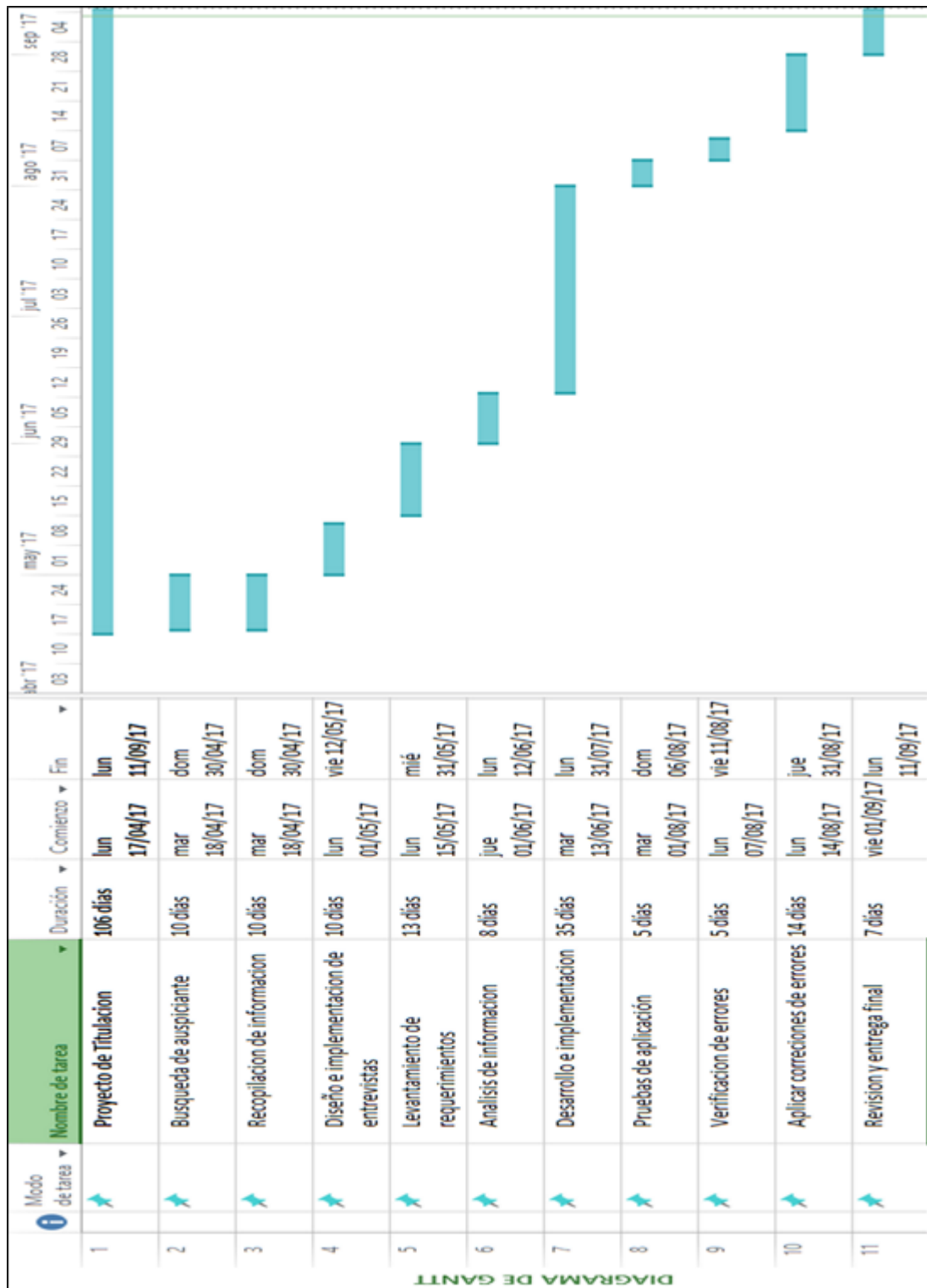
Objetivo	Factibilidad de lograrse	Impacto Ambiental	Relevancia	Sostenibilidad	Total	Categoría
Controlar las ventas adecuadamente con registro de ingresos	Incrementar los ingresos en la empresa (4)	Desarrolla procesos organizados (4)	Satisfacer las necesidades de los clientes (4)	Implementar nuevos procesos a futuro (4)	16	Medio Alto
Crear controles para el inventario de productos	Evidenciar falencias para dar solución (4)	Eliminar conflictos entre personal (5)	Mejorar el desarrollo de procesos (4)	Mejora el ambiente laboral (4)	17	Medio Alto
Mejorar tiempos de respuesta en el proceso de facturación	Mejorar el desarrollo de procesos actuales (4)	Mantener el rol del negocio (4)	Evitar accidentes laborales (5)	Mejora en el intelecto de los empleados (5)	18	Medio Alto
Entrega de reportes de manera óptima	Optimizar tiempo humano en actividades planificadas (4)	Identificar posibles soluciones en el transcurso de actividades (4)	Seguir los procesos de operación mediante una planificación (5)	Capacitar al personal sobre los nuevos procesos para mayor eficacia (4)	17	Medio Alto
Mejorar servicio al cliente	Los beneficios del sistema son mayores a los costos (4)	Incentiva el desarrollo de la empresa (4)	Eficiencia en el rol del negocio (4)	Reducción de tiempo y conformidad de atención (4)	16	Medio Alto
	20 puntos	21 puntos	22 puntos	21 puntos	84	Alta
					PUNTOS	



Anexo A.0 3 Matriz de Marco Lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
FIN DEL PROYECTO Manejo de información consistente de manera ágil, con seguridad confiabilidad e integridad	Inexistencia de Información redundante Disminución de errores	Encuestas al personal involucrado Inspecciones para medir la factibilidad de la empresa	Falta de compromiso del personal para cumplir con los procesos de manera correcta Entrega de información concisa
PROPOSITO DEL PROYECTO Automatizar procesos para alcanzar mayor productividad Eficiente control de existencias de mercadería	Manejo de información segura Mejora en la agilidad y control de procesos Optimización en tiempos de respuesta	Emisión de reportes El sistema realizara reportes para constancia de datos	Mal uso de la tecnología
COMPONENTES DEL PROYECTO Controlar las ventas adecuadamente con registro de ingresos Crear controles de inventario de productos Efectuar actividades con mayor rapidez	Elaboración de balance general de acuerdo a los datos registrados Inventarios correctos según el registro Perdidas de información reducidas Manejo de información confiable y segura	Encuestas al personal involucrado Registros de ventas realizados Aumento de rapidez en la toma de decisiones por parte de gerencia en la compra de productos	Inadecuada capacitación al personal Pérdida de información Mal uso de la tecnología
ACTIVIDADES DEL PROYECTO Realizar controles y capacitaciones para el correcto uso del aplicativo Realizar el estudio correcto para los posibles ingresos de registros Registrar ingresos y egresos de artículos Seleccionar datos para el ingreso de facturas Generar correctamente los pedidos de mercadería	Registro de usuarios Registro de artículos Emisión de facturas Emisión de reportes	Encuesta al personal involucrado Satisfacción de parte del personal y del cliente	Falta de apoyo en la adaptación a los nuevos procesos

Anexo A.0 4 Cronograma





MANUAL DE INSTALACIÓN



INDICE

Contenido	Página
Índice de Figuras.....	82
Instalación de SQL Server 2012.....	83
Instalación de Visual Studio 2013.....	85

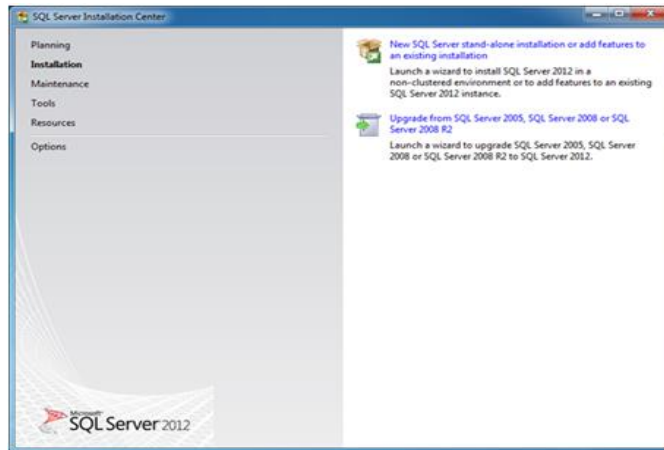


Índice de Figuras

<i>Manual de Instalación Figura 1 Pantalla de inicio.....</i>	<i>77</i>
<i>Manual de Instalación Figura 2 Interfaz de Instalacion</i>	<i>77</i>
<i>Manual de Instalación Figura 3 Instancia del motor de base de datos.....</i>	<i>78</i>
<i>Manual de Instalación Figura 4 Finalizacion de la instalación.....</i>	<i>78</i>
<i>Manual de Instalación Figura 5 Instalación de Visual Studio</i>	<i>79</i>
<i>Manual de Instalación Figura 6 Características de Visual Studio</i>	<i>79</i>
<i>Manual de Instalación Figura 7 Finalización de Instalación de Visual Studio</i>	<i>80</i>

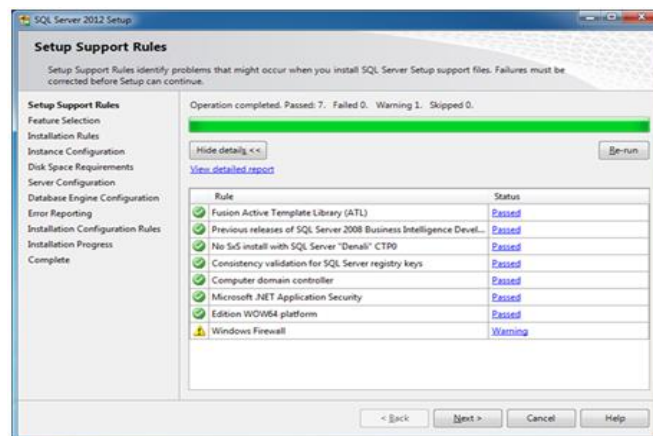
1.2 Instalación Microsoft SQL Server 2012 R2 Express

- Procedemos con la descarga de la edición Express desde la página de Microsoft en el siguiente link <https://www.microsoft.com/en-us/download/details.aspx?id=29062>, luego de descargado el archivo, se ejecuta el programa:



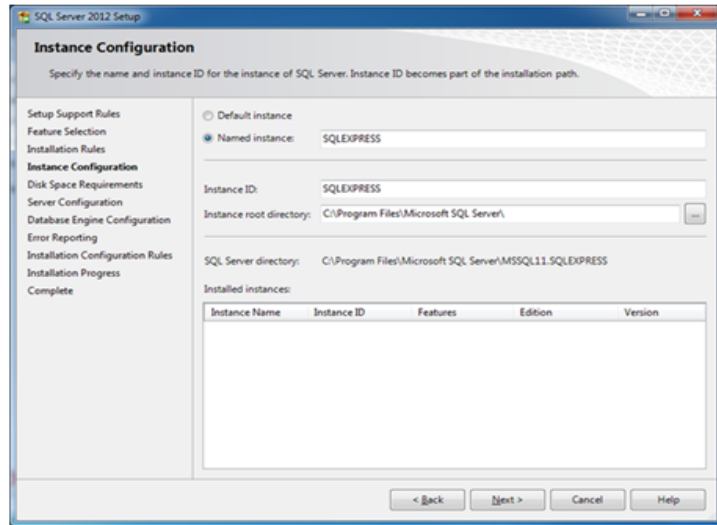
Manual de Instalación Figura 1 Pantalla de inicio

- Seleccionar la primera opción del panel de la derecha, marcar la aceptación de los términos de licenciamiento y opcionalmente, marcar la casilla de enviar datos hacia Microsoft. Presionar Next y esperar unos minutos. Aparecerá esta ventana:



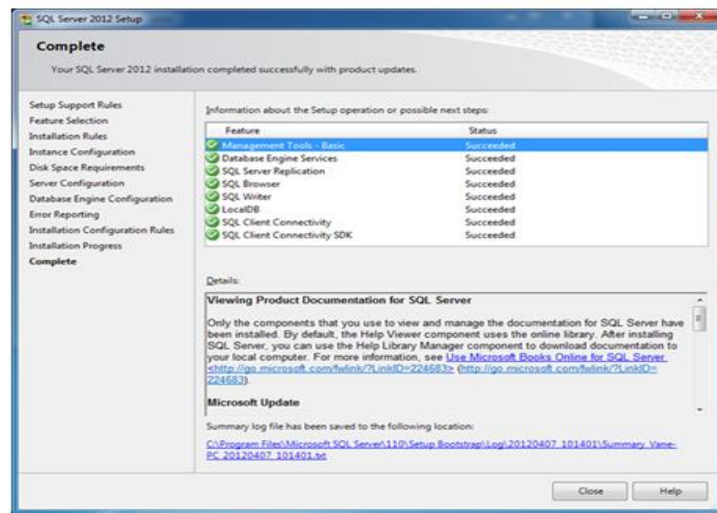
Manual de Instalación Figura 2 Interfaz de Instalación

- Digitamos la instancia del motor de la base de datos SQLEXPRESS



Manual de Instalación Figura 3 Instancia del motor de base de datos

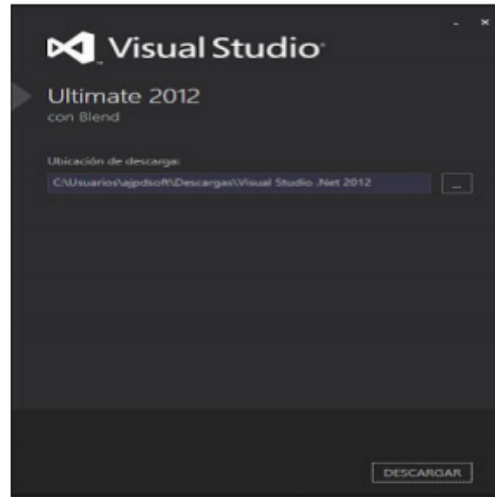
- Finalizamos la instalación y reiniciamos el pc para que se guarden los cambios



Manual de Instalación Figura 4 Finalización de la instalación

1.3 Instalación Visual Studio 2012

- Para instalar Visual Studio 2012 descargamos el programa del siguiente link [https://msdn.microsoft.com/en-us/library/dd831853\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/dd831853(v=vs.120).aspx), ejecutamos el instalador y nos aparecerá la siguiente pantalla



Manual de Instalación Figura 5 Instalación de Visual Studio

- Seleccionamos las características que para instalar y sus complementos.



Manual de Instalación Figura 6 Características de Visual Studio

- Finalizamos la instalación y reiniciamos la PC para que los cambios surtan efecto



Manual de Instalación Figura 7 Finalización de Instalación de Visual Studio



MANUAL DE USUARIO



Contenido	Página
Índice de Figuras	83
1.1 Ingreso al Aplicativo	84
1.2 Interfaz Usuarios	85
1.3 Interfaz Productos.....	86
1.4 Mantenimiento Tablas	88
1.5 Generar Orden de Compra.....	89
1.6 Verificar Orden de Compra	90
1.7 Generar Factura	91
1.8 Verificar Factura.....	92
1.9 Reporte de Artículos.....	92
1.10 Reporte de Ventas	93



Índice de Figuras

<i>Manual de Usuario Figura 1: Ingreso al Aplicativo</i>	84
<i>Manual de Usuario Figura 2: Login Incorrecto</i>	84
<i>Manual de Usuario Figura 3: Interfaz Usuarios</i>	85
<i>Manual de Usuario Figura 4 Interfaz Productos</i>	86
<i>Manual de Usuario Figura 5: Interfaz Marca Producto</i>	88
<i>Manual de Usuario Figura 6: Interfaz Categoría</i>	88
<i>Manual de Usuario Figura 7: Interfaz Proveedores</i>	89
<i>Manual de Usuario Figura 8 Generar Orden de Compra.....</i>	89
<i>Manual de Usuario Figura 9: Verificar Orden de Compra</i>	90
<i>Manual de Usuario Figura 10: Interfaz Generar Factura</i>	91
<i>Manual de Usuario Figura 11: Verificar Factura.....</i>	92
<i>Manual de Usuario Figura 12 Reporte de Productos en Stock.....</i>	92
<i>Manual de Usuario Figura 13 Reporte de Ventas</i>	93

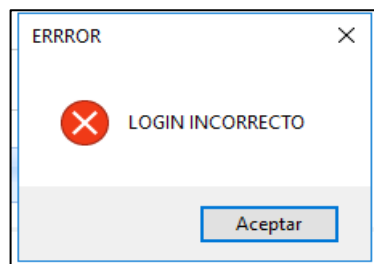
1.1 Ingreso al Aplicativo



Manual de Usuario Figura 1: Ingreso al Aplicativo

En la pantalla principal del sistema procedemos a ingresar en los campos correspondientes, nombre de usuario y password designado por el administrador del sistema.

En caso de digitar de manera errónea, se mostrará el siguiente mensaje de error, favor verificar bien sus credenciales de ingreso.



Manual de Usuario Figura 2: Login Incorrecto



1.2 Interfaz Usuarios

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantías

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Mantenimiento Usuarios

USUARIO: CONTRASEÑA: NOMBRE-APELLIDOS:

TIPO: **ADMINISTRADOR** Nro DNI: DIRECCION:

TELEFONO: ESTADO: ACTIVO INACTIVO

Listado Usuarios

Usuario	Contraseña	Nombre	Nombre_Apellidos	DNI	Telefono	Dirección	Estado
JOSE	123456	ADMINISTRADOR	JOSE LOPEZ	46012455	945231130	CHILLOGALLO	ACTIVO
KEVIN	12345	ADMINISTRADOR	KEVIN VERA	70553988	955361753	FLORIDA	ACTIVO
ADRIANA	12345	JEFE DE ALMACEN	ADRIANA SEGARRA	1564654	165655	LA FORESTAL	ACTIVO
USER1	12345	CLIENTE	USER1001	834634	17213213	LA FORESTAL	INACTIVO

[Crear filtro](#)

Manual de Usuario Figura 3: Interfaz Usuarios

Una vez ingresado al aplicativo, damos clic en la primera opción del submenú: Datos Personales, se nos desplegará la interfaz de mantenimiento de usuarios.

Nota: Para acceder a este módulo es necesario que el usuario sea Administrador

Para crear nuevo usuario:

1. Dar clic en el botón "Nuevo"
2. Llenamos los campos con los datos del usuario (Nombre Usuario, Password, Nombres-Apellidos, Cedula, Dirección, Teléfono)
3. Seleccionamos el tipo de usuario: Administrador-Usuario
4. Presionar en el botón "Guardar"
5. Una vez creado el usuario aparecerá en el menú "Listado de Usuarios"

Para editar usuarios:

1. En el menú “Listado de Usuarios” Seleccionamos el usuario que requerimos editar
2. Modificamos los campos del usuario (Nombre Usuario, Password, Nombres- Apellidos, Cedula, Dirección, Teléfono)
3. Presionamos en el botón: “Editar”
4. Una vez editado el usuario aparecerá en el menú “Listado de Usuarios”

1.3 Interfaz Productos

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR -KEVIN VERA

MANTENIMIENTO DE PRODUCTOS DE ALMACEN

INGRESO DE DATOS

NOMBRE: MARCA: DESCRIPCION:

TIPO: STOCK: PRECIO COMPRA: PRECIO VENTA:

ESTADO: ACTIVO INACTIVO

LISTADO - BUSQUEDA DE DATOS

NOMBRE	TIPO	MARCA	PRECIO_COMPRA	PRECIO_VENTA	STOCK	DESCRIPCION	ESTADO
JABON LIQUIDO	JABONES	JHONSON BABY	125,00	129,00	231	JHONSON BABY 250ML	ACTIVO
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA	50,00	60,00	-8	PAPEL HIGIENICO *6	ACTIVO
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT	28,00	35,00	52	PAPEL HIGIENICO * 12	ACTIVO
JABON PARA MANOS	JABONES	AXION	52,00	62,00	-11	JABON PARA MANOS 500G	ACTIVO
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE	15,00	25,00	10	DESINFECTANTE PISOS 250ML FRAGANCIA LIMON	ACTIVO

[Crear filtro](#)

Manual de Usuario Figura 4 Interfaz Productos

Para acceder al módulo de Productos, damos clic en la opción del submenú:

Productos, se nos desplegara la interfaz de Mantenimiento de Productos



Para crear nuevo producto:

1. Dar clic en el botón "Nuevo"
2. Llenamos los campos con los datos del producto (Nombre producto, Descripción, Precio Compra, Precio Venta)
3. Seleccionamos la marca de producto, tipo de producto.
4. Presionar en el botón "Guardar"
5. Una vez creado el producto aparecerá en el menú "Listado de Productos"

Para editar productos:

1. En el menú "Listado de productos" Seleccionamos el producto que queremos editar
2. Modificamos los campos del producto (Nombre producto, Descripción, Precio Compra, Precio Venta)
3. Presionamos en el botón: "Editar"
4. Una vez editado el usuario aparecerá en el menú "Listado de Productos"

Para el menú de (Tipo Producto, Categorías, Garantía y Proveedores), se mantiene la misma secuencia para la creación u edición de los campos correspondientes.

1.4 Mantenimiento Tablas

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR -KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantias

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Productos por Marcas

MANTENIMIENTO MARCA DEL PRODUCTO

INGRESO DE DATOS

NOMBRE: Nombre Marca Producto

ESTADO: ACTIVO INACTIVO

NUEVO GUARDAR EDITAR

LISTADO - BUSQUEDA DE DATOS

NOMBRE	ESTADO
JHONSON BABY	1
SOFT	1
FAMILIA	1
DESENGRASANTE	1
AXION	1

Crear filtro

Manual de Usuario Figura 5: Interfaz Marca Producto

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR -KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantias

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Productos por Marcas

MANTENIMIENTO TIPO DE PRODUCTO

INGRESO DE DATOS

NOMBRE: Nombre Tipo Producto

ESTADO: ACTIVO INACTIVO

NUEVO GUARDAR EDITAR

LISTADO - BUSQUEDA DE DATOS

NOMBRE	ESTADO
AMBIENTALES	1
JABONES	1
PAPEL HIGIENICO	1
LIMPIEZA PISOS	1

Crear filtro

Manual de Usuario Figura 6: Interfaz Categoría



MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantías

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

Productos por Marcas

MANTENIMIENTO DE PROVEEDORES

INGRESO DE DATOS

NOMBRE EMPRESA: Identificación Empresa RUC: 11 dígitos del 0-9 DIRECCION: Lugar donde se encuentra ubicada la empresa

NOMBRE CONTACTO: Nombres y Apellidos TELEFONO: 9 dígitos del 0-9 EMAIL: Correo electrónico para comunicarse

ESTADO: ACTIVO INACTIVO CUENTA BANCARIA: Banco - Numero de Cuenta

LISTADO PROVEEDORES

NOMBRE_COMPANIA	NOMBRE_CONTACTO	RUC	BANCARIO	TELEFONO	DIRECCION	EMAIL	ESTADO
PYDACO	JORGE ANDRADE	45487884545	Bco Pichincha - 57585525	099956565	CARAPUNGO		ACTIVO
SUPERUNO	LUIS GOMEZ	54845454545	Bco Guayaquil - 25465462	099955413	CALDERON		ACTIVO
ZAIMELLA	ANITA SAQUILANDA	48454845454	Bco Pacifico - 2565467	099975454	EL PINTADO		ACTIVO
COANOTV	HERNAN GONZALEZ	8412945246	Bco Pichincha - 2564654	099925411	LA MANGRALANA		INACTIVO

[Crear filtro](#)

Manual de Usuario Figura 7: Interfaz Proveedores

1.5 Generar Orden de Compra

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorías

Garantías

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

REGISTRO DE ORDEN DE COMPRA

INGRESO LISTADO

Orden de Compra

AUTORIZADO: KEVIN VERA -USUARIO: KEVIN

PROVEEDOR:

DESCRIPCION: Comentario alternativo

ESTADO: ENVIADO ACTIVADO ELIMINADO

Arrastre una columna aquí...

Sin datos para mostrar

MONTO TOTAL:

Buscar Producto

NOMBRE	TIPO	MARCA
JABON LIQUIDO	JABONES	JHONSON BABY
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT
JABON PARA MANOS	JABONES	AXION
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE

[Crear filtro](#)

Manual de Usuario Figura 8 Generar Orden de Compra

Para generar la orden de compra a proveedor, damos clic en el boton “Generar orden de compra” del menu

1. Seleccionar el icono de lupa que se encuentra en la parte derecha del campo de proveedor, se desplegara una lista para seleccionar el proveedor que desea generar la orden.

2. En el campo Descripción, se coloca un motivo de generación de compra.
3. Seleccionar el tipo de Estado: Enviado-Activado
4. En el listado "Buscar Producto" Seleccionar los productos que desea adquirir
5. Se desplegará un menú solicitando que cantidad requiere del producto.
6. Una vez ingresado, verificar el monto total.
7. Dar clic en el botón Guardar

1.6 Verificar Orden de Compra



Carcelen Alto Av Republica Dominicana y Sebastian Ortiz Telefono: 2506134 - Cristian Cevallos

 **PROVEASEO**

R.U.C 20120302515

ORDEN COMPRA

C0001

NOMBRE PROVEEDOR : PYDACO **RUC:** 45487884545

AUTORIZADO : JOSE LOPEZ 02/04/2017

CANT.	NOMBRE	TIPO	MARCA	PRECIO	IMPORTE
10	PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA	50.00	500.00
				MONTO T.	500.00

Manual de Usuario Figura 9: Verificar Orden de Compra

Una vez realizada la orden de compra, se desplegará la orden de compra generada posterior para envío u impresión.

1.7 Generar Factura



The screenshot shows the 'MODULO DE FACTURACION' interface. The user is 'KEVIN VERA' with the role 'ADMINISTRADOR'. The main area is titled 'Nueva Factura' and includes a form for creating a new invoice. The form is pre-filled with 'AUTORIZADO: KEVIN VERA -USUARIO: KEVIN'. Below this, there are input fields for 'NOMBRES' (with a placeholder 'Datos personales del comprador') and 'DNI/RUC' (with a placeholder 'Numero Dni / Ruc'). There are three buttons: 'NUEVO', 'GUARDAR', and 'EDITAR'. Below the buttons is a table with the header 'Arrastre una columna aquí...' and the text 'Sin datos para mostrar'. At the bottom right, there is a 'MONTO TOTAL:' field with a yellow background.

On the right side, there is a 'Buscar Producto' section with a table of products:

NOMBRE	TIPO	MARCA
JABON LIQUIDO	JABONES	JHONSON BABY
PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA
PAPEL HIGIENICO * 12	PAPEL HIGIENICO	SOFT
JABON PARA MANOS	JABONES	AXION
DESINFECTANTE PISOS 250ML	LIMPIEZA PISOS	DESENGRASANTE

There is a 'Crear filtro' link at the bottom of the table.

Manual de Usuario Figura 10: Interfaz Generar Factura

Para generar una nueva factura, damos clic en el botón “Generar Factura” del menú.

1. En la ventana de generar factura, damos clic en el botón “Nuevo”.
2. Colocamos el nombre del cliente y su DNI
3. Buscamos los productos que desea adquirir el cliente en el listado “Buscar Producto”
4. Cada vez que seleccionemos el producto se desplegara una ventana indicando la cantidad que desea facturar.
5. Una vez seleccionado los productos, damos clic en el botón “Guardar”



1.8 Verificar Factura

MODULO DE FACTURACION

USUARIO: KEVIN TIPO: ADMINISTRADOR - KEVIN VERA

REPORTE DE DOCUMENTO DE VENTA

Carcelen Alto Av Republica Dominicana y Sebastian Ortiz Telefono: 2506134 - Cristian Cevallos

R.U.C 20120302515

FACTURA
V0001

PROVEASEO

NOMBRES JULIO SALAS **DNI-RUC** 70553988 **5/28/2017 12:00:00 AM**

CANT.	PRODUCTO	TIPO	MARCA	PRECIO U.	TOTAL	GARANTIA	PRECIO G.	IMPORTE
1	PAPEL HIGIENICO *6	PAPEL HIGIENICO	FAMILIA	60.00	50.00	GARANTIA 1 AÑO	15.00	65.00
MONTO T.								65.00

Carcelen Alto Av Republica Dominicana y Sebastian Ortiz Telefono: 2506134 - Cristian Cevallos

FIRMA

Manual de Usuario Figura 11: Verificar Factura

Una vez generada la factura se despliega la factura generada para su posterior envío u impresión.

1.9 Reporte de Articulos

Mantenimientos Usuarios

Salir

Datos Personales

Mantenimientos Tablas

Cientes

Productos

Tipo Producto

Categorias

Garantias

Proveedores

Ordenes de Compra/Facturas

Generar Orden de Compra

Verificar Orden de Compra

Generar Factura

Verificar Factura

Registrar Entrada de Productos

Listar Entrada de Productos

Reportes

Stock Productos

TIPO PRODUCTO MARCA PRODUCTO

PROVEASEO
Productos de Limpieza **REPORTE DE PRODUCTOS POR GRUPOS DE TIPO DE PRODUCTO**

TIPO PRODUCTO	NOMBRE	MARCA	PRECIO COMPRA	PRECIO VENTA	STOCK
JABONES	JABON LIQUIDO	JHONSON BABY	125.00 \$	129.00 \$	231
	JABON PARA MANOS	AXION	52.00 \$	62.00 \$	-11
LIMPIEZA PISOS	DESINFECTANTE PISOS 250ML	DESENGRASANTE	15.00 \$	25.00 \$	10
PAPEL HIGIENICO	PAPEL HIGIENICO *6	FAMILIA	50.00 \$	60.00 \$	-8
	PAPEL HIGIENICO *12	SOFT	28.00 \$	35.00 \$	52

Manual de Usuario Figura 12 Reporte de Productos en Stock



Para acceder al reporte de artículos, damos clic en el botón “Stock Productos”, se desplegará un listado con los productos en Stock

1.10 Reporte de Ventas

REPORTE DE VENTAS DEL DIA					
BUSCAR FACTURA					
Numero Venta	FECHA	Monto Total	DOCUMENTO	NOMBRES	Dni Ruc
V0001	26/05/2017	65,00	FACTURA	JULIO SALAS	70553988
V0002	27/05/2017	162,00	FACTURA	MARIA VEGA	54854554
V0003	13/06/2017	225,00	FACTURA	ludios	32548978
V0004	15/06/2017	71,00	FACTURA	juan	65487594
V0005	07/09/2017	327,00	FACTURA	LOURDES	172106444
V0006	07/09/2017	2147,00	FACTURA	JORGE VIVANCO	17654654
V0007	07/09/2017	535,00	FACTURA	JULIAN	1321321
V0008	08/09/2017	2515,00	FACTURA	JULIAN BENITEZ	1721067369
V0009	08/09/2017	615,00	FACTURA	JUAN	16865654
V0010	08/09/2017	615,00	FACTURA	LUIS ANDRADE	17546546

[Crear filtro](#)

Manual de Usuario Figura 13 Reporte de Ventas

Para acceder al reporte de ventas, damos clic en el botón “Reporte de Ventas”, se nos desplegara un listado con todas las ventas realizadas, se puede crear en la opción “Crear Filtro” para ordenar de acuerdo a criterio del usuario



MANUAL TECNICO



Contenido

Página

1.1 Diccionario de Datos.....	93
1.2 Programación.....	98
Clase Usuario.....	98
Clase Venta.....	109
Clase Productos.....	119
Clase Orden de Compra.....	129
Clase Proveedor.....	135
1.3 Script Base de Datos.....	144

1.1 Diccionario de Datos

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
13	KVFACTURACION	GARANTIA	Id_Garantia	NO	int	NULL
14	KVFACTURACION	GARANTIA	Nombre	NO	varchar	30
15	KVFACTURACION	GARANTIA	Descripcion	YES	varchar	50
16	KVFACTURACION	GARANTIA	Precio	YES	decimal	NULL
17	KVFACTURACION	GARANTIA	Estado	NO	varchar	10

Figura 52 Diccionario de Datos: Tabla Garantía

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
18	KVFACTURACION	MARCA_PRODUCTO	Id_Marca	NO	int	NULL
19	KVFACTURACION	MARCA_PRODUCTO	Nombre	NO	varchar	30
20	KVFACTURACION	MARCA_PRODUCTO	Estado	NO	tinyint	NULL

Figura 53 Diccionario de Datos: Tabla Marca_Producto

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
21	KVFACTURACION	ORDEN_COMPRA	IdOrdenCompra	NO	int	NULL
22	KVFACTURACION	ORDEN_COMPRA	IdProveedor	NO	int	NULL
23	KVFACTURACION	ORDEN_COMPRA	IdUsuario	NO	int	NULL
24	KVFACTURACION	ORDEN_COMPRA	NumeroOrdenCompra	NO	varchar	10
25	KVFACTURACION	ORDEN_COMPRA	MontoTotal	NO	decimal	NULL
26	KVFACTURACION	ORDEN_COMPRA	Fecha	NO	date	NULL
27	KVFACTURACION	ORDEN_COMPRA	Estado	NO	varchar	10
28	KVFACTURACION	ORDEN_COMPRA	Descripcion	YES	varchar	50

Figura 54 Diccionario de Datos: Tabla Orden_Compra

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
29	KVFACTURACION	PRODUCTO	IdProducto	NO	int	NULL
30	KVFACTURACION	PRODUCTO	Nombre	NO	varchar	40
31	KVFACTURACION	PRODUCTO	Descripcion	YES	varchar	50
32	KVFACTURACION	PRODUCTO	Id_TipoProducto	NO	int	NULL
33	KVFACTURACION	PRODUCTO	Id_Marca	NO	int	NULL
34	KVFACTURACION	PRODUCTO	Precio_Compra	NO	decimal	NULL
35	KVFACTURACION	PRODUCTO	Precio_Venta	NO	decimal	NULL
36	KVFACTURACION	PRODUCTO	Stock	NO	int	NULL
37	KVFACTURACION	PRODUCTO	Estado	NO	varchar	10

Figura 55 Diccionario de Datos: Tabla Producto



	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
38	KVFACTURACION	PROVEEDOR	IdProveedor	NO	int	NULL
39	KVFACTURACION	PROVEEDOR	NombreCompañia	NO	varchar	30
40	KVFACTURACION	PROVEEDOR	NombreContacto	NO	varchar	30
41	KVFACTURACION	PROVEEDOR	Ruc	NO	char	11
42	KVFACTURACION	PROVEEDOR	Bancario	YES	varchar	50
43	KVFACTURACION	PROVEEDOR	Telefono	YES	char	9
44	KVFACTURACION	PROVEEDOR	Direccion	YES	varchar	50
45	KVFACTURACION	PROVEEDOR	Email	YES	varchar	50
46	KVFACTURACION	PROVEEDOR	Estado	NO	varchar	15

Figura 56 Diccionario de Datos: Tabla Proveedor

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
47	KVFACTURACION	TIPO_DOCUMENTO	IdTipoDocumento	NO	int	NULL
48	KVFACTURACION	TIPO_DOCUMENTO	Nombre	NO	varchar	20

Figura 57 Diccionario de Datos: Tabla Tipo_Documento

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
49	KVFACTURACION	TIPO_PRODUCTO	Id_TipoProducto	NO	int	NULL
50	KVFACTURACION	TIPO_PRODUCTO	Nombre	YES	varchar	30
51	KVFACTURACION	TIPO_PRODUCTO	Estado	YES	tinyint	NULL

Figura 58 Diccionario de Datos: Tabla Tipo_Producto

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
52	KVFACTURACION	TIPO_USUARIO	IdTipoUsuario	NO	int	NULL
53	KVFACTURACION	TIPO_USUARIO	Nombre	NO	char	15

Figura 59 Diccionario de Datos: Tabla Tipo_Usuario

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
54	KVFACTURACION	VENTA	Id_Venta	NO	int	NULL
55	KVFACTURACION	VENTA	IdUsuario	NO	int	NULL
56	KVFACTURACION	VENTA	IdTipoDocumento	NO	int	NULL
57	KVFACTURACION	VENTA	Nombres	NO	varchar	50
58	KVFACTURACION	VENTA	DniRuc	NO	varchar	20
59	KVFACTURACION	VENTA	NumeroVenta	NO	varchar	10
60	KVFACTURACION	VENTA	MontoTotal	NO	decimal	NULL
61	KVFACTURACION	VENTA	Fecha	NO	date	NULL

Figura 60 Diccionario de Datos: Tabla Venta

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
1	KVFACTURACION	USUARIO	IdUsuario	NO	int	NULL
2	KVFACTURACION	USUARIO	Nombre_Usuario	NO	varchar	20
3	KVFACTURACION	USUARIO	Contraseña_Usuario	NO	varchar	20
4	KVFACTURACION	USUARIO	IdTipoUsuario	NO	int	NULL
5	KVFACTURACION	USUARIO	Nombre_Apellidos	NO	varchar	50
6	KVFACTURACION	USUARIO	Dni	NO	char	8
7	KVFACTURACION	USUARIO	Telefono	YES	char	9
8	KVFACTURACION	USUARIO	Direccion	YES	varchar	50
9	KVFACTURACION	USUARIO	Estado	NO	varchar	10

Figura 61 Diccionario de Datos: Tabla Usuario

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
1	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdDetalleOrdenCompra	NO	int	NULL
2	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdOrdenCompra	NO	int	NULL
3	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdProducto	NO	int	NULL
4	KVFACTURACION	DETALLE_ORDEN_COMPRA	Cantidad	NO	int	NULL
5	KVFACTURACION	DETALLE_ORDEN_COMPRA	Monto	NO	decimal	NULL

Figura 62 Diccionario de Datos: Tabla Detalle_Orden_Compra

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
6	KVFACTURACION	DETALLE_VENTA	IdDetalleVenta	NO	int	NULL
7	KVFACTURACION	DETALLE_VENTA	Id_Venta	NO	int	NULL
8	KVFACTURACION	DETALLE_VENTA	IdProducto	NO	int	NULL
9	KVFACTURACION	DETALLE_VENTA	IdGarantia	NO	int	NULL
10	KVFACTURACION	DETALLE_VENTA	Cantidad	NO	int	NULL
11	KVFACTURACION	DETALLE_VENTA	Total	NO	decimal	NULL
12	KVFACTURACION	DETALLE_VENTA	Monto	NO	decimal	NULL

Figura 63 Diccionario de Datos: Tabla Detalle_Venta

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME
1	KVFACTURACION	sysdiagrams	diagram_id	PK_sysdiagr_C2B05B61440A0D77
2	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdDetalleOrdenCompra	PK_DETALLE_ORDEN_COMPRA
3	KVFACTURACION	DETALLE_VENTA	IdDetalleVenta	PK_DETALLE_VENTA
4	KVFACTURACION	GARANTIA	Id_Garantia	PK_GARANTIA
5	KVFACTURACION	MARCA_PRODUCTO	Id_Marca	PK_MARCA_PRODUCTO
6	KVFACTURACION	ORDEN_COMPRA	IdOrdenCompra	PK_ORDEN_COMPRA
7	KVFACTURACION	PRODUCTO	IdProducto	PK_PRODUCTO
8	KVFACTURACION	PROVEEDOR	IdProveedor	PK_PROVEEDOR
9	KVFACTURACION	TIPO_DOCUMENTO	IdTipoDocumento	PK_TIPO_DOCUMENTO
10	KVFACTURACION	TIPO_PRODUCTO	Id_TipoProducto	PK_TIPO_PRODUCTO
11	KVFACTURACION	TIPO_USUARIO	IdTipoUsuario	PK_TIPO_USUARIO
12	KVFACTURACION	USUARIO	IdUsuario	PK_USUARIO
13	KVFACTURACION	VENTA	Id_Venta	PK_VENTA

Figura 64 Diccionario de Datos: Primary Key

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME
1	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdOrdenCompra	FK_DETALLE_ORDEN_COMPRA_ORDEN_COMPRA
2	KVFACTURACION	DETALLE_ORDEN_COMPRA	IdProducto	FK_DETALLE_ORDEN_COMPRA_PRODUCTO
3	KVFACTURACION	DETALLE_VENTA	IdGarantia	FK_DETALLE_VENTA_GARANTIA
4	KVFACTURACION	DETALLE_VENTA	IdProducto	FK_DETALLE_VENTA_PRODUCTO
5	KVFACTURACION	DETALLE_VENTA	Id_Venta	FK_DETALLE_VENTA_VENTA
6	KVFACTURACION	ORDEN_COMPRA	IdProveedor	FK_ORDEN_COMPRA_PROVEEDOR
7	KVFACTURACION	ORDEN_COMPRA	IdUsuario	FK_ORDEN_COMPRA_USUARIO
8	KVFACTURACION	PRODUCTO	Id_Marca	FK_PRODUCTO_MARCA_PRODUCTO
9	KVFACTURACION	PRODUCTO	Id_TipoProducto	FK_PRODUCTO_TIPO_PRODUCTO
10	KVFACTURACION	USUARIO	IdTipoUsuario	FK_USUARIO_TIPO_USUARIO
11	KVFACTURACION	VENTA	IdTipoDocumento	FK_VENTA_TIPO_DOCUMENTO
12	KVFACTURACION	VENTA	IdUsuario	FK_VENTA_USUARIO

Figura 65 Diccionario de Datos: Foreign Key



1.2 Programación

Clase Usuario

```
Imports System.Data.SqlClient
Imports CAPA_ENTIDAD
Public Class CpDCIs_USUARIO
    Public dapprov As New SqlDataAdapter
    Public Sub CapturarNombreTipo(ByVal obj As
CAPA_ENTIDAD.CpECIs_USUARIO)
        Dim cn As New SqlConnection(cadena)
        Dim dausuario As New SqlDataAdapter
        cn.Open()
        Try
            dausuario.SelectCommand = cn.CreateCommand
            dausuario.SelectCommand.CommandText = "PA_Capturar_TipoUsuario"
            dausuario.SelectCommand.CommandType =
CommandType.StoredProcedure
            dausuario.SelectCommand.Parameters.AddWithValue("@nom", obj.usu)
            dausuario.SelectCommand.ExecuteNonQuery()
            obj.NomTipo = dausuario.SelectCommand.ExecuteScalar()
        Catch ex As Exception
            MsgBox(ex.Message, MsgBoxStyle.Question)
            Throw New Exception(ex.Message)
        Finally
            If cn.State = ConnectionState.Open Then
                cn.Dispose()
                cn.Close()
            End If
        End Try
    End Sub
    Public Sub CapturarCodigoUsuario(ByVal obj As
CAPA_ENTIDAD.CpECIs_USUARIO)
```



```
Dim cn As New SqlConnection(cadena)
Dim dausuario As New SqlDataAdapter
cn.Open()
Try
    dausuario.SelectCommand = cn.CreateCommand
    dausuario.SelectCommand.CommandText = "PA_CapturarCodigoUsuario"
    dausuario.SelectCommand.CommandType =
CommandType.StoredProcedure
    dausuario.SelectCommand.Parameters.AddWithValue("@usu", obj.usu)
    dausuario.SelectCommand.ExecuteNonQuery()
    obj.Idusuario = dausuario.SelectCommand.ExecuteScalar()
Catch ex As Exception
    MsgBox(ex.Message, MsgBoxStyle.Question)
    Throw New Exception(ex.Message)
Finally
    If cn.State = ConnectionState.Open Then
        cn.Dispose()
        cn.Close()
    End If
End Try
End Sub
```

```
Public Sub CapturarNombreApellidoUsuario(ByVal obj As
CAPA_ENTIDAD.CpECIs_USUARIO)
```

```
Dim cn As New SqlConnection(cadena)
Dim dausuario As New SqlDataAdapter
cn.Open()
Try
    dausuario.SelectCommand = cn.CreateCommand
    dausuario.SelectCommand.CommandText =
"PA_CapturarNombreApellidoUsuario"
```



```
dausuario.SelectCommand.CommandType =  
CommandType.StoredProcedure  
  
dausuario.SelectCommand.Parameters.AddWithValue("@usu", obj.usu)  
dausuario.SelectCommand.ExecuteNonQuery()  
obj.NomApe = dausuario.SelectCommand.ExecuteScalar()  
Catch ex As Exception  
    MsgBox(ex.Message, MsgBoxStyle.Question)  
    Throw New Exception(ex.Message)  
Finally  
    If cn.State = ConnectionState.Open Then  
        cn.Dispose()  
        cn.Close()  
    End If  
End Try  
End Sub  
  
Public Function logeo(ByVal n As String, ByVal cl As String, ByVal obj As  
CAPA_ENTIDAD.CpEcls_USUARIO) As Boolean  
    ' Dim cn As New SqlConnection("Data Source=ASUSI7\SQLEXPRESS;Initial  
Catalog=KVFACTURACION;Integrated Security=True")  
    Dim cn As New SqlConnection("Data Source=ASUSI7\SQLEXPRESS;Initial  
Catalog=KVFACTURACION;Integrated Security=True")  
    Dim comando As New SqlCommand  
    Dim parnombre As New SqlParameter  
    Dim parclave As New SqlParameter  
    Dim parvalidar As New SqlParameter  
    n = obj.usu  
    cl = obj.cla  
    Try  
        cn.Open()  
        comando = cn.CreateCommand  
        comando.CommandText = "PA_LOGIN"  
        comando.CommandType = CommandType.StoredProcedure  
        comando.Connection = cn
```



```
parnombre.ParameterName = "@nombre"
```

```
parclave.ParameterName = "@clave"
```

```
parvalidar.ParameterName = "@validar"
```

```
parnombre.Value = n
```

```
parclave.Value = cl
```

```
parvalidar.Direction = ParameterDirection.Output
```

```
parvalidar.SqlDbType = SqlDbType.Bit
```

```
parvalidar.Size = 1
```

```
comando.Parameters.Add(parnombre)
```

```
comando.Parameters.Add(parclave)
```

```
comando.Parameters.Add(parvalidar)
```

```
comando.ExecuteNonQuery()
```

```
Catch ex As Exception
```

```
Throw ex
```

```
Finally
```

```
If cn.State = ConnectionState.Open Then
```

```
cn.Close()
```

```
cn.Dispose()
```

```
End If
```

```
End Try
```

```
Return parvalidar.Value
```

```
End Function
```

```
Public Function MOSTRAR_USUARIO(ByVal obj As  
CAPA_ENTIDAD.CpECLs_USUARIO)
```

```
Dim cn As New SqlConnection(cadena)
```



Dim dtprov As New Data.DataTable

Try

cn.Open()

daprov.SelectCommand = cn.CreateCommand

daprov.SelectCommand.CommandText = "PA_ListarUsuario"

daprov.SelectCommand.CommandType = CommandType.StoredProcedure

daprov.SelectCommand.ExecuteNonQuery()

daprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

Public Function MOSTRAR_TIPO_USUARIO(ByVal obj As
CAPA_ENTIDAD.CpECLs_USUARIO)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try

cn.Open()

daprov.SelectCommand = cn.CreateCommand

daprov.SelectCommand.CommandText = "PA_ListarTipoUsuario"

daprov.SelectCommand.CommandType = CommandType.StoredProcedure

daprov.SelectCommand.ExecuteNonQuery()

daprov.Fill(dtprov)

Return dtprov

Catch ex As Exception



MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

Public Sub REGISTRAR_USUARIO(ByVal obj As
CAPA_ENTIDAD.CpECIs_USUARIO)

Dim cn As New SqlConnection(cadena)

cn.Open()

Try

dapprov.InsertCommand = cn.CreateCommand

dapprov.InsertCommand.CommandText = "PA_RegistraUsuario"

dapprov.InsertCommand.CommandType = CommandType.StoredProcedure

dapprov.InsertCommand.Parameters.AddWithValue("@nom", obj.usu)

dapprov.InsertCommand.Parameters.AddWithValue("@PASS", obj.cla)

dapprov.InsertCommand.Parameters.AddWithValue("@idtipo", obj.Idtipo)

dapprov.InsertCommand.Parameters.AddWithValue("@NomApe",
obj.NomApe)

dapprov.InsertCommand.Parameters.AddWithValue("@dni", obj.dni)

dapprov.InsertCommand.Parameters.AddWithValue("@telefono", obj.tele)

dapprov.InsertCommand.Parameters.AddWithValue("@direccion",
obj.direccion)

dapprov.InsertCommand.Parameters.AddWithValue("@estado", obj.estado)

dapprov.InsertCommand.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then



cn.Close()

cn.Dispose()

End If

End Try

End Sub

Public Sub MODIFICAR_USUARIO(ByVal obj As
CAPA_ENTIDAD.CpECIs_USUARIO)

Dim cn As New SqlConnection(cadena)

Try

cn.Open()

dapprov.UpdateCommand = cn.CreateCommand

dapprov.UpdateCommand.CommandText = "PA_EditarUsuario"

dapprov.UpdateCommand.CommandType =
CommandType.StoredProcedure

dapprov.UpdateCommand.Parameters.AddWithValue("@id", obj.Idusuario)

dapprov.UpdateCommand.Parameters.AddWithValue("@nom", obj.usu)

dapprov.UpdateCommand.Parameters.AddWithValue("@PASS", obj.cla)

dapprov.UpdateCommand.Parameters.AddWithValue("@idtipo", obj.Idtipo)

dapprov.UpdateCommand.Parameters.AddWithValue("@NomApe",
obj.NomApe)

dapprov.UpdateCommand.Parameters.AddWithValue("@dni", obj.dni)

dapprov.UpdateCommand.Parameters.AddWithValue("@telefono", obj.tele)

dapprov.UpdateCommand.Parameters.AddWithValue("@direccion",
obj.direccion)

dapprov.UpdateCommand.Parameters.AddWithValue("@estado", obj.estado)

dapprov.UpdateCommand.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()



cn.Close()

End If

End Try

End Sub

End Class

Clase Venta

Imports CAPA_ENTIDAD

Imports System.Data.SqlClient

Public Class CpD_Cls_VENTA

Public dapprov As New SqlDataAdapter

Public Sub REGISTRAR_VENTA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_VENTA)

Dim cn As New SqlConnection(cadena)

cn.Open()

Try

dapprov.InsertCommand = cn.CreateCommand

dapprov.InsertCommand.CommandText = "PA_RegistrarVenta"

dapprov.InsertCommand.CommandType = CommandType.StoredProcedure

dapprov.InsertCommand.Parameters.AddWithValue("@idusuario",
obj.idUsuario)

dapprov.InsertCommand.Parameters.AddWithValue("@idtipo", obj.idtipo)

dapprov.InsertCommand.Parameters.AddWithValue("@nombres",
obj.nombres)

dapprov.InsertCommand.Parameters.AddWithValue("@dniruc", obj.dniruc)

dapprov.InsertCommand.Parameters.AddWithValue("@NumeroVenta",
obj.numero)

dapprov.InsertCommand.Parameters.AddWithValue("@MontoTotal",
obj.Montototal)

dapprov.InsertCommand.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then



```
cn.Close()  
cn.Dispose()  
End If  
End Try  
End Sub
```

```
Public Sub REGISTRAR_DETALLE_VENTA(ByVal obj As  
CAPA_ENTIDAD.CpE_Cls_VENTA)
```

```
Dim cn As New SqlConnection(cadena)  
cn.Open()  
Try  
dapprov.InsertCommand = cn.CreateCommand  
dapprov.InsertCommand.CommandText = "PA_RegistrarDetalleVenta"  
dapprov.InsertCommand.CommandType = CommandType.StoredProcedure  
dapprov.InsertCommand.Parameters.AddWithValue("@idventa",  
obj.idVenta)  
dapprov.InsertCommand.Parameters.AddWithValue("@idpro",  
obj.idproducto)  
dapprov.InsertCommand.Parameters.AddWithValue("@idGara",  
obj.idgarantia)  
dapprov.InsertCommand.Parameters.AddWithValue("@cantidad",  
obj.cantidad)  
dapprov.InsertCommand.Parameters.AddWithValue("@total", obj.total)  
dapprov.InsertCommand.Parameters.AddWithValue("@monto", obj.Monto)  
dapprov.InsertCommand.ExecuteNonQuery()  
Catch ex As Exception  
MsgBox(ex.Message)  
Throw New Exception(ex.Message)  
Finally  
If cn.State = ConnectionState.Open Then  
cn.Close()  
cn.Dispose()  
End If  
End Try
```



End Sub

```
Public Sub CAPTURA_CODIGO_VENTA(ByVal obj As  
CAPA_ENTIDAD.CpE_Cls_VENTA)
```

```
Dim cn As New SqlConnection(cadena)
```

```
Try
```

```
cn.Open()
```

```
daprov.SelectCommand = cn.CreateCommand
```

```
daprov.SelectCommand.CommandText = "PA_CapturarCodigoVenta"
```

```
daprov.SelectCommand.CommandType = CommandType.StoredProcedure
```

```
daprov.SelectCommand.ExecuteNonQuery()
```

```
obj.idVenta = daprov.SelectCommand.ExecuteScalar()
```

```
Catch ex As Exception
```

```
MsgBox(ex.Message, MsgBoxStyle.Question)
```

```
' Throw New Exception(ex.Message)
```

```
Finally
```

```
If cn.State = ConnectionState.Open Then
```

```
cn.Dispose()
```

```
cn.Close()
```

```
End If
```

```
End Try
```

```
End Sub
```

```
Public Function BUSCAR_VENTA(ByVal obj As  
CAPA_ENTIDAD.CpE_Cls_VENTA)
```

```
Dim cn As New SqlConnection(cadena)
```

```
Dim dtprov As New Data.DataTable
```

```
Try
```

```
cn.Open()
```

```
daprov.SelectCommand = cn.CreateCommand
```

```
daprov.SelectCommand.CommandText = "PA_ListarVentas"
```

```
daprov.SelectCommand.CommandType = CommandType.StoredProcedure
```

```
' daprov.SelectCommand.Parameters.AddWithValue("@fecha", obj.fecha)
```

```
daprov.SelectCommand.ExecuteNonQuery()
```



```
dapprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

    MsgBox(ex.Message)

    Throw New Exception(ex.Message)

Finally

    If cn.State = ConnectionState.Open Then

        cn.Dispose()

        cn.Close()

    End If

End Try

End Function

Public Sub CAPTURA_NUMERO_VENTA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_VENTA)

    Dim cn As New SqlConnection(cadena)

    Try

        cn.Open()

        dapprov.SelectCommand = cn.CreateCommand

        dapprov.SelectCommand.CommandText = "PA_GenerarNumeroVenta"

        dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

        dapprov.SelectCommand.ExecuteNonQuery()

        obj.numero = dapprov.SelectCommand.ExecuteScalar()

    Catch ex As Exception

        MsgBox(ex.Message, MsgBoxStyle.Question)

        ' Throw New Exception(ex.Message)

    Finally

        If cn.State = ConnectionState.Open Then

            cn.Dispose()

            cn.Close()

        End If

    End Try

End Sub
```



Public Sub CapturarStockProducto(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_VENTA)

Dim cn As New SqlConnection(cadena)

Dim dausuario As New SqlDataAdapter

cn.Open()

Try

dausuario.SelectCommand = cn.CreateCommand

dausuario.SelectCommand.CommandText = "PA_CapturarStockProducto"

dausuario.SelectCommand.CommandType =
CommandType.StoredProcedure

dausuario.SelectCommand.Parameters.AddWithValue("@idproducto",
obj.idproducto)

dausuario.SelectCommand.ExecuteNonQuery()

obj.stock = dausuario.SelectCommand.ExecuteScalar()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Question)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Sub

Public Sub MODIFICAR_STOCK_PRODUCTO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_VENTA)

Dim cn As New SqlConnection(cadena)

Try

cn.Open()

dapprov.UpdateCommand = cn.CreateCommand

dapprov.UpdateCommand.CommandText = "PA_EditaStockProducto"

dapprov.UpdateCommand.CommandType =
CommandType.StoredProcedure



```
dapprov.UpdateCommand.Parameters.AddWithValue("@id", obj.idproducto)
dapprov.UpdateCommand.Parameters.AddWithValue("@stocktotal",
obj.stock)
dapprov.UpdateCommand.ExecuteNonQuery()
Catch ex As Exception
MsgBox(ex.Message)
Throw New Exception(ex.Message)
Finally
If cn.State = ConnectionState.Open Then
cn.Dispose()
cn.Close()
End If
End Try
End Sub
End Class
```

Clase Productos

```
Imports CAPA_ENTIDAD
Imports System.Data.SqlClient
Public Class CpD_Cls_PRODUCTO
Public dapprov As New SqlDataAdapter

Public Function MOSTRAR_PRODUCUTO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)
Dim cn As New SqlConnection(cadena)
Dim dtprov As New Data.DataTable
Try
cn.Open()
dapprov.SelectCommand = cn.CreateCommand
dapprov.SelectCommand.CommandText = "PA_ListarProducto"
```



```
dapprov.SelectCommand.CommandType = CommandType.StoredProcedure
dapprov.SelectCommand.ExecuteNonQuery()
dapprov.Fill(dtprov)
Return dtprov
Catch ex As Exception
    MsgBox(ex.Message)
    Throw New Exception(ex.Message)
Finally
    If cn.State = ConnectionState.Open Then
        cn.Dispose()
        cn.Close()
    End If
End Try
End Function

Public Function MOSTRAR_PRODUCTO_ESTADO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)
    Dim cn As New SqlConnection(cadena)
    Dim dtprov As New Data.DataTable
    Try
        cn.Open()
        dapprov.SelectCommand = cn.CreateCommand
        dapprov.SelectCommand.CommandText = "PA_ListarProductoEstado"
        dapprov.SelectCommand.CommandType = CommandType.StoredProcedure
        dapprov.SelectCommand.ExecuteNonQuery()
        dapprov.Fill(dtprov)
        Return dtprov
    Catch ex As Exception
        MsgBox(ex.Message)
        Throw New Exception(ex.Message)
    Finally
        If cn.State = ConnectionState.Open Then
            cn.Dispose()
```



cn.Close()

End If

End Try

End Function

Public Sub REGISTRAR_PRODUCTO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)

Dim cn As New SqlConnection(cadena)

cn.Open()

Try

dapprov.InsertCommand = cn.CreateCommand

dapprov.InsertCommand.CommandText = "PA_RegistrarProducto"

dapprov.InsertCommand.CommandType = CommandType.StoredProcedure

dapprov.InsertCommand.Parameters.AddWithValue("@nom", obj.Nombre)

dapprov.InsertCommand.Parameters.AddWithValue("@des",
obj.descripcion)

dapprov.InsertCommand.Parameters.AddWithValue("@precio_compra",
obj.precio_compra)

dapprov.InsertCommand.Parameters.AddWithValue("@precio_venta",
obj.precio_venta)

dapprov.InsertCommand.Parameters.AddWithValue("@stock", obj.stock)

dapprov.InsertCommand.Parameters.AddWithValue("@esta", obj.Estado)

dapprov.InsertCommand.Parameters.AddWithValue("@idtipo", obj.idtipo)

dapprov.InsertCommand.Parameters.AddWithValue("@idmarca",
obj.idmarca)

dapprov.InsertCommand.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Close()

cn.Dispose()

End If



End Try

End Sub

Public Sub MODIFICAR_PRODUCTO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)

Dim cn As New SqlConnection(cadena)

Try

cn.Open()

dapprov.UpdateCommand = cn.CreateCommand

dapprov.UpdateCommand.CommandText = "PA_EditarProducto"

dapprov.UpdateCommand.CommandType =
CommandType.StoredProcedure

dapprov.UpdateCommand.Parameters.AddWithValue("@id", obj.idproducto)

dapprov.UpdateCommand.Parameters.AddWithValue("@nom", obj.Nombre)

dapprov.UpdateCommand.Parameters.AddWithValue("@des",
obj.descripcion)

dapprov.UpdateCommand.Parameters.AddWithValue("@precio_compra",
obj.precio_compra)

dapprov.UpdateCommand.Parameters.AddWithValue("@precio_venta",
obj.precio_venta)

dapprov.UpdateCommand.Parameters.AddWithValue("@stock", obj.stock)

dapprov.UpdateCommand.Parameters.AddWithValue("@esta", obj.Estado)

dapprov.UpdateCommand.Parameters.AddWithValue("@idtipo", obj.idtipo)

dapprov.UpdateCommand.Parameters.AddWithValue("@idmarca",
obj.idmarca)

dapprov.UpdateCommand.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try



End Sub

Public Function BUSCAR_PRODUCTO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try

cn.Open()

daprov.SelectCommand = cn.CreateCommand

daprov.SelectCommand.CommandText = "PA_BuscarProducto"

daprov.SelectCommand.CommandType = CommandType.StoredProcedure

daprov.SelectCommand.Parameters.AddWithValue("@nom", obj.Nombre)

daprov.SelectCommand.Parameters.AddWithValue("@tipo", obj.Nombre)

daprov.SelectCommand.Parameters.AddWithValue("@opcion", obj.opcion)

daprov.SelectCommand.ExecuteNonQuery()

daprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

Public Function BUSCAR_PRODUCTOS_ACTIVOS(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try

cn.Open()



```
dapprov.SelectCommand = cn.CreateCommand  
dapprov.SelectCommand.CommandText = "PA_BuscarProductoActivo"  
dapprov.SelectCommand.CommandType = CommandType.StoredProcedure  
dapprov.SelectCommand.Parameters.AddWithValue("@nom", obj.Nombre)  
dapprov.SelectCommand.Parameters.AddWithValue("@tipo", obj.Nombre)  
dapprov.SelectCommand.Parameters.AddWithValue("@opcion", obj.opcion)  
dapprov.SelectCommand.ExecuteNonQuery()  
dapprov.Fill(dtprov)  
Return dtprov  
Catch ex As Exception  
    MsgBox(ex.Message)  
    Throw New Exception(ex.Message)  
Finally  
    If cn.State = ConnectionState.Open Then  
        cn.Dispose()  
        cn.Close()  
    End If  
End Try  
End Function
```

```
Public Sub MODIFICAR_STOCK_PRODUCTO(ByVal obj As  
CAPA_ENTIDAD.CpE_Cls_PRODUCTO)  
    Dim cn As New SqlConnection(cadena)  
    Try  
        cn.Open()  
        dapprov.UpdateCommand = cn.CreateCommand  
        dapprov.UpdateCommand.CommandText = "PA_EditaStockProducto"  
        dapprov.UpdateCommand.CommandType =  
CommandType.StoredProcedure  
        dapprov.UpdateCommand.Parameters.AddWithValue("@id", obj.idproducto)  
        dapprov.UpdateCommand.Parameters.AddWithValue("@stocktotal",  
obj.stock)  
        dapprov.UpdateCommand.ExecuteNonQuery()
```



Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Sub

End Class

Clase Orden de Compra

Imports CAPA_ENTIDAD

Imports System.Data.SqlClient

Public Class CpD_Cls_ORDEN_COMPRA

Public dapprov As New SqlDataAdapter

Public Sub REGISTRAR_ORDEN_COMPRA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_ORDEN_COMPRA)

Dim cn As New SqlConnection(cadena)

cn.Open()

Try

dapprov.InsertCommand = cn.CreateCommand

dapprov.InsertCommand.CommandText = "PA_RegistrarOrdenCompra"

dapprov.InsertCommand.CommandType = CommandType.StoredProcedure

dapprov.InsertCommand.Parameters.AddWithValue("@idprove",
obj.idproveedor)

dapprov.InsertCommand.Parameters.AddWithValue("@idusuario",
obj.idUsuario)

dapprov.InsertCommand.Parameters.AddWithValue("@numero",
obj.numero)

dapprov.InsertCommand.Parameters.AddWithValue("@MontoTotal",
obj.Montototal)



```
dapprov.InsertCommand.Parameters.AddWithValue("@estado", obj.Estado)
dapprov.InsertCommand.Parameters.AddWithValue("@des",
obj.descripcion)
dapprov.InsertCommand.ExecuteNonQuery()
```

Catch ex As Exception

```
MsgBox(ex.Message)
```

```
'Throw New Exception(ex.Message)
```

Finally

```
If cn.State = ConnectionState.Open Then
```

```
cn.Close()
```

```
cn.Dispose()
```

```
End If
```

End Try

End Sub

```
Public Sub REGISTRAR_DETALLE_ORDEN_COMPRA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_ORDEN_COMPRA)
```

```
Dim cn As New SqlConnection(cadena)
```

```
cn.Open()
```

Try

```
dapprov.InsertCommand = cn.CreateCommand
```

```
dapprov.InsertCommand.CommandText =
"PA_RegistrarDetalleOrdenCompra"
```

```
dapprov.InsertCommand.CommandType = CommandType.StoredProcedure
```

```
dapprov.InsertCommand.Parameters.AddWithValue("@idorden",
obj.idOrdenCompra)
```

```
dapprov.InsertCommand.Parameters.AddWithValue("@idpro",
obj.idproducto)
```

```
dapprov.InsertCommand.Parameters.AddWithValue("@cantidad",
obj.cantidad)
```

```
dapprov.InsertCommand.Parameters.AddWithValue("@monto", obj.Monto)
```

```
dapprov.InsertCommand.ExecuteNonQuery()
```

Catch ex As Exception



MsgBox(ex.Message)

'Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Close()

cn.Dispose()

End If

End Try

End Sub

Public Sub CAPTURA_CODIGO_ORDEN_COMPRA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_ORDEN_COMPRA)

Dim cn As New SqlConnection(cadena)

Try

cn.Open()

dapprov.SelectCommand = cn.CreateCommand

dapprov.SelectCommand.CommandText =

"PA_CapturarCodigoOrdenCompra"

dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

dapprov.SelectCommand.ExecuteNonQuery()

obj.idOrdenCompra = dapprov.SelectCommand.ExecuteScalar()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Question)

' Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Sub



Public Sub CAPTURA_NUMERO_ORDENCOMPRA(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_ORDEN_COMPRA)

Dim cn As New SqlConnection(cadena)

Try

cn.Open()

dapprov.SelectCommand = cn.CreateCommand

dapprov.SelectCommand.CommandText =
"PA_GenerarNumeroOrdenCompra"

dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

dapprov.SelectCommand.ExecuteNonQuery()

obj.numero = dapprov.SelectCommand.ExecuteScalar()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Question)

' Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Sub

End Class

Class Proveedor

Imports CAPA_ENTIDAD

Imports System.Data.SqlClient

Public Class CpD_Cls_PROVEEDOR

Public dapprov As New SqlDataAdapter

Public Function MOSTRAR_PROVEEDORES(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try



cn.Open()

dapprov.SelectCommand = cn.CreateCommand

dapprov.SelectCommand.CommandText = "PA_ListarProveedor"

dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

dapprov.SelectCommand.ExecuteNonQuery()

dapprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

Public Sub REGISTRAR_PROVEEDOR(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)

Dim cn As New SqlConnection(cadena)

cn.Open()

Try

dapprov.InsertCommand = cn.CreateCommand

dapprov.InsertCommand.CommandText = "PA_RegistrarProveedor"

dapprov.InsertCommand.CommandType = CommandType.StoredProcedure

dapprov.InsertCommand.Parameters.AddWithValue("@nomcompa",
obj.NombreCompañia)

dapprov.InsertCommand.Parameters.AddWithValue("@nomcomtac",
obj.NombreContacto)

dapprov.InsertCommand.Parameters.AddWithValue("@ruc", obj.ruc)

dapprov.InsertCommand.Parameters.AddWithValue("@bancario",
obj.Bancario)



```
dapprov.InsertCommand.Parameters.AddWithValue("@tele", obj.telefono)
dapprov.InsertCommand.Parameters.AddWithValue("@dire", obj.direccion)
dapprov.InsertCommand.Parameters.AddWithValue("@email", obj.Email)
dapprov.InsertCommand.Parameters.AddWithValue("@esta", obj.Estado)
dapprov.InsertCommand.ExecuteNonQuery()
```

Catch ex As Exception

```
MsgBox(ex.Message)
Throw New Exception(ex.Message)
```

Finally

```
If cn.State = ConnectionState.Open Then
    cn.Close()
    cn.Dispose()
```

End If

End Try

End Sub

```
Public Sub MODIFICAR_PROVEEDOR(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)
```

```
Dim cn As New SqlConnection(cadena)
```

```
Try
```

```
cn.Open()
dapprov.UpdateCommand = cn.CreateCommand
dapprov.UpdateCommand.CommandText = "PA_EditarProveedor"
dapprov.UpdateCommand.CommandType =
CommandType.StoredProcedure
```

```
dapprov.UpdateCommand.Parameters.AddWithValue("@id",
obj.idproveedor)
```

```
dapprov.UpdateCommand.Parameters.AddWithValue("@nomcompa",
obj.NombreCompañia)
```

```
dapprov.UpdateCommand.Parameters.AddWithValue("@nomcontac",
obj.NombreContacto)
```

```
dapprov.UpdateCommand.Parameters.AddWithValue("@ruc", obj.ruc)
```



```
dapprov.UpdateCommand.Parameters.AddWithValue("@bancario",  
obj.Bancario)  
  
dapprov.UpdateCommand.Parameters.AddWithValue("@tele", obj.telefono)  
dapprov.UpdateCommand.Parameters.AddWithValue("@dire",  
obj.direccion)  
  
dapprov.UpdateCommand.Parameters.AddWithValue("@email", obj.Email)  
dapprov.UpdateCommand.Parameters.AddWithValue("@esta", obj.Estado)  
dapprov.UpdateCommand.ExecuteNonQuery()  
  
Catch ex As Exception  
MsgBox(ex.Message)  
Throw New Exception(ex.Message)  
  
Finally  
If cn.State = ConnectionState.Open Then  
cn.Dispose()  
cn.Close()  
End If  
End Try  
End Sub  
  
Public Function BUSCAR_PROVEEDOR(ByVal obj As  
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)  
Dim cn As New SqlConnection(cadena)  
Dim dtprov As New Data.DataTable  
Try  
cn.Open()  
dapprov.SelectCommand = cn.CreateCommand  
dapprov.SelectCommand.CommandText = "PA_BuscarProveedor"  
dapprov.SelectCommand.CommandType = CommandType.StoredProcedure  
dapprov.SelectCommand.Parameters.AddWithValue("@ruc", obj.ruc)  
dapprov.SelectCommand.Parameters.AddWithValue("@nomcompa",  
obj.NombreCompañia)  
dapprov.SelectCommand.Parameters.AddWithValue("@opcion", obj.opcion)  
dapprov.SelectCommand.ExecuteNonQuery()  
dapprov.Fill(dtprov)
```



Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

Public Function MOSTRAR_PROVEEDORES_ESTADO(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try

cn.Open()

dapprov.SelectCommand = cn.CreateCommand

dapprov.SelectCommand.CommandText = "PA_ListarProveedorEstado"

dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

dapprov.SelectCommand.ExecuteNonQuery()

dapprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try



End Function

Public Function BUSCAR_PROVEEDOR_ACTIVOS(ByVal obj As
CAPA_ENTIDAD.CpE_Cls_PROVEEDOR)

Dim cn As New SqlConnection(cadena)

Dim dtprov As New Data.DataTable

Try

cn.Open()

dapprov.SelectCommand = cn.CreateCommand

dapprov.SelectCommand.CommandText = "PA_BuscarProveedorActivo"

dapprov.SelectCommand.CommandType = CommandType.StoredProcedure

dapprov.SelectCommand.Parameters.AddWithValue("@ruc", obj.ruc)

dapprov.SelectCommand.Parameters.AddWithValue("@nomcompa",
obj.NombreCompañia)

dapprov.SelectCommand.Parameters.AddWithValue("@opcion", obj.opcion)

dapprov.SelectCommand.ExecuteNonQuery()

dapprov.Fill(dtprov)

Return dtprov

Catch ex As Exception

MsgBox(ex.Message)

Throw New Exception(ex.Message)

Finally

If cn.State = ConnectionState.Open Then

cn.Dispose()

cn.Close()

End If

End Try

End Function

End Class



1.3 Script Base de Datos

```
USE [master]
```

```
GO
```

```
/****** Object: Database [KVFACTURACION] Script Date: 11/09/2017  
23:07:23 *****/
```

```
CREATE DATABASE [KVFACTURACION]
```

```
CONTAINMENT = NONE
```

```
ON PRIMARY
```

```
( NAME = N'KVFACTURACION', FILENAME = N'c:\Program Files\Microsoft  
SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\KVFACTURACION.mdf' ,  
SIZE = 4160KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
```

```
LOG ON
```

```
( NAME = N'KVFACTURACION_log', FILENAME = N'c:\Program  
Files\Microsoft SQL  
Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\KVFACTURACION_log.ldf' ,  
SIZE = 1040KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET COMPATIBILITY_LEVEL =  
110
```

```
GO
```

```
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
```

```
begin
```

```
EXEC [KVFACTURACION].[dbo].[sp_fulltext_database] @action = 'enable'
```

```
end
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET ANSI_NULL_DEFAULT OFF
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET ANSI_NULLS OFF
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET ANSI_PADDING OFF
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET ANSI_WARNINGS OFF
```

```
GO
```

```
ALTER DATABASE [KVFACTURACION] SET ARITHABORT OFF
```



GO

ALTER DATABASE [KVFACTURACION] SET AUTO_CLOSE ON

GO

ALTER DATABASE [KVFACTURACION] SET AUTO_CREATE_STATISTICS
ON

GO

ALTER DATABASE [KVFACTURACION] SET AUTO_SHRINK OFF

GO

ALTER DATABASE [KVFACTURACION] SET AUTO_UPDATE_STATISTICS
ON

GO

ALTER DATABASE [KVFACTURACION] SET
CURSOR_CLOSE_ON_COMMIT OFF

GO

ALTER DATABASE [KVFACTURACION] SET CURSOR_DEFAULT GLOBAL

GO

ALTER DATABASE [KVFACTURACION] SET
CONCAT_NULL_YIELDS_NULL OFF

GO

ALTER DATABASE [KVFACTURACION] SET NUMERIC_ROUNDABORT
OFF

GO

ALTER DATABASE [KVFACTURACION] SET QUOTED_IDENTIFIER OFF

GO

ALTER DATABASE [KVFACTURACION] SET RECURSIVE_TRIGGERS OFF

GO

ALTER DATABASE [KVFACTURACION] SET ENABLE_BROKER

GO

ALTER DATABASE [KVFACTURACION] SET
AUTO_UPDATE_STATISTICS_ASYNC OFF

GO

ALTER DATABASE [KVFACTURACION] SET
DATE_CORRELATION_OPTIMIZATION OFF

GO

ALTER DATABASE [KVFACTURACION] SET TRUSTWORTHY OFF



GO

```
ALTER DATABASE [KVFACTURACION] SET  
ALLOW_SNAPSHOT_ISOLATION OFF
```

GO

```
ALTER DATABASE [KVFACTURACION] SET PARAMETERIZATION  
SIMPLE
```

GO

```
ALTER DATABASE [KVFACTURACION] SET  
READ_COMMITTED_SNAPSHOT OFF
```

GO

```
ALTER DATABASE [KVFACTURACION] SET HONOR_BROKER_PRIORITY  
OFF
```

GO

```
ALTER DATABASE [KVFACTURACION] SET RECOVERY SIMPLE
```

GO

```
ALTER DATABASE [KVFACTURACION] SET MULTI_USER
```

GO

```
ALTER DATABASE [KVFACTURACION] SET PAGE_VERIFY CHECKSUM
```

GO

```
ALTER DATABASE [KVFACTURACION] SET DB_CHAINING OFF
```

GO

```
ALTER DATABASE [KVFACTURACION] SET FILESTREAM(  
NON_TRANSACTED_ACCESS = OFF )
```

GO

```
ALTER DATABASE [KVFACTURACION] SET TARGET_RECOVERY_TIME =  
0 SECONDS
```

GO

```
USE [KVFACTURACION]
```

GO

```
/****** Object: StoredProcedure [dbo].[PA_Capturar_TipoUsuario] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO



```
create procedure [dbo].[PA_Capturar_TipoUsuario]
```

```
@nom Nchar(20)
```

```
as
```

```
select TIPO_USUARIO .Nombre from USUARIO, TIPO_USUARIO WHERE  
USUARIO .IdTipoUsuario = TIPO_USUARIO .IdTipoUsuario
```

```
AND USUARIO .Nombre_Usuario = @nom
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_CapturarCodigoOrdenCompra]  
Script Date: 18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create PROCEDURE [dbo].[PA_CapturarCodigoOrdenCompra]
```

```
AS
```

```
SELECT MAX(IdOrdenCompra ) FROM ORDEN_COMPRA
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_CapturarCodigoUsuario] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE [dbo].[PA_CapturarCodigoUsuario]
```

```
@usu varchar(20)
```

```
as
```

```
select IdUsuario from USUARIO where Nombre_Usuario = @usu
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_CapturarCodigoVenta] Script Date:  
18/09/2017 23:07:23 *****/
```



SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_CapturarCodigoVenta]

AS

SELECT MAX(Id_Venta) FROM VENTA

GO

/***** Object: StoredProcedure [dbo].[PA_CapturarNombreApellidoUsuario]
Script Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_CapturarNombreApellidoUsuario]

@usu varchar(20)

as

select Nombre_Apellidos from USUARIO where Nombre_Usuario = @usu

GO

/***** Object: StoredProcedure [dbo].[PA_CapturarPrecioGarantia] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_CapturarPrecioGarantia]

@idGarantia int

as

select Precio from GARANTIA where Id_Garantia = @idGarantia



GO

/***** Object: StoredProcedure [dbo].[PA_CapturarStockProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_CapturarStockProducto]

@idproducto varchar(20)

as

select Stock from PRODUCTO where IdProducto = @idproducto

GO

/***** Object: StoredProcedure [dbo].[PA_EditarGarantia] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_EditarGarantia]

@ID int,

@nom varchar(20),

@precio decimal(8,2),

@des varchar(50),

@estado varchar(10)

as

begin

if exists (select *from GARANTIA

where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),

'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @nom and Id_Garantia <> @ID)

raiserror('NO SE PUEDE EDITAR LA GARANTIA! YA EXISTE !',16,1)

else



UPDATE GARANTIA

```
SET Nombre = @nom ,Precio =@precio ,Descripcion =@des,Estado= @estado
WHERE Id_Garantia = @ID
END
```

GO

```
/****** Object: StoredProcedure [dbo].[PA_EditarMarcaProducto] Script Date:
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE PROCEDURE [dbo].[PA_EditarMarcaProducto]
```

```
@ID int,
```

```
@nom varchar(20),
```

```
@estado tinyint
```

```
as
```

```
begin
```

```
if exists ( select *from MARCA_PRODUCTO
```

```
where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),
```

```
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')
```

```
= @nom and Id_Marca <> @ID)
```

```
raiserror('NO SE PUEDE EDITAR LA MARCA DEL PRODUCTO ! YA EXISTE
!',16,1)
```

```
else
```

```
UPDATE MARCA_PRODUCTO
```

```
SET Nombre = @nom ,Estado= @estado
```

```
WHERE Id_Marca = @ID
```

```
END
```

GO



/***** Object: StoredProcedure [dbo].[PA_EditarProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_EditarProducto]

@ID int,

@Nom varchar(40),

@des varchar(max),

@precio_compra decimal(8,2),

@precio_venta decimal(8,2),

@stock int,

@esta varchar(10),

@idtipo int,

@idmarca int

AS

BEGIN

if exists (select * from PRODUCTO where (Nombre = @Nom) AND IdProducto
<> @ID)

raiserror('EXISTE YA OTRO PRODUCTO CON ESOS DATOS',16,1)

else

UPDATE PRODUCTO

SET Nombre= @nom , Descripcion =@des ,Precio_Compra =@precio_compra
,Precio_Venta =@precio_venta,STOCK =@stock,Estado =@esta,Id_TipoProducto
=@idtipo ,Id_Marca = @idmarca

WHERE IdProducto = @ID

END

GO

/***** Object: StoredProcedure [dbo].[PA_EditarProveedor] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO



SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_EditarProveedor]

@ID int,

@nomcompa VARCHAR(30),

@nomcontac varchar(30),

@ruc char(11),

@Bancario varchar(50),

@tele char(9),

@dire Varchar(50),

@email Varchar(50),

@esta varchar(10)

AS

BEGIN

if exists (select *from PROVEEDOR

where (replace(replace(replace(replace(replace(ltrim(upper(NombreCompañia)),
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @nomcompa OR Ruc = @RUC) and IdProveedor <> @ID)

raiserror('EXISTE YA OTRO PORVEEDOR CON ESOS DATOS',16,1)

else

UPDATE PROVEEDOR

SET NombreCompañia = @nomcompa , NombreContacto =@nomcontac ,Ruc
=@ruc,bancario=@ Bancario,Telefono=@tele,Direccion=@dire,Email =@email
,Estado =@esta

WHERE IdProveedor= @ID

END

GO

/***** Object: StoredProcedure [dbo].[PA_EditarTipoProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO



SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_EditarTipoProducto]

@ID int,

@nom varchar(20),

@estado tinyint

as

begin

if exists (select *from TIPO_PRODUCTO

where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),

'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @nom and Id_TipoProducto <> @ID)

raiserror('NO SE PUEDE EDITAR EL TIPO DE PRODUCTO ! YA EXISTE
!',16,1)

else

UPDATE TIPO_PRODUCTO

SET Nombre = @nom ,Estado= @estado

WHERE Id_TipoProducto = @ID

END

GO

/***** Object: StoredProcedure [dbo].[PA_EditarUsuario] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_EditarUsuario]

@ID int,

@nom varchar(20),

@pass varchar(20),

@idtipo int,

@NomApe varchar(50),



```
@dni char(8),
@telefono varchar(9),
@direccion varchar(50),
@estado VARCHAR(10)
as
begin
if exists ( select *from Usuario
where replace(replace(replace(replace(replace(ltrim(upper(Nombre_Usuario)),
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')
= @nom and IdUsuario <> @ID )
raiserror('NO PUEDE EDITAR EL NOMBRE YA EXISTE ',16,1)
else
UPDATE Usuario
SET Nombre_Usuario = @nom , Contraseña_Usuario =@pass ,IdTipoUsuario
=@idtipo,Nombre_ApellidoS =@NomApe ,Dni =@dni,telefono=@telefono
,Direccion = @direccion ,Estado =@estado
WHERE IdUsuario = @ID
END

GO

/***** Object: StoredProcedure [dbo].[PA_EditaStockProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_EditaStockProducto]
@id int,
@stocktotal varchar(20)
as
update Producto
set Stock = @stocktotal
```



where IdProducto =@id

GO

/****** Object: StoredProcedure [dbo].[PA_GenerarNumeroOrdenCompra]
Script Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_GenerarNumeroOrdenCompra]

as

DECLARE @codigo VARCHAR(6)

SET @codigo=(SELECT MAX(IdOrdenCompra) FROM ORDEN_COMPRA)

SET @codigo = 'C' + RIGHT('000'+LTRIM(RIGHT (ISNULL(
@codigo,'000'),6)+1),4)

SELECT @codigo AS NUMERO

GO

/****** Object: StoredProcedure [dbo].[PA_GenerarNumeroVenta] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[PA_GenerarNumeroVenta]

as

DECLARE @codigo VARCHAR(6)

SET @codigo=(SELECT MAX(Id_Venta) FROM VENTA)

SET @codigo = 'V' + RIGHT('000'+LTRIM(RIGHT (ISNULL(
@codigo,'000'),6)+1),4)

SELECT @codigo AS NUMERO



GO

/***** Object: StoredProcedure [dbo].[PA_ListarGarantia] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarGarantia]

AS

BEGIN

SET NOCOUNT ON;

SELECT Id_Garantia ,NOMBRE,PRECIO,DESCRIPCION,ESTADO from
GARANTIA

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarGarantiaActivo] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarGarantiaActivo]

AS

BEGIN

SET NOCOUNT ON;

SELECT Nombre + ' = ' + CONVERT(varchar(10), Precio) AS DATOS
,Id_Garantia from GARANTIA where Estado = 'ACTIVO'

END



GO

/***** Object: StoredProcedure [dbo].[PA_ListarMarcaProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarMarcaProducto]

AS

BEGIN

SET NOCOUNT ON;

SELECT Id_marca AS ID,NOMBRE,ESTADO from MARCA_PRODUCTO

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarMarcaProductoActivo] Script
Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarMarcaProductoActivo]

AS

BEGIN

SET NOCOUNT ON;

SELECT * from MARCA_PRODUCTO where Estado = 1

END

GO

/***** Object: StoredProcedure [dbo].[Pa_ListarOrdenCompra] Script Date:
18/09/2017 23:07:23 *****/



SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Pa_ListarOrdenCompra]

as

SELECT o.IdOrdenCompra ,o.NumeroOrdenComra AS
Num_ORDEN_COMPRA,o.FECHA ,o.MontoTotal as
MONTO_TOTAL,p.IdProveedor ,p.NombreCompañia AS NOMBRE_COMPañIA
,p.RUC ,o.ESTADO

from ORDEN_COMPRA as o , PROVEEDOR as p WHERE o.IdProveedor =
p.IdProveedor

GO

/***** Object: StoredProcedure [dbo].[PA_ListarProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_ListarProducto]

AS

BEGIN

SET NOCOUNT ON;

SELECT P.IdProducto AS ID ,P.NOMBRE ,P.Id_TipoProducto,T.Nombre AS
TIPO,P.Id_Marca ,M.Nombre AS MARCA ,P.PRECIO_COMPRA
,P.PRECIO_VENTA,P.STOCK,P.DESCRIPCION,P.ESTADO from PRODUCTO
AS P , TIPO_PRODUCTO AS T ,MARCA_PRODUCTO AS M

WHERE p.Id_TipoProducto = t.Id_TipoProducto and p.Id_Marca = m.Id_Marca

END

GO



/***** Object: StoredProcedure [dbo].[PA_ListarProductoEstado] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_ListarProductoEstado]

AS

BEGIN

SET NOCOUNT ON;

SELECT P.IdProducto AS ID ,P.NOMBRE ,P.Id_TipoProducto,T.Nombre AS
TIPO,P.Id_Marca ,M.Nombre AS MARCA ,P.PRECIO_COMPRA
,P.PRECIO_VENTA,P.STOCK,P.DESCRIPCION,P.ESTADO from PRODUCTO
AS P , TIPO_PRODUCTO AS T ,MARCA_PRODUCTO AS M

WHERE p.Id_TipoProducto = t.Id_TipoProducto and p.Id_Marca = m.Id_Marca
and p.Estado = 'ACTIVO'

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarProveedor] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarProveedor]

AS

BEGIN

SET NOCOUNT ON;

SELECT IdProveedor AS ID,NombreCompañía AS
NOMBRE_COMPañÍA,NombreContacto AS NOMBRE_CONTACTO
,RUC,BANCARIO,TELEFONO,DIRECCION,EMAIL,ESTADO from
PROVEEDOR



END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarProveedorEstado] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create PROCEDURE [dbo].[PA_ListarProveedorEstado]

AS

BEGIN

SET NOCOUNT ON;

SELECT IdProveedor AS ID,NombreCompañia ,NombreContacto ,Ruc,Telefono
FROM PROVEEDOR

WHERE Estado = 'ACTIVO'

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarTipoProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarTipoProducto]

AS

BEGIN

SET NOCOUNT ON;

SELECT Id_TipoProducto AS ID,NOMBRE,ESTADO from TIPO_PRODUCTO

END



GO

/***** Object: StoredProcedure [dbo].[PA_ListarTipoProductoActivo] Script
Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarTipoProductoActivo]

AS

BEGIN

SET NOCOUNT ON;

SELECT * from TIPO_PRODUCTO where Estado = 1

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarTipoUsuario] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarTipoUsuario]

AS

BEGIN

SET NOCOUNT ON;

SELECT distinct IdTipoUsuario , Nombre from Tipo_Usuario

END

GO



/***** Object: StoredProcedure [dbo].[PA_ListarUsuario] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_ListarUsuario]

AS

BEGIN

 SET NOCOUNT ON;

 SELECT u.IdUsuario AS ID,u.Nombre_Usuario AS
 Usuario,u.Contraseña_Usuario as Contraseña,u.IdTipoUsuario,t.Nombre
 ,u.Nombre_Apellidos ,u.dni as DNI

 ,u.Telefono ,u.Direccion,u.Estado from Usuario as u , TIPO_USUARIO as t
 where u.IdTipoUsuario = t.IdTipoUsuario

END

GO

/***** Object: StoredProcedure [dbo].[PA_ListarVentas] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_ListarVentas]

as

SELECT v.Id_Venta ,v.NumeroVenta ,v.FECHA ,v.MontoTotal,t.Nombre as
DOCUMENTO, v.NOMBRES ,v.DniRuc FROM VENTA as v
,TIPO_DOCUMENTO as t

where v.IdTipoDocumento = t.IdTipoDocumento

GO



/****** Object: StoredProcedure [dbo].[PA_LOGIN] Script Date: 18/09/2017
23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[PA_LOGIN]

@nombre VARCHAR(20),

@clave VARCHAR(20),

@validar bit output

as

declare @contar int

select @contar=count(*) from USUARIO

where Nombre_Usuario =@nombre and Contraseña_Usuario = @clave and Estado =
'ACTIVO'

if @contar>0

set @validar=1

else

set @validar =0

GO

/****** Object: StoredProcedure [dbo].[PA_RegistraGarantia] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_RegistraGarantia]

@nom varchar(20),

@precio decimal(8,2),

@des varchar(50),

@estado varchar(10)

as



begin

if exists (select *from GARANTIA

where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),

'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @nom)

raiserror('EL NOMNRE DE GARANTIA YA EXISTE ',16,1)

else

insert into GARANTIA (nombre,PRECIO,DESCRIPCION,estado)

values(@nom,@PRECIO,@DES,@estado)

declare @cantidad as bigint

select @cantidad=coalesce(max([id_Garantia]), 0) from [dbo].[GARANTIA] ;

DBCC CHECKIDENT ('[dbo].[GARANTIA]', RESEED, @cantidad)

END

GO

/****** Object: StoredProcedure [dbo].[PA_RegistrarDetalleOrdenCompra]
Script Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_RegistrarDetalleOrdenCompra]

@idOrden int,

@idpro int,

@cantidad int,

@monto decimal(8,2)

as

insert into DETALLE_ORDEN_COMPRA (IdOrdenCompra,Cantidad ,Monto
,IdProducto)

values(@idOrden ,@cantidad ,@monto ,@idpro)



```
declare @cantida as bigint
```

```
select @cantida=coalesce(max([IdDetalleOrdenCompra]), 0) from  
[dbo].[DETALLE_ORDEN_COMPRA] ;
```

```
DBCC CHECKIDENT ('[dbo].[DETALLE_ORDEN_COMPRA]', RESEED,  
@cantida)
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_RegistrarDetalleVenta] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[PA_RegistrarDetalleVenta]
```

```
@idventa int,
```

```
@idpro int,
```

```
@idGara int,
```

```
@cantidad int,
```

```
@total decimal(8,2),
```

```
@monto decimal(8,2)
```

```
as
```

```
insert into DETALLE_VENTA (Id_Venta ,IdProducto ,IdGarantia ,Cantidad ,Total  
,Monto )
```

```
values(@idventa ,@idpro ,@idGara ,@cantidad ,@total ,@monto )
```

```
declare @cantida as bigint
```

```
select @cantida=coalesce(max([IdDetalleVenta]), 0) from  
[dbo].[DETALLE_VENTA] ;
```

```
DBCC CHECKIDENT ('[dbo].[DETALLE_VENTA]', RESEED, @cantida)
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_RegistrarMarcaProducto] Script  
Date: 18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```



GO

SET QUOTED_IDENTIFIER ON

GO

```
create procedure [dbo].[PA_RegistrarMarcaProducto]
```

```
@nom varchar(20),
```

```
@estado tinyint
```

```
as
```

```
begin
```

```
if exists ( select *from MARCA_PRODUCTO
```

```
where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),
```

```
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')
```

```
= @nom )
```

```
raiserror('EL NOMNRE DE MARCA DEL PRODUCTO YA EXISTE ',16,1)
```

```
else
```

```
insert into MARCA_PRODUCTO (nombre,estado)
```

```
values(@nom,@estado )
```

```
declare @cantidad as bigint
```

```
select @cantidad=coalesce(max([id_Marca]), 0) from  
[dbo].[MARCA_PRODUCTO] ;
```

```
DBCC CHECKIDENT ('[dbo].[MARCA_PRODUCTO]', RESEED, @cantidad)
```

```
END
```

GO

```
/****** Object: StoredProcedure [dbo].[PA_RegistrarOrdenCompra] Script Date:  
18/09/2017 23:07:23 *****/
```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
CREATE procedure [dbo].[PA_RegistrarOrdenCompra]
```

```
@idprove int,
```

```
@idusuario int,
```



@numero varchar(10),

@MontoTotal decimal(8,2),

@estado char(10),

@des varchar(max)

as

insert into ORDEN_COMPRA (idproveedor,IdUsuario ,NumeroOrdenComra
,MontoTotal,Fecha ,Estado ,descripcion)

values(@idprove ,@idusuario ,@numero ,@MontoTotal,GETDATE(),@estado
,@des)

declare @cantidad as bigint

select @cantidad=coalesce(max([idOrdenCompra]), 0) from
[dbo].[ORDEN_COMPRA] ;

DBCC CHECKIDENT ('[dbo].[ORDEN_COMPRA]', RESEED, @cantidad)

GO

/***** Object: StoredProcedure [dbo].[PA_RegistrarProducto] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[PA_RegistrarProducto]

@Nom varchar(40),

@des varchar(max),

@precio_compra decimal(8,2),

@precio_venta decimal(8,2),

@stock int,

@esta varchar(10),

@idtipo int,

@idmarca int

as



begin

if exists (select *from PRODUCTO

where replace(replace(replace(replace(replace(ltrim(upper(Nombre)),

'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @nom)

raiserror('EL NOMNRE DEL PRODUCTO YA EXISTE ',16,1)

else

insert into PRODUCTO (Nombre,Descripcion ,Precio_Compra ,Precio_Venta
,Stock ,Estado,Id_TipoProducto,Id_Marca)

values(@Nom,@des ,@precio_compra
,@precio_venta,@stock,@esta,@idtipo,@idmarca)

declare @cantidad as bigint

select @cantidad=coalesce(max([idPRODUCTO]), 0) from [dbo].[PRODUCTO] ;

DBCC CHECKIDENT ('[dbo].[PRODUCTO]', RESEED, @cantidad)

END

GO

/***** Object: StoredProcedure [dbo].[PA_RegistrarProveedor] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_RegistrarProveedor]

@NomCompa varchar(30),

@nomComtac varchar(30),

@RUC char(11),

@Bancario varchar(50),

@tele char(9),

@dire Varchar(50),

@email Varchar(50),

@esta varchar(10)



as

begin

if exists (select *from PROVEEDOR

where replace(replace(replace(replace(replace(ltrim(upper(NombreCompañia)),

'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')

= @NomCompa or Ruc = @RUC)

raiserror('EL DATO YA EXISTE EN LA BASE DE DATOS',16,1)

else

insert into PROVEEDOR

(NombreCompañia,NombreContacto,Ruc,Bancario,Telefono,Direccion,Email
,Estado)

values(@NomCompa ,@nomComtac,@RUC,@Bancario ,@tele,@dire,@email
,@esta)

declare @cantidad as bigint

select @cantidad=coalesce(max([idProveedor]), 0) from [dbo].[PROVEEDOR] ;

DBCC CHECKIDENT ('[dbo].[PROVEEDOR]', RESEED, @cantidad)

END

GO

/***** Object: StoredProcedure [dbo].[PA_RegistrarVenta] Script Date:
18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[PA_RegistrarVenta]

@idusuario int,

@idtipo int,

@nombres varchar(30),

@dniruc varchar(10),

@NumeroVenta varchar(10),

@MontoTotal decimal(8,2)



as

```
insert into VENTA(IdUsuario ,IdTipoDocumento ,Nombres ,DniRuc,NumeroVenta
,MontoTotal,Fecha)
```

```
values(@idusuario ,@idtipo ,@nombres ,@dniruc ,@NumeroVenta
,@MontoTotal,GETDATE())
```

```
declare @cantidad as bigint
```

```
select @cantidad=coalesce(max([id_Venta]), 0) from [dbo].[VENTA] ;
```

```
DBCC CHECKIDENT ('[dbo].[VENTA]', RESEED, @cantidad)
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_RegistraTipoProducto] Script Date:
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[PA_RegistraTipoProducto]
```

```
@nom varchar(20),
```

```
@estado tinyint
```

```
as
```

```
begin
```

```
if exists ( select *from TIPO_PRODUCTO
```

```
where replace(replace(replace(replace(replace(ltrim(upper(Nombre ))),
```

```
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')
```

```
= @nom )
```

```
raiserror('EL NOMNRE DE TIPO DE PRODUCTO YA EXISTE ',16,1)
```

```
else
```

```
insert into TIPO_PRODUCTO (nombre,estado)
```

```
values(@nom,@estado )
```

```
declare @cantidad as bigint
```



```
select @cantidad=coalesce(max([id_TipoProducto]), 0) from  
[dbo].[TIPO_PRODUCTO] ;
```

```
DBCC CHECKIDENT ('[dbo].[TIPO_PRODUCTO]', RESEED, @cantidad)
```

```
END
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_RegistraUsuario] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[PA_RegistraUsuario]
```

```
@nom varchar(20),
```

```
@pass varchar(20),
```

```
@idtipo int,
```

```
@NomApe varchar(50),
```

```
@dni char(8),
```

```
@telefono varchar(9),
```

```
@direccion varchar(50),
```

```
@estado VARCHAR(10)
```

```
as
```

```
begin
```

```
if exists ( select *from Usuario
```

```
where replace(replace(replace(replace(replace(ltrim(upper(Nombre_Usuario)),
```

```
'Á','A'),'É','E'),'Í','I'),'Ó','O'),'Ú','U')
```

```
= @nom )
```

```
raiserror('EL USUARIO YA EXISTE EN LA BASE DE DATOS',16,1)
```

```
else
```

```
insert into usuario ( Nombre_Usuario ,Contraseña_Usuario  
,IdTipoUsuario,Nombre_ApellidoS ,Dni,Telefono,Direccion,Estado )
```

```
values(@nom,@pass ,@idtipo,@NomApe ,@dni ,@telefono,@direccion,@estado )
```



```
declare @cantidad as bigint
```

```
select @cantidad=coalesce(max([IdUsuario]), 0) from [dbo].[Usuario] ;
```

```
DBCC CHECKIDENT ('[dbo].[usuario]', RESEED, @cantidad)
```

```
END
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_ReportarBoleta] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[PA_ReportarBoleta]
```

```
@idventa int
```

```
as
```

```
select v.*,d.IdProducto ,p.Nombre as NomProducto,p.Precio_Venta ,t.Nombre as  
TIPO, m.Nombre as MARCA,G.Nombre AS GARANTIA, G.Precio AS  
PrecioGarantia
```

```
,d.Cantidad ,d.Total ,d.Monto
```

```
from venta as v , DETALLE_VENTA as d,PRODUCTO as p , TIPO_PRODUCTO  
AS T , MARCA_PRODUCTO AS M,GARANTIA as g
```

```
where v.Id_Venta = d.Id_Venta and d.IdProducto = p.IdProducto AND P.Id_Marca  
= M.Id_Marca AND p.Id_TipoProducto = t.Id_TipoProducto
```

```
and d.IdGarantia = g.Id_Garantia AND v.Id_Venta = @idventa
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[PA_ReportarOrdenCompra] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```



```
create procedure [dbo].[PA_ReportarOrdenCompra]
```

```
@idorden int
```

```
as
```

```
select o.IdOrdenCompra ,o.NumeroOrdenCompra ,o.MontoTotal ,o.Fecha,o.Estado  
,o.Descripcion, pro.NombreCompañia ,pro.Ruc ,pro.Bancario, u.Nombre_Apellidos,
```

```
u.Nombre_Usuario ,p.Nombre ,t.Nombre as TIPO ,m.Nombre AS  
MARCA,p.Precio_Compra ,d.Cantidad ,d.Monto
```

```
from ORDEN_COMPRA as o , DETALLE_ORDEN_COMPRA as d ,  
PRODUCTO as p, TIPO_PRODUCTO AS T , MARCA_PRODUCTO AS M ,  
PROVEEDOR AS PRO,USUARIO AS U
```

```
where o.IdOrdenCompra = d.IdOrdenCompra and d.IdProducto = p.IdProducto  
AND P.Id_Marca = M.Id_Marca AND p.Id_TipoProducto = t.Id_TipoProducto and
```

```
pro.IdProveedor = o.IdProveedor and u.IdUsuario = o.IdUsuario AND  
O.IdOrdenCompra = @idorden
```

```
GO
```

```
/****** Object: Table [dbo].[DETALLE_ORDEN_COMPRA] Script Date:  
18/09/2017 23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[DETALLE_ORDEN_COMPRA](
```

```
[IdDetalleOrdenCompra] [int] IDENTITY(1,1) NOT NULL,
```

```
[IdOrdenCompra] [int] NOT NULL,
```

```
[IdProducto] [int] NOT NULL,
```

```
[Cantidad] [int] NOT NULL,
```

```
[Monto] [decimal](8, 2) NOT NULL,
```

```
CONSTRAINT [PK_DETALLE_ORDEN_COMPRA] PRIMARY KEY  
CLUSTERED
```

```
(
```

```
[IdDetalleOrdenCompra] ASC
```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[DETALLE_VENTA] Script Date: 18/09/2017  
23:07:23 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[DETALLE_VENTA](
```

```
    [IdDetalleVenta] [int] IDENTITY(1,1) NOT NULL,
```

```
    [Id_Venta] [int] NOT NULL,
```

```
    [IdProducto] [int] NOT NULL,
```

```
    [IdGarantia] [int] NOT NULL,
```

```
    [Cantidad] [int] NOT NULL,
```

```
    [Total] [decimal](8, 2) NOT NULL,
```

```
    [Monto] [decimal](8, 2) NOT NULL,
```

```
    CONSTRAINT [PK_DETALLE_VENTA] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [IdDetalleVenta] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[GARANTIA] Script Date: 18/09/2017 23:07:23  
*****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```



GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[GARANTIA](

[Id_Garantia] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [varchar](30) NOT NULL,

[Descripcion] [varchar](50) NULL,

[Precio] [decimal](8, 2) NULL,

[Estado] [varchar](10) NOT NULL,

CONSTRAINT [PK_GARANTIA] PRIMARY KEY CLUSTERED

(

[Id_Garantia] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[MARCA_PRODUCTO] Script Date: 18/09/2017
23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[MARCA_PRODUCTO](

[Id_Marca] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [varchar](30) NOT NULL,

[Estado] [tinyint] NOT NULL,

CONSTRAINT [PK_MARCA_PRODUCTO] PRIMARY KEY CLUSTERED



```
(  
    [Id_Marca] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
  
) ON [PRIMARY]  
  
GO  
SET ANSI_PADDING OFF  
  
GO  
/***** Object: Table [dbo].[ORDEN_COMPRA]   Script Date: 18/09/2017  
23:07:23 *****/  
  
SET ANSI_NULLS ON  
  
GO  
SET QUOTED_IDENTIFIER ON  
  
GO  
SET ANSI_PADDING ON  
  
GO  
CREATE TABLE [dbo].[ORDEN_COMPRA](  
    [IdOrdenCompra] [int] IDENTITY(1,1) NOT NULL,  
    [IdProveedor] [int] NOT NULL,  
    [IdUsuario] [int] NOT NULL,  
    [NumeroOrdenComra] [varchar](10) NOT NULL,  
    [MontoTotal] [decimal](8, 2) NOT NULL,  
    [Fecha] [date] NOT NULL,  
    [Estado] [varchar](10) NOT NULL,  
    [Descripcion] [varchar](50) NULL,  
    CONSTRAINT [PK_ORDEN_COMPRA] PRIMARY KEY CLUSTERED  
(  
    [IdOrdenCompra] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
  
) ON [PRIMARY]
```



```
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PRODUCTO]   Script Date: 18/09/2017 23:07:23
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PRODUCTO](
    [IdProducto] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](40) NOT NULL,
    [Descripcion] [varchar](50) NULL,
    [Id_TipoProducto] [int] NOT NULL,
    [Id_Marca] [int] NOT NULL,
    [Precio_Compra] [decimal](8, 2) NOT NULL,
    [Precio_Venta] [decimal](8, 2) NOT NULL,
    [Stock] [int] NOT NULL,
    [Estado] [varchar](10) NOT NULL,
    CONSTRAINT [PK_PRODUCTO] PRIMARY KEY CLUSTERED
    (
        [IdProducto] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```



/****** Object: Table [dbo].[PROVEEDOR] Script Date: 18/09/2017 23:07:23
*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[PROVEEDOR](

[IdProveedor] [int] IDENTITY(1,1) NOT NULL,

[NombreCompañia] [varchar](30) NOT NULL,

[NombreContacto] [varchar](30) NOT NULL,

[Ruc] [char](11) NOT NULL,

[Bancario] [varchar](50) NULL,

[Telefono] [char](9) NULL,

[Direccion] [varchar](50) NULL,

[Email] [varchar](50) NULL,

[Estado] [varchar](15) NOT NULL,

CONSTRAINT [PK_PROVEEDOR] PRIMARY KEY CLUSTERED

(

[IdProveedor] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

/****** Object: Table [dbo].[TIPO_DOCUMENTO] Script Date: 18/09/2017
23:07:23 *****/

SET ANSI_NULLS ON

GO



SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[TIPO_DOCUMENTO](
    [IdTipoDocumento] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](20) NOT NULL,
    CONSTRAINT [PK_TIPO_DOCUMENTO] PRIMARY KEY CLUSTERED
(
    [IdTipoDocumento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[TIPO_PRODUCTO] Script Date: 18/09/2017
23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[TIPO_PRODUCTO](
    [Id_TipoProducto] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](30) NULL,
    [Estado] [tinyint] NULL,
    CONSTRAINT [PK_TIPO_PRODUCTO] PRIMARY KEY CLUSTERED
(
    [Id_TipoProducto] ASC
```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
/****** Object: Table [dbo].[TIPO_USUARIO]   Script Date: 18/09/2017 23:07:23  
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[TIPO_USUARIO](
```

```
    [IdTipoUsuario] [int] IDENTITY(1,1) NOT NULL,
```

```
    [Nombre] [char](15) NOT NULL,
```

```
    CONSTRAINT [PK_TIPO_USUARIO] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [IdTipoUsuario] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
/****** Object: Table [dbo].[USUARIO]   Script Date: 18/09/2017 23:07:23  
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```



SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[USUARIO](

[IdUsuario] [int] IDENTITY(1,1) NOT NULL,
[Nombre_Usuario] [varchar](20) NOT NULL,
[Contraseña_Usuario] [varchar](20) NOT NULL,
[IdTipoUsuario] [int] NOT NULL,
[Nombre_Apellidos] [varchar](50) NOT NULL,
[Dni] [char](8) NOT NULL,
[Telefono] [char](9) NULL,
[Direccion] [varchar](50) NULL,
[Estado] [varchar](10) NOT NULL,

CONSTRAINT [PK_USUARIO] PRIMARY KEY CLUSTERED

(

[IdUsuario] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[VENTA] Script Date: 18/09/2017 23:07:23 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[VENTA](



```
[Id_Venta] [int] IDENTITY(1,1) NOT NULL,  
[IdUsuario] [int] NOT NULL,  
[IdTipoDocumento] [int] NOT NULL,  
[Nombres] [varchar](50) NOT NULL,  
[DniRuc] [varchar](20) NOT NULL,  
[NumeroVenta] [varchar](10) NOT NULL,  
[MontoTotal] [decimal](8, 2) NOT NULL,  
[Fecha] [date] NOT NULL,  
  
CONSTRAINT [PK_VENTA] PRIMARY KEY CLUSTERED  
(  
    [Id_Venta] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
  
SET ANSI_PADDING OFF  
  
GO  
  
ALTER TABLE [dbo].[DETALLE_ORDEN_COMPRA] WITH CHECK ADD  
CONSTRAINT [FK_DETALLE_ORDEN_COMPRA_ORDEN_COMPRA]  
FOREIGN KEY([IdOrdenCompra])  
REFERENCES [dbo].[ORDEN_COMPRA] ([IdOrdenCompra])  
  
GO  
  
ALTER TABLE [dbo].[DETALLE_ORDEN_COMPRA] CHECK CONSTRAINT  
[FK_DETALLE_ORDEN_COMPRA_ORDEN_COMPRA]  
  
GO  
  
ALTER TABLE [dbo].[DETALLE_ORDEN_COMPRA] WITH CHECK ADD  
CONSTRAINT [FK_DETALLE_ORDEN_COMPRA_PRODUCTO] FOREIGN  
KEY([IdProducto])  
REFERENCES [dbo].[PRODUCTO] ([IdProducto])  
  
GO  
  
ALTER TABLE [dbo].[DETALLE_ORDEN_COMPRA] CHECK CONSTRAINT  
[FK_DETALLE_ORDEN_COMPRA_PRODUCTO]  
  
GO
```



```
ALTER TABLE [dbo].[DETALLE_VENTA] WITH CHECK ADD  
CONSTRAINT [FK_DETALLE_VENTA_GARANTIA] FOREIGN  
KEY([IdGarantia])
```

```
REFERENCES [dbo].[GARANTIA] ([Id_Garantia])
```

```
GO
```

```
ALTER TABLE [dbo].[DETALLE_VENTA] CHECK CONSTRAINT  
[FK_DETALLE_VENTA_GARANTIA]
```

```
GO
```

```
ALTER TABLE [dbo].[DETALLE_VENTA] WITH CHECK ADD  
CONSTRAINT [FK_DETALLE_VENTA_PRODUCTO] FOREIGN  
KEY([IdProducto])
```

```
REFERENCES [dbo].[PRODUCTO] ([IdProducto])
```

```
GO
```

```
ALTER TABLE [dbo].[DETALLE_VENTA] CHECK CONSTRAINT  
[FK_DETALLE_VENTA_PRODUCTO]
```

```
GO
```

```
ALTER TABLE [dbo].[DETALLE_VENTA] WITH CHECK ADD  
CONSTRAINT [FK_DETALLE_VENTA_VENTA] FOREIGN KEY([Id_Venta])
```

```
REFERENCES [dbo].[VENTA] ([Id_Venta])
```

```
GO
```

```
ALTER TABLE [dbo].[DETALLE_VENTA] CHECK CONSTRAINT  
[FK_DETALLE_VENTA_VENTA]
```

```
GO
```

```
ALTER TABLE [dbo].[ORDEN_COMPRA] WITH CHECK ADD CONSTRAINT  
[FK_ORDEN_COMPRA_PROVEEDOR] FOREIGN KEY([IdProveedor])
```

```
REFERENCES [dbo].[PROVEEDOR] ([IdProveedor])
```

```
GO
```

```
ALTER TABLE [dbo].[ORDEN_COMPRA] CHECK CONSTRAINT  
[FK_ORDEN_COMPRA_PROVEEDOR]
```

```
GO
```

```
ALTER TABLE [dbo].[ORDEN_COMPRA] WITH CHECK ADD CONSTRAINT  
[FK_ORDEN_COMPRA_USUARIO] FOREIGN KEY([IdUsuario])
```

```
REFERENCES [dbo].[USUARIO] ([IdUsuario])
```

```
GO
```

```
ALTER TABLE [dbo].[ORDEN_COMPRA] CHECK CONSTRAINT  
[FK_ORDEN_COMPRA_USUARIO]
```



GO

```
ALTER TABLE [dbo].[PRODUCTO] WITH CHECK ADD CONSTRAINT  
[FK_PRODUCTO_MARCA_PRODUCTO] FOREIGN KEY([Id_Marca])
```

```
REFERENCES [dbo].[MARCA_PRODUCTO] ([Id_Marca])
```

GO

```
ALTER TABLE [dbo].[PRODUCTO] CHECK CONSTRAINT  
[FK_PRODUCTO_MARCA_PRODUCTO]
```

GO

```
ALTER TABLE [dbo].[PRODUCTO] WITH CHECK ADD CONSTRAINT  
[FK_PRODUCTO_TIPO_PRODUCTO] FOREIGN KEY([Id_TipoProducto])
```

```
REFERENCES [dbo].[TIPO_PRODUCTO] ([Id_TipoProducto])
```

GO

```
ALTER TABLE [dbo].[PRODUCTO] CHECK CONSTRAINT  
[FK_PRODUCTO_TIPO_PRODUCTO]
```

GO

```
ALTER TABLE [dbo].[USUARIO] WITH CHECK ADD CONSTRAINT  
[FK_USUARIO_TIPO_USUARIO] FOREIGN KEY([IdTipoUsuario])
```

```
REFERENCES [dbo].[TIPO_USUARIO] ([IdTipoUsuario])
```

GO

```
ALTER TABLE [dbo].[USUARIO] CHECK CONSTRAINT  
[FK_USUARIO_TIPO_USUARIO]
```

GO

```
ALTER TABLE [dbo].[VENTA] WITH CHECK ADD CONSTRAINT  
[FK_VENTA_TIPO_DOCUMENTO] FOREIGN KEY([IdTipoDocumento])
```

```
REFERENCES [dbo].[TIPO_DOCUMENTO] ([IdTipoDocumento])
```

GO

```
ALTER TABLE [dbo].[VENTA] CHECK CONSTRAINT  
[FK_VENTA_TIPO_DOCUMENTO]
```

GO

```
ALTER TABLE [dbo].[VENTA] WITH CHECK ADD CONSTRAINT  
[FK_VENTA_USUARIO] FOREIGN KEY([IdUsuario])
```

```
REFERENCES [dbo].[USUARIO] ([IdUsuario])
```

GO

```
ALTER TABLE [dbo].[VENTA] CHECK CONSTRAINT  
[FK_VENTA_USUARIO]
```

GO



USE [master]

GO

ALTER DATABASE [KVFACTURACION] SET READ_WRITE

GO

BIBLIOGRAFIA

- Vélez, D., Flores, O., Ch, S., Castillo, A., Andres, E., vargas, j., Santana, T., Arenas, E., Uría, J., Uría, J., jaimes, l., Balderas, M., Sanhueza, R., Balderas, M., morales, j., Balderas, M., Guerrero, S., Celeita, P., Guzman, L., C., A., niето, j., ceron, c., garcía, f., Rose, J. and Romero, D. (2017). *Normas Apa 2016 – Edición 6*. [online] Normasapa.net. Disponible en: <http://normasapa.net/normas-apa-2016/> [Acceso 15 Sep. 2017].
- Robidoux, G. (2017). *SQL Server INFORMATION_SCHEMA views Tutorial*. [online] Mssqltips.com. Disponible en: <https://www.mssqltips.com/sqlservertutorial/179/sql-server-informationschema-views-tutorial/> [Acceso 15 Sep. 2017].
- Desarrolloweb.com. (2017). ¿Qué es un diccionario de datos? [online] Disponible en: <https://desarrolloweb.com/faq/452.php> [Acceso 15 Sep. 2017].