



INSTITUTO TECNOLÓGICO
“CORDILLERA”

CARRERA DE ANÁLISIS DE SISTEMAS

SISTEMATIZACIÓN DE LOS PROCESOS DE FACTURACIÓN
ELECTRÓNICA MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA
VIASCALA UBICADA EN LA CIUDAD DE QUITO.

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

Autor: Maycol Vinicio Flores Flores

Tutor: Lic. Carlos Romero

Quito, Abril 2015

DECLARACIÓN DE APROBACIÓN TUTOR Y LECTOR

En mi calidad de tutor del trabajo sobre el proyecto: **“SISTEMATIZACIÓN DE LOS PROCESOS DE FACTURACIÓN ELECTRÓNICA MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA VIASCALA UBICADA EN LA CIUDAD DE QUITO”**, presentado por el ciudadano: Flores Flores Maycol Vinicio, estudiante de la Escuela de Análisis y Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril del 2015

Lic. Carlos Romero

TUTOR

Ing. Jaime Padilla

LECTOR

Ing. Cristian Prado

DIRECTOR DE PROYECTOS

Ing. Hugo Heredia

DIRECTOR DE CARRERA

DECLARATORIA

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 los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Maycol Vinicio Flores Flores

CESIÓN DE DERECHOS

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante Flores Flores Maycol Vinicio, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “Sistematización de los procesos de facturación electrónica mediante una aplicación web para la empresa Viascala ubicada en la ciudad de Quito”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita

todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de

éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvencción, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los 27 días del mes de Abril del dos mil quince.

f) _____

C.C. N° 1750402859
CEDENTE

f) _____

Instituto Superior Tecnológico Cordillera
CESIONARIO

AGRADECIMIENTO

Agradezco a Dios, a mis padres por la oportunidad que me dieron para superarme, por su apoyo incondicional, gracias a ellos pude culminar esta enriquecedora etapa de mi vida.

A un ser admirable en especial, que inculcó en mí, confianza y perseverancia para vencer todos los obstáculos, a quien considero amiga y madre, gracias por estar conmigo dándome fuerzas para seguir adelante.

A mis docentes y compañeros con los que he compartido tantas experiencias en las aulas permitiéndome adquirir sabidurías que me servirán en mi futura vida como profesional.

A mi tutor que siempre supo aconsejarme, por su comprensión y paciencia. Gracias Licenciado Carlos Romero.

DEDICATORIA

A mis Padres Marcelo y Dolores;
por su comprensión, amor y apoyo absoluto;
a mis hermanos Sandy y Darwin;
a todos mis familiares y amigos;
en especial a una persona
quela llevo en mi corazón por estar a mi lado
siempre en los buenos y malos momentos.

ÍNDICE GENERAL

Título	Página
DECLARACIÓN DE APROBACIÓN TUTOR Y LECTOR	i
DECLARATORIA.....	ii
CESIÓN DE DERECHOS	iii
AGRADECIMIENTO	vi
DEDICATORIA	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS	xiv
ÍNDICE DE FIGURAS	xvii
RESUMEN EJECUTIVO	xxii
ABSTRACT	xxiv
Capítulo I: Antecedentes	1
1.01 Contexto	1
1.02 Justificación.....	2
1.03 Definición del problema central	3
1.03.1. Análisis de la Matriz de Fuerzas	4
Capítulo II: Análisis de Involucrados	5
2.01 Requerimientos.....	6
2.01.1 Descripción del sistema actual.....	6
2.01.2 Visión y alcance.....	6
2.01.2.1. Visión:	6

2.01.2.2. Alcance:	6
2.01.3 Entrevistas	8
2.01.4 Matriz de Requerimientos	9
2.01.5 Descripción detallada	10
2.02 Mapeo de involucrados:	19
2.03 Matriz de involucrados	20
Capítulo III: Problemas y Objetivos	21
3.01 Árbol de problemas	21
3.02 Árbol de objetivos	22
3.02.01 Análisis del Árbol de Problemas	23
3.02.02 Análisis del Árbol de Objetivos	23
3.03 Diagramas de casos de uso	24
3.04 Casos de uso de realización	27
3.05 Diagrama de secuencias del sistema	36
3.06 Especificación de casos de uso	39
Capítulo IV: Análisis de Alternativas	42
4.01 Matriz de Análisis de Alternativas	42
4.01.01 Análisis Matriz de Alternativas	43
4.02 Matriz de Impactos de Objetivos	45
4.03 Estándares para el Diseño de Clases	46
4.03.02 Elementos	48

4.04 Diagrama de Clases	50
4.05 Modelo Lógico – Físico	51
4.06 Diagrama de Componentes.....	52
4.07. Diagramas de Estrategias	53
4.08 Matriz de Marco Lógico.....	54
4.09 Vistas Arquitectónicas.....	55
4.09.01 Vista Lógica.....	55
4.09.02 Vista Física	55
4.09.03 Vista Desarrollo	56
4.09.04. Vista de Procesos	56
Capítulo V: Propuesta	57
5.01 Especificación de estándares de programación	57
5.01.01 Objetos para Ejecución	58
5.01.02 Objetos Multimedia	58
5.01.03 Objetos Fuentes.....	59
5.01.04 Arquitectura de Software	59
5.01.05 Arquitectura 3 capaz cliente/ servidor	60
5.01.06.02 Comentario de las Funciones	60
5.01.06.03 Estructuras de Control.....	61
5.01.06.04 Uso de Variables	61
5.02 Diseño de Interfaces de Usuario.....	62

5.02.01 Tipos de Interfaces de Usuario	62
5.02.01.01 Interfaz de Hardware.....	62
5.02.01.02 Interfaz de Software	63
5.02.01.03 Interfaz de Software-Hardware	63
5.03 Especificación de pruebas de unidad.....	71
5.04 Especificación de pruebas de aceptación	74
5.05 Especificación de pruebas de carga.....	77
5.06 Configuración del Ambiente mínima/ideal	80
Capítulo VI: Aspectos Administrativos	84
6.01 Recursos	84
6.01.01 Recurso Humano.....	84
6.01.02 Recurso Material.....	84
6.01.03 Recurso Tecnológico	85
6.02 Presupuesto.....	85
6.03 Cronograma	86
Capítulo VII: Conclusiones y Recomendaciones.....	87
7.01 Conclusiones	87
7.02 Recomendaciones	88
ANEXOS.....	89
A.01 Diagrama de Clases	90
A.02 Modelo Lógico	91

A.03 Modelo Físico	92
A.04 Vista Lógica.....	93
A.05 Cronograma de Actividades	94
A.06 Manual de Usuario.....	95
1. Introducción	96
2. Interfaces de Usuario	96
2.1 Inicio de Sesión	96
2.2 Página Principal:	97
2.3 Página de Configuración:	98
2.4 Página de Usuarios:	99
2.5 Página de Clientes:	100
2.6 Página de Proveedores:	102
2.7 Página de Productos (inventarios):	104
2.8 Página de Factura Electrónica:	106
A.07 Manual Técnico	112
1. Introducción	113
2. Objetivo.....	113
3. Contenido.....	113
3.1 Declarar entidades	113
3.2 Insertar cliente	115
3.3 Actualizar clientes	116

3.4 Eliminar cliente	117
3.5 Listar cliente en grid view	117
3.6 Capa de negocios.....	118
3.7 Validacion de campos	119
3.8 Autocomplete	119
3.9 Operaciones automáticas de cálculo en comprobantes	120
3.10 Validar cedula	121
3.11 Validar Ruc	122
3.12 Encriptar y desencriptar contraseña	122
3.13 Coneccion a la base de datos.....	124
3.14 PDF Y XML.....	124
3.15 Script de base de datos	127
3.16 Procedimientos almacenados	138
A.08 Manual de Instalación.....	145
BIBLIOGRAFÍA	163

ÍNDICE DE TABLAS

Título	Página
Tabla 1.....	3
Tabla 2	8
Tabla 3.....	9
Tabla 4.....	10
Tabla 5.....	11
Tabla 6.....	12
Tabla 7.....	13
Tabla 8.....	14
Tabla 9.....	15
Tabla 10.....	16
Tabla 11.....	17
Tabla 12.....	18
Tabla 13.....	20
Tabla 14.....	28
Tabla 15.....	29
Tabla 16.....	30
Tabla 17.....	31
Tabla 18.....	32
Tabla 19.....	33
Tabla 20.....	34
Tabla 21.....	35
Tabla 22.....	39
Tabla 23.....	39

Tabla 24.....	40
Tabla 25.....	40
Tabla 26.....	41
Tabla 27.....	41
Tabla 28.....	42
Tabla 29.....	45
Tabla 30.....	48
Tabla 31.....	54
Tabla 32.....	58
Tabla 33.....	58
Tabla 34.....	59
Tabla 35.....	71
Tabla 36.....	72
Tabla 37.....	72
Tabla 38.....	73
Tabla 39.....	73
Tabla 40.....	74
Tabla 41.....	75
Tabla 42.....	75
Tabla 43.....	76
Tabla 44.....	76
Tabla 45.....	77
Tabla 46.....	78
Tabla 47.....	79
Tabla 48.....	81

Tabla 49.....	85
Tabla 50.....	86

ÍNDICE DE FIGURAS

Título	Página
Figura 1	19
Figura 2	21
Figura 3	22
Figura 4	24
Figura 5	25
Figura 6	25
Figura 7	26
Figura 8	26
Figura 9	27
Figura 10	27
Figura 11.	28
Figura 12.	29
Figura 13	31
Figura 14	32
Figura 15	33
Figura 16	34
Figura 17	36
Figura 18	36
Figura 19	37
Figura 20	37
Figura 21	38
Figura 22	38

Figura 23	44
Figura 24	52
Figura 25	53
Figura 26	55
Figura 27	56
Figura 28	56
Figura 29	63
Figura 30	63
Figura 31	64
Figura 32	64
Figura 33	65
Figura 34	65
Figura 35	66
Figura 36	66
Figura 37	67
Figura 38	67
Figura 39	68
Figura 40	68
Figura 41	69
Figura 42	69
Figura 43	70
Figura 44	70
Figura 45	78
Figura 46	80
Figura 47	90

Figura 48	91
Figura 49	92
Figura 50	93
Figura 51	94
Figura 52	96
Figura 53	97
Figura 54	98
Figura 55	98
Figura 56	99
Figura 57	99
Figura 58	100
Figura 59	100
Figura 60	101
Figura 61	101
Figura 62	102
Figura 63	102
Figura 64	103
Figura 65	103
Figura 66	104
Figura 67	104
Figura 68	105
Figura 69	105
Figura 70	106
Figura 71	106
Figura 72	107

Figura 73	107
Figura 74	108
Figura 75	108
Figura 76	109
Figura 77	109
Figura 78	110
Figura 79	110
Figura 80	111
Figura 81	111
Figura 82	150
Figura 83	150
Figura 84	151
Figura 85	151
Figura 86	152
Figura 87	152
Figura 88	153
Figura 89	153
Figura 90	154
Figura 91	155
Figura 92	155
Figura 93	156
Figura 94	157
Figura 95	157
Figura 96	158
Figura 97	158

Figura 98	159
Figura 99	159
Figura 100	160
Figura 101	160
Figura 102	160
Figura 103	161
Figura 104	161
Figura 105	162

RESUMEN EJECUTIVO

El presente proyecto se ha elaborado con la finalidad de crear un sistema de facturación electrónica mediante una aplicación web para la empresa Viascala ubicada en la ciudad de Quito, cuya actividad se desarrolla en el área de ventas por lo tanto es necesario optimizar los servicios de facturación electrónica para brindar un servicio de calidad a sus clientes. Básicamente el proyecto contiene siete capítulos a desarrollarse, en el primero y segundo capítulo se analiza la situación actual de la empresa Viascala a la que se pretende mejorar, a través del diagnóstico se determinaron fortalezas y oportunidades; una de las principales fortalezas es la factura electrónica la cual nos permite optimizar el servicio y disminuir el tiempo en sus actividades brindando seguridad de información y disminuyendo fraudes, además se detalla que entidades están involucradas en el proyecto tanto de manera directa como indirecta.

En el capítulo tres se detallan los problemas y causas que se presentan en la situación actual del manejo de la empresa así como también los objetivos que se debe cumplir para llegar a la situación deseada y poder mejorar para brindar un mejor servicio. También se detalla gráficamente cada uno de los procesos que se llevan a cabo dentro de la empresa, mediante el uso del lenguaje unificado de modelado más conocido como UML, cada gráfico es descrito para lograr un mejor entendimiento del proceso que se pretende mostrar. En el capítulo cuatro se presenta la matriz de análisis de alternativas mediante la cual se pretende demostrar los fines que se desea cumplir y de tal manera promover el cambio deseado, además se presenta el diagrama de clases mediante el cual se crea el modelo lógico y físico, el mismo que determina como está estructurada la base de datos del sistema.

En el capítulo cinco se detalla los estándares de programación que se utilizan para el correcto manejo del lenguaje ASP.NET en la elaboración de la aplicación, además se especifica cómo se le dará el nombre a cada control que se utiliza para tomar la información que se desee ingresar a la base de datos. También se muestra como se ha diseñado cada uno de los formularios, los cuales son destinados para diferentes procesos, como para el listado de registros, la eliminación y modificación de los mismos.

El capítulo seis detalla los aspectos administrativos, en donde se ha establecido los recursos tanto económicos como humanos que se ha utilizado en la planificación y elaboración del proyecto para el cual se fomentó el uso de software libre ya que se utilizó como lenguaje de programación ASP.NET, como motor de base de datos SQL Server 2008, con el objetivo de minimizar los costos de elaboración e implementación del sistema.

Por último, el capítulo siete presenta las conclusiones y recomendaciones que se ha obtenido a través de la elaboración del proyecto.

ABSTRACT

This project has been developed with the aim of creating an electronic billing system using a web application for the company Viascala Located in the city of Quito, which is active in the sales area is therefore necessary to optimize billing services electronics to provide a quality service to its customers. Basically the project contains seven chapters to develop in the first and second chapter the current situation of the company Viascala which aims to improve analyzes, through diagnosis strengths and opportunities were identified; one of the main strengths is the electronic invoice which allows us to optimize service and reduce time in its activities providing information security and reducing fraud also detailed that entities are involved in the project both directly and indirectly.

In chapter three the problems and causes that occur in the current situation of the management of the company as well as the objectives that must be met to reach the desired situation and to improve to better serve detailing. Also graphically detailing each of the processes that take place within the company, through the use of modeling known as UML unified language, each graph is described for a better understanding of the process that is intended to show. In chapter four matrix analysis of alternatives by which it is intended to demonstrate the ends that are to be met and so promote the desired change occurs, besides the class diagram in which the logical and physical model is created is presented , determined as it is structured that the database system.

In chapter five programming standards used for the correct handling of ASP.NET lenguaje in developing detailed implementation also specifies how to give a name to each control that is used to make the information wishing to enter the database. Also

shown as designed each of the forms, which are intended for different processes to the list of records, removal and modification thereof.

The sixth chapter details the administrative, where it has been established both economic and human resources that has been used in the planning and preparation of the project for which the use of free software is encouraged as it was used as a language of ASP programming. NET, such as database engine SQL Server 2008 data, in order to minimize the costs of developing and implementing the system.

Finally, chapter seven presents the conclusions and recommendations obtained through the development of the project.

Capítulo I: Antecedentes

1.01 Contexto

La empresa “VIASCALA” fue creada en 1995 en Quito-Ecuador especializada en el diseño, planificación, fabricación e instalación stands, islas, muebles de oficina, hogar y mueblería en general, la cual cuenta con un personal altamente capacitado y combinado con la experiencia alcanzada en sus años de trabajo, brindan una gran oportuna respuesta a todas las exigencias de nuestros clientes, la misma que les ha permitido tener una buena presencia en el mercado empresarial de muebles.

Su misión como empresa es indemnizar a las familias ecuatorianas así como a otras empresas o particulares con sus productos de atractivas características en un marco de originalidad entregando a cada uno de nuestros distinguidos clientes eficacia, detalle y comodidad.

Su visión como empresa es llegar a ser una de las mueblerías de mayor calidad y reconocimiento en el país, destacándose por la calidad y cantidad de productos innovadores, con ideas frescas y creativas para todos los gustos, al alcance de todo bolsillo.

1.02 Justificación

(santiago, 2008-2009) “La importancia de realizar un sistema de facturación electrónica surge de la innegable necesidad de otorgar validez legal al ejemplar electrónico de los documentos tributarios de compra y venta tales como facturas, notas de crédito, notas de débito, guías de despacho, ya que con ello se optimiza la operación de las empresas y del Servicio de Rentas Internas (SRI).

Está demostrado que la Facturación Electrónica disminuye la evasión tributaria y la corrupción, aumenta la transparencia, ayuda a fortalecer el gobierno corporativo, y en general mejora la economía. Factura Electrónica se encuentra desarrollándose con la participación de pocas empresas, sin embargo se espera que en un futuro este sistema sea adoptado por más contribuyentes. Para poder fomentar la adopción de la facturación electrónica en Ecuador, así como poder trazar estrategias de difusión y promover políticas para su masificación es fundamental tener un diagnóstico claro y actualizado de los estados de avance y penetración del sistema.

Este proyecto tiene por objeto mejorar el panorama actual de la adopción de factura electrónica en el país ya sea desde el punto de vista de las empresas usuarias del sistema al igual que desde la Administración Tributaria. De acuerdo a las experiencias de países como España, México, Chile han llegado a determinar que la facturación electrónica genera grandes beneficios tanto del emisor como del receptor y que existen muchas razones para adoptar la facturación electrónica, que van desde los económicos hasta los ecológicos”.

1.03 Definición del problema central

Tabla 1

Análisis de Matriz de Fuerzas T

SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
Inadecuado control de los procesos de facturación	Inexistencia de un sistema de facturación				Adecuado control de los procesos de facturación
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Es obligado por el SRI	2	3	3	2	Personal no capacitado
Es por seguridad de la empresa	2	3	4	3	Documentos no legalizados de la empresa
Por rapidez en sus actividades diarias	3	4	4	2	No cumplimiento con los requisitos establecidos.
Asignación correcta de los procesos de facturación.	3	4	4	3	No cuentan con capital necesario

Notas:

1 – 2 = Nivel Bajo

3 – 4 = Nivel Media

5 = Nivel Alto

I = Intensidad

PC = Potencial de Cambio

Fuente: Maycol Flores

1.03.1. Análisis de la Matriz de Fuerzas

Al finalizar el análisis la matriz de fuerzas tenemos como resultado las intensidades y sus potenciales de cambio, las fuerzas impulsadoras y bloqueadoras, y por tal motivo en los procesos que se realizan en la empresa se ven mejoras.

Fuerza Impulsadora: **Es obligado por el SRI**

I = 2 Es el ente que determina el marco legal sobre las empresas en el Ecuador.

PC = 3 Dentro de la planificación del SRI se encuentra la implementación de la Facturación electrónica para todas las empresas en el Ecuador

Fuerza Impulsadora: **Es por seguridad de la empresa**

I = 2 Se tiene una mejor seguridad de información de la empresa.

PC = 3 Se obtiene un control de no duplicidad y pérdida de información.

Fuerza Impulsadora: **Por rapidez en sus actividades diarias**

I = 3 Se logra rapidez en los procesos de la facturación mediante el uso de un aplicativo web.

PC = 4 El tener un aplicativo web permitirá a la empresa mejorar el servicio y aumentar la productividad de la misma.

Fuerza Impulsadora: **Asignación correcta de los procesos de facturación.**

I = 3 Se evidencia la situación real sobre los procesos que se realizan en la empresa.

PC = 4 Se obtiene un equilibrio que determine los resultados reales que se realizan en la empresa.

Fuerza Bloqueadora: **Personal no capacitado**

I = 3 No existen suficientes capacitaciones a los empleados.

PC = 2 Falta de conocimientos

Fuerza Bloqueadora: **Documentos no legalizados de la empresa**

I = 4 No existe la suficiente responsabilidad para tener la documentación en regla.

PC = 3 Falta de permisos para el funcionamiento de la empresa.

Fuerza Bloqueadora: **No cumplimiento con los requisitos establecidos.**

I = 4 No cumplimiento de requisitos establecidos por el SRI.

PC = 2 Falta de información.

Fuerza Bloqueadora: **No cuentan con capital necesario**

I = 4 No existe una organización adecuada para los ingresos y egresos de la empresa.

PC = 3 Falta de organización y conocimiento.

Capítulo II: Análisis de Involucrados

2.01 Requerimientos

2.01.1 Descripción del sistema actual

Actualmente el sistema se lo maneja a mano y haciendo declaraciones manuales con el SRI y emitiendo facturas pre-impreso, archivado en carpetas según fechas de elaboración. La dependencia de la empresa a un contador ha producido una despreocupación de actualización a un sistema por lo que se tiene manejos 70% a mano de los procesos totales de la empresa, debido a los problemas suscitados a lo largo del tiempo se ha tomado la decisión de la implementación del sistema y a causas de los nuevos mandatos establecidos por el SRI es un requerimiento cumplir con la facturación electrónica.

2.01.2 Visión y alcance

2.01.2.1. Visión:

Cumplir con las leyes tributarias establecidas por el SRI, adquiriendo un software de facturación electrónica el mismo que garantice a la empresa tener altos estándares de seguridad de información para minimizar riesgos de fraude, ahorrando costos operacionales y mejorando la imagen de la empresa.

2.01.2.2. Alcance:

Crear un software orientado a la web que automatice los procesos de facturación electrónica el mismo que contendrá con los siguientes módulos:

Módulo de Seguridad:

- Seguridad con sesiones (Administrador y Cliente) con sus respectivos privilegios dentro del sistema.
- Seguridad con tiempo límite de sesión.
- Seguridad con encriptación.

Módulo de Mantenimiento:

- Mantenimiento de tablas donde se podrá (insertar, actualizar y eliminar) la información de la base de datos.

Reglas del Negocio:

- **Módulo de Facturación:**
 - Factura Inteligente (llenado automático).
 - Conversión de factura a formato XML.
 - Conversión de factura a formato PDF.
- **Módulo de Inventario:**
 - Se detallara los ingresos y salidas de los productos (Kardex).

Módulo de Reportes:

- Se detallara las facturas de las ventas de la empresa.

2.01.3 Entrevistas

Tabla 2 Diseño de entrevista

DISEÑO ENTREVISTA		
Identificador: 001		
Preguntas	Objetivos	Análisis posterior
¿Cuál es la situación actual con respecto a la facturación en la empresa?	Determinar la situación actual de la empresa.	La empresa carece de un sistema de facturación por lo cual trae complicaciones en diferentes áreas tales como contabilidad e inventarios.
¿Qué inconvenientes presenta la empresa respecto a la facturación?	Determinar los problemas actuales de la empresa.	El proceso actualmente se lo lleva a cabo a mano, en cuadernos se registra diariamente la facturación teniendo una copia de la factura almacenada en un archivo con riesgos de que esta se pierda y no mantienen actualizada la información en tiempo real.
¿Qué beneficios le daría un sistema de facturación a la empresa?	Obtener un listado de los beneficios de usar un sistema.	El sistema optimizará el tiempo de respuesta. Se reducirá al mínimo el uso de papel.
¿Qué procesos se desea sistematizar y por qué?	Determinar los procesos que se desean mejorar y sus razones.	Se desea sistematizar el proceso de facturación para facilitar el trabajo de contabilidad así como mantener la información segura, apegándose a las nuevas normas establecidas por el SRI.
¿Quién tendrá acceso al sistema de Facturación?	Determinar las personas que tendrán acceso a la información y al sistema.	El sistema será utilizado por el departamento de contabilidad tales como ventas así como tendrá acceso el administrador en este caso los dos socios.
¿Cuáles son los factores críticos para determinar el éxito del sistema?	Determinar los puntos más importantes para que el sistema sea un éxito.	Los factores más importantes serán la seguridad a la información, el corto tiempo de respuesta que debe tener y el cumplimiento de los requisitos impuestos por el SRI.

Notas: En esta tabla se detalla la entrevista que se llevó a cabo para hacer el levantamiento de información.

Fuente: Maycol Flores

2.01.4 Matriz de Requerimientos

Tabla 3

Especificación de requerimientos funcionales y no funcionales.

MATRIZ DE REQUERIMIENTOS						
Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Involucrados
REQUERIMIENTOS FUNCIONALES						
RF001	Los clientes deben contener Ruc o Cedula para ser registrados.	Dueño	Alta	Sistema	En revisión	Ventas, cobranzas, contabilidad
RF002	Se tendrá un control total de facturas	Dueño	Alta	Sistema	En revisión	Sri, contabilidad
RF003	Se permitirá el registro de productos	Dueño	Media	Usuario	En revisión	Ventas, contabilidad, socios
RF004	Se generara la factura electrónica.	Dueño	Alta	Sistema	En revisión	Ventas, cobranzas, contabilidad
RF005	Se podrá elegir el tipo de pago	Dueño	Alta	Sistema	En revisión	Ventas, cobranzas, contabilidad, socios
RF006	Se permitirá devoluciones	Dueño	Alta	Sistema	En revisión	Ventas, cobranzas, contabilidad, socios
REQUERIMIENTOS NO FUNCIONALES						
NRF001	La aplicación no podrá ser instalada en dispositivos móviles.	Dueño	Media	Sistema	En revisión	IT
NRF002	La aplicación no funcionara con navegadores desactualizados	Dueño	Alta	Usuario	En revisión	Ventas, cobranzas, contabilidad, socios
NRF003	El sistema no cumplirá con todos los procesos del módulo de contabilidad.	Socio	Media	Sistema	En revisión	Ventas, cobranzas, contabilidad, socios

Notas: En esta tabla se detallan los requerimientos funcionales y no funcionales que se necesitan establecer en el sistema

1 – 3 = Nivel de Prioridad Baja
4 – 6 = Nivel de Prioridad Media
7 – 10 = Nivel de Prioridad Alta

Fuente: Maycol Flores

2.01.5 Descripción detallada

Tabla 4

Especificación de requerimiento funcional N°1

Los clientes deben contener Ruc o Cedula para ser registrados		ESTADO	Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	RF001		
TIPO DE REQUERIMIENTO	Critico	TIPO DE REQUERIMIENTO	Funcional
DATOS DE ENTRADA	Datos completos de clientes.		
DESCRIPCION	Los clientes deben contener RUC o Cedula para poder ser registrados en el sistema.		
DATOS DE SALIDA	Información de la factura a nombre del cliente		
RESULTADOS ESPERADOS	Generar la factura con los datos del cliente ingresados en el sistema.		
ORIGEN	Dueño		
DIRIGIDO A	Usuarios de acceso al sistema		
PRIORIDAD	5		
REQUERIMIENTOS ASOCIADOS	Interfaz gráfica intuitiva y seguridad para el uso de la aplicación.		
ESPECIFICACION			
PRECONDICIONES	Es obligatorio que el cliente sea registrado con sus datos en el sistema.		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Si cumple con lo establecido por el cliente.		

RF001= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja

4 – 6 = Nivel de Prioridad Media

7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 001.

Fuente: Maycol Flores

Tabla 5

Especificación de requerimiento funcional N°2

Se tendrá un control total de facturas.		ESTADO	Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	RF002		
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	FUNCIONAL
DATOS DE ENTRADA	Datos de la empresa, clientes y costos.		
DESCRIPCION	Se realizara una verificación de datos para la generación de la factura.		
DATOS DE SALIDA	Se obtendrá la Factura electrónica		
RESULTADOS ESPERADOS	Cumplir parámetros establecidos por el SRI para efectuar la facturación electrónica mediante este archivo XML.		
ORIGEN	Empresa		
DIRIGIDO A	SRI		
PRIORIDAD	5		
REQUERIMIENTOS ASOCIADOS	Interfaz gráfica intuitiva y seguridad para el uso de la aplicación.		
ESPECIFICACION	Especificaciones establecidas en los parámetros del SRI.		
PRECONDICIONES	Es necesario convertir la factura en formato XML para su validación con el SRI.		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Siguiendo requerimientos del SRI.		

RF002= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja
4 – 6 = Nivel de Prioridad Media
7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 002.

Fuente: Maycol Flores

Tabla 6

Especificación de requerimiento funcional N°3

Se permitirá el registro de productos		ESTADO	Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios
FECHA DE CREACION	DE 18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	RF003		
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	FUNCIONAL
DATOS DE ENTRADA	Productos de entrada o de salida (Stock)		
DESCRIPCION	Se llevara un registro de productos de entrada y de salida para facilitar los inventarios.		
DATOS DE SALIDA	Reporte de acuerdo a patrón de búsqueda.		
RESULTADOS ESPERADOS	Generar reportes de productos según la necesidad de forma rápida.		
ORIGEN	Empresa		
DIRIGIDO A	Usuarios del sistema		
PRIORIDAD	4		
REQUERIMIENTOS ASOCIADOS	Generación de reportes de forma rápida y sencilla		
ESPECIFICACION	Formato estándar con información requerida.		
PRECONDICIONES	Es necesario registrar los productos en el stock para verificar existencias		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Acordado con el dueño de la empresa.		

RF003= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja
4 – 6 = Nivel de Prioridad Media
7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 003.

Fuente: Maycol Flores

Tabla 7

Especificación de requerimiento funcional N°4

Se generara la factura electrónica.		ESTADO	Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	RF004		
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	FUNCIONAL
DATOS DE ENTRADA	Información del cliente y la venta.		
DESCRIPCION	Ingresando los datos del cliente y los detalles de la venta se obtiene la factura.		
DATOS DE SALIDA	Datos de la empresa, cliente y productos.		
RESULTADOS ESPERADOS	Generar la factura electrónica con su respectiva validación con el SRI.		
ORIGEN	Empresa		
DIRIGIDO A	Usuarios del sistema		
PRIORIDAD	5		
REQUERIMIENTOS ASOCIADOS	Requerimientos del SRI		
ESPECIFICACION	Se enviara la factura al SRI para su validación		
PRECONDICIONES	Es necesario convertir la factura en formato XML para su validación con el SRI		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Si cada factura es validada sin errores con el SRI		

RF004= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja
4 – 6 = Nivel de Prioridad Media
7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 004.

Fuente: Maycol Flores

Tabla 8

Especificación de requerimiento funcional N°5

Se podrá elegir el tipo de pago		ESTADO	Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	RF005		
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	FUNCIONAL
DATOS DE ENTRADA	Información de forma de pago.		
DESCRIPCION	Se ingresara la forma de pago acordada con el cliente		
DATOS DE SALIDA	Generar la factura sin problemas.		
RESULTADOS ESPERADOS	Tener un control con las formas de pagos por los clientes.		
ORIGEN	Empresa		
DIRIGIDO A	Usuarios del sistema		
PRIORIDAD	5		
REQUERIMIENTOS ASOCIADOS	Banco		
ESPECIFICACION	Se procederá con la cancelación una vez autorizada la forma de pago		
PRECONDICIONES	Es necesario ingresar la forma de pago en el sistema.		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Si se elige correctamente la forma de pago con los clientes.		

RF005= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja

4 – 6 = Nivel de Prioridad Media

7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 005.

Fuente: Maycol Flores

Tabla 9

Especificación de requerimiento funcional N°6

Se permitirá devoluciones		ESTADO		Análisis
CREADO POR	Dueño	ACTUALIZAR POR	Socios	
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014	
IDENTIFICADOR	RF006			
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	FUNCIONAL	
DATOS DE ENTRADA	Ingresar productos a devolver en el sistema.			
DESCRIPCION	Se ingresara las devoluciones en el sistema para registrar en el stock los productos de entrada y de salida.			
DATOS DE SALIDA	Devolución y registro de información exitosa			
RESULTADOS ESPERADOS	Tener un control con las devoluciones para verificarlas en el stock			
ORIGEN	Empresa			
DIRIGIDO A	Usuarios del sistema			
PRIORIDAD	5			
REQUERIMIENTOS ASOCIADOS	Ninguna			
ESPECIFICACION	Se procederá con devolución una vez que haya registrado en el sistema.			
PRECONDICIONES	Es necesario ingresar la devolución en el sistema.			
POSCONDICIONES	Ninguna			
CRITERIOS DE ACEPTACION	Ingreso satisfactorio de información de devoluciones en el sistema.			

RF006= identificador del requerimiento funcional

1 – 3 = Nivel de Prioridad Baja
4 – 6 = Nivel de Prioridad Media
7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos y especificaciones que interfieren en el requerimiento funcional 006.

Fuente: Maycol Flores

Tabla 10

Especificación de requerimiento no funcional N°1

La aplicación no podrá ser instalada en dispositivos móviles.		ESTADO	Análisis
CREADO POR	Socios	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	NRF001		
TIPO DE REQUERIMIENTO	NO CRITICO	TIPO DE REQUERIMIENTO	NO FUNCIONAL
DATOS DE ENTRADA	Sistema y base de datos		
DESCRIPCION	El alojamiento del sistema deberá ser en un servidor de escritorio.		
DATOS DE SALIDA	Sistema y base de datos		
RESULTADOS ESPERADOS	La implementación deberá ser correcta en el servidor para el funcionamiento en las terminales.		
ORIGEN	Empresa		
DIRIGIDO A	IT		
PRIORIDAD	4		
REQUERIMIENTOS ASOCIADOS	Servicios levantados y programas adicionales necesarios		
ESPECIFICACION	Servicios levantados y programas adicionales necesarios		
PRECONDICIONES	Ninguna		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Si no influye con el funcionamiento de los demás servidores y tienen un funcionamiento estable.		

RF001= identificador del requerimiento no funcional

1 – 3 = Nivel de Prioridad Baja

4 – 6 = Nivel de Prioridad Media

7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos, actores y especificaciones que interfieren en el requerimiento no funcional 001.

Fuente: Maycol Flores

Tabla 11

Especificación de requerimiento no funcional N°2

La aplicación no funcionara con navegadores desactualizados.		ESTADO	Análisis
CREADO POR	Socios	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	NRF002		
TIPO DE REQUERIMIENTO	CRITICO	TIPO DE REQUERIMIENTO	NO FUNCIONAL
DATOS DE ENTRADA	Interacción con la interfaz de usuario		
DESCRIPCION	Aplicación ejecutada desde los navegadores actualizados		
DATOS DE SALIDA	Interfaz gráfica de usuario		
RESULTADOS ESPERADOS	La funcionalidad en los navegadores actualizados.		
ORIGEN	Empresa		
DIRIGIDO A	Usuarios del sistema		
PRIORIDAD	4		
REQUERIMIENTOS ASOCIADOS	Ninguna		
ESPECIFICACION	Ninguna		
PRECONDICIONES	Ninguna		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Funcionalidad con los navegadores actualizados.		

RF002= identificador del requerimiento no funcional

1 – 3 = Nivel de Prioridad Baja

4 – 6 = Nivel de Prioridad Media

7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos, actores y especificaciones que interfieren en el requerimiento no funcional 002.

Fuente: Maycol Flores

Tabla 12

Especificación de requerimiento no funcional N°3

El sistema no cumplirá con todos los procesos del módulo de contabilidad.		ESTADO	Análisis
CREADO POR	Socios	ACTUALIZAR POR	Socios
FECHA DE CREACION	18/11/2014	FECHA DE ACTUALIZACION	18/11/2014
IDENTIFICADOR	NRF003		
TIPO DE REQUERIMIENTO	NO CRITICO	TIPO DE REQUERIMIENTO	NO FUNCIONAL
DATOS DE ENTRADA	Ninguna		
DESCRIPCION	El sistema no cumplirá con todos los procesos del módulo de contabilidad.		
DATOS DE SALIDA	Ninguna		
RESULTADOS ESPERADOS	Cumplir únicamente con lo anteriormente planteado.		
ORIGEN	Analista de sistemas		
DIRIGIDO A	Usuarios del sistema		
PRIORIDAD	3		
REQUERIMIENTOS ASOCIADOS	SRI		
ESPECIFICACION	Ninguna		
PRECONDICIONES	Ninguna		
POSCONDICIONES	Ninguna		
CRITERIOS DE ACEPTACION	Cumplir únicamente con lo anteriormente planteado.		

RF003= identificador del requerimiento no funcional

1 – 3 = Nivel de Prioridad Baja

4 – 6 = Nivel de Prioridad Media

7 – 10= Nivel de Prioridad Alta

Notas: En esta tabla se detallan todos los procesos, actores y especificaciones que interfieren en el requerimiento no funcional 003.

Fuente: Maycol Flores

2.02 Mapeo de involucrados:

Este proyecto tiene como objetivo facilitar los procesos de factura electrónica que satisfaga las necesidades de consumo de los clientes y su entorno, atendiendo a las necesidades de los actores localizados en sus alrededores y protegiendo el recurso con apoyo de todos los actores involucrados en el manejo, los cuales se integran de forma ordenada para cumplir con sus respectivas obligaciones:

Contador: Es la persona que posee la mayor autoridad dentro del sistema.

Gerente: Es el dueño de la empresa que delega posiciones dentro del sistema.

Empleados: Los empleados son los que siguen las disposiciones del gerente.

Estudiante: Es el encargado de la realización del sistema.

Clientes: Son las empresas o personas que solicitan servicios a la empresa.

SRI: Servicio de rentas internas.



Figura 1: Mapeo de Involucrados. Esta figura muestra la distribución de los involucrados en el proceso de la Facturación. Estableciendo quienes se involucran de forma directa o indirecta, así como también indica cual es el Área en que interviene cada involucrado.

Fuente: Maycol Flores

2.03 Matriz de involucrados

Tabla 13

Matriz de involucrados

Actores involucrados	Intereses sobre el problema central	Problemas percibidos	Recursos mandatos y capacidades	Intereses sobre el proyecto	Conflictos potenciales
Gerente	Generar reportes de forma ágil.	Rutinas poco eficientes al momento de generar reportes.	Recursos técnico, tecnológico, humano.	Optimizar el proceso de facturación	La aplicación no funcione de acuerdo a los requerimientos.
Contador	Aumentar el control en el proceso de facturación	Uso elevado de tiempo al momento de realizar procesos manualmente	Recursos humanos y tecnológico	Verificar que no existan errores en las facturas.	Que el sistema no cubra todos los procesos de facturación.
Administrador	Accede de forma ágil a la información.	Dificultad para buscar la información.	Recursos humanos.	Acceder automáticamente a la información.	No poder usar de forma correcta el sistema
Empleados	Generar facturas fácilmente	Errores al momento de generar facturas manualmente.	Recursos humanos, materiales	Realizar Facturas fácilmente	Conflictos al momento de la validación con el SRI
Estudiante	Realizar el sistema de facturación para la empresa.	Dificultad para la realización del sistema.	Recursos técnicos, tecnológico.	Optimizar el proceso de facturación.	Que el sistema no cumpla los requisitos del Sri.

Notas: Matriz de Análisis de Involucrados. Se detalla lo más posible sobre la participación de los involucrados en el proyecto.

Fuente: Maycol Flores

Capítulo III: Problemas y Objetivos

3.01 Árbol de problemas

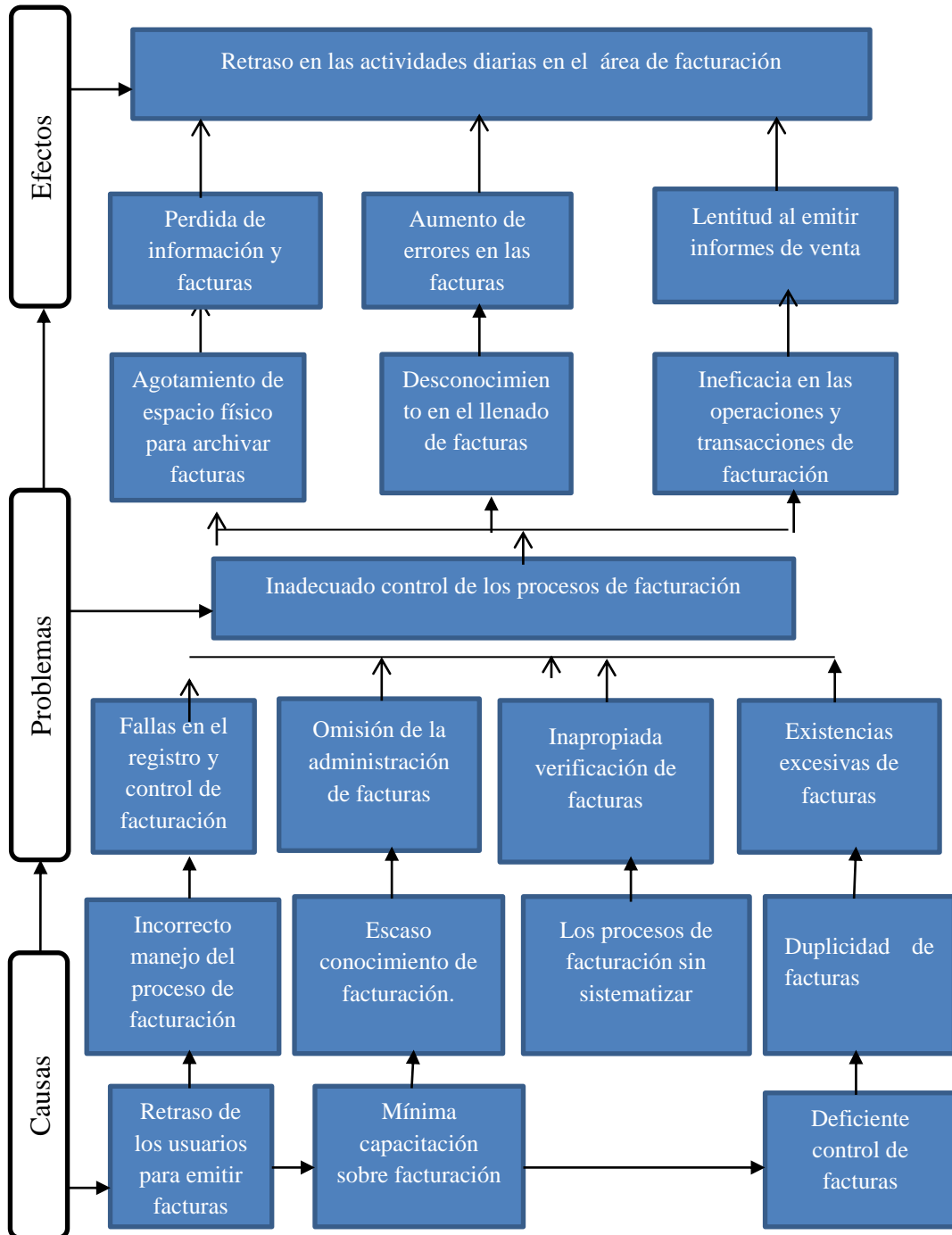


Figura 2: Gráfico del árbol de Problemas.- Esta figura explica los problemas que existen con respecto al problema central con sus causas y efectos.

Fuente: Maycol Flores

3.02 Árbol de objetivos

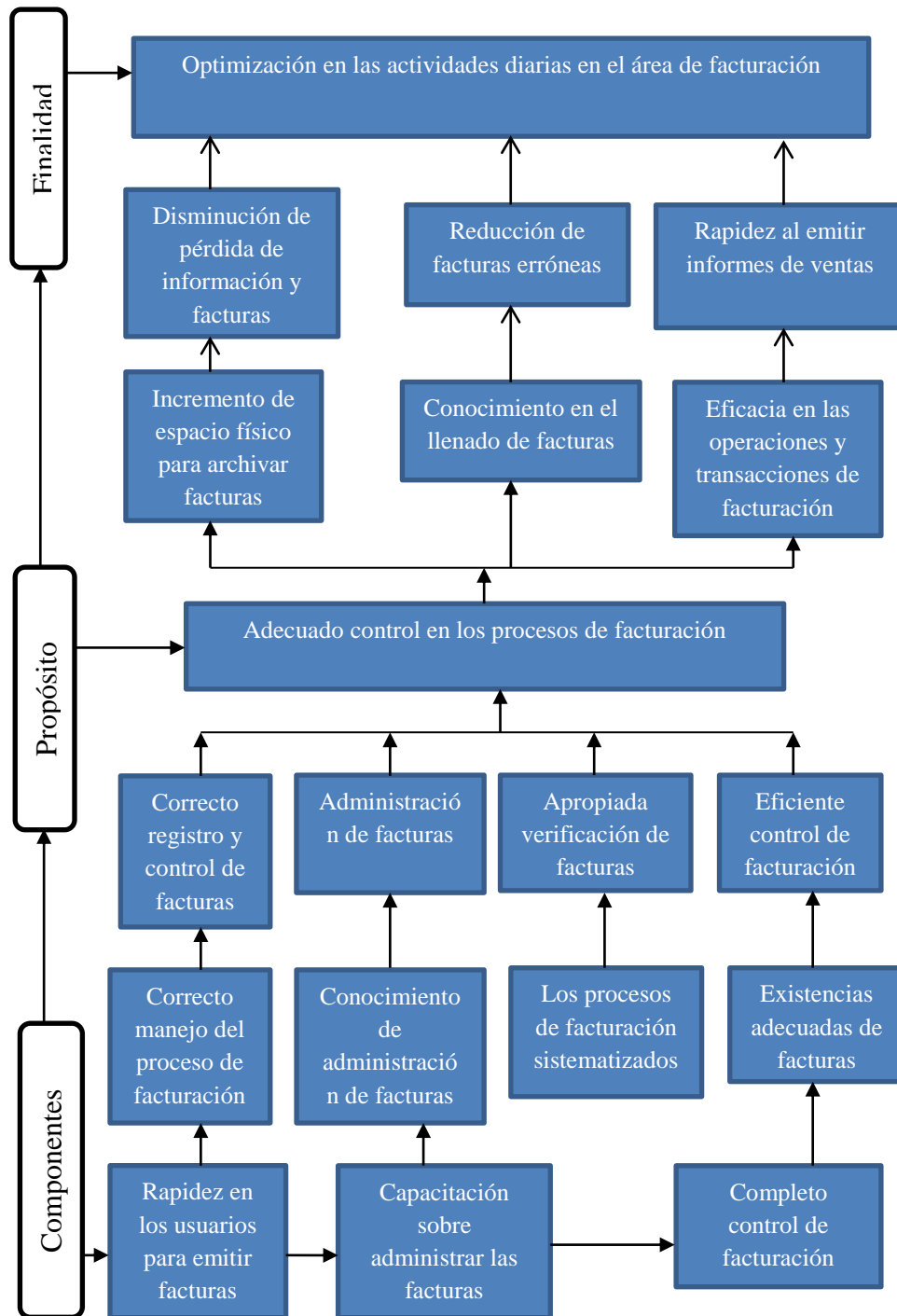


Figura 3: Gráfico del árbol de Objetivos. Esta figura explica los fines y medios para encarar el problema de desarrollo identificado.

Fuente: Maycol Flores

3.02.01 Análisis del Árbol de Problemas

En el presente árbol de problemas se pueden ver las situaciones empeoradas que se representa en el desarrollo del proyecto donde se especifica la situación actual de la empresa en los últimos años para dar solución a la misma. Para ello debemos tomar en cuenta que el problema principal que se va a solucionar con el software es tener un mayor control de facturación percibidos en la empresa.

3.02.02 Análisis del Árbol de Objetivos

Con el análisis del árbol de objetivos podemos ir solucionando cada uno de los errores que se están presentando actualmente en la empresa desde hace algunos años, reduciendo el porcentaje de errores en los procesos de facturación, motivando a los involucrados al uso de la tecnología y sus aplicaciones informáticas para obtener como resultado un buen control de inventarios en bodega y facturación de la empresa.

3.03 Diagramas de casos de uso

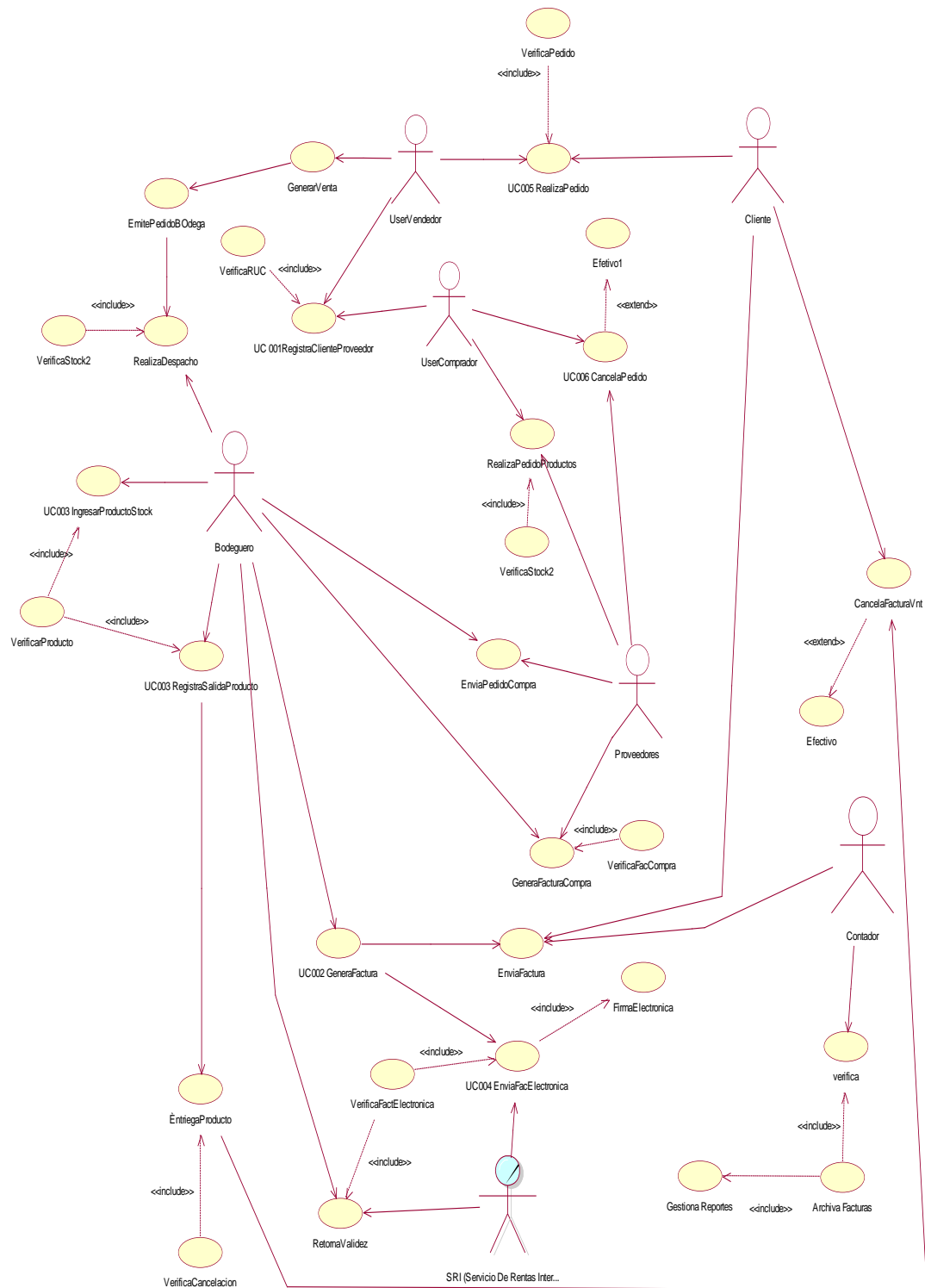


Figura 4: Diagrama de casos de uso general.

Fuente: Maycol Flores

Casos De Uso Vendedor

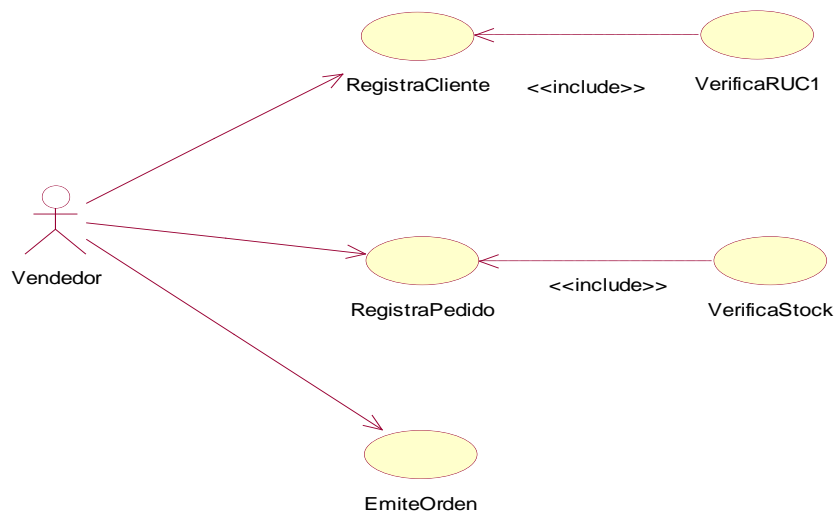


Figura 5: Diagrama de casos de uso del vendedor. El actual caso de uso se refiere al registro de clientes que realiza el vendedor, el cual debe verificar que el cliente posea RUC, también para el registro de pedidos, el cual deberá emitir la orden al bodeguero.

Fuente: Maycol Flores

Casos de Uso Comprador

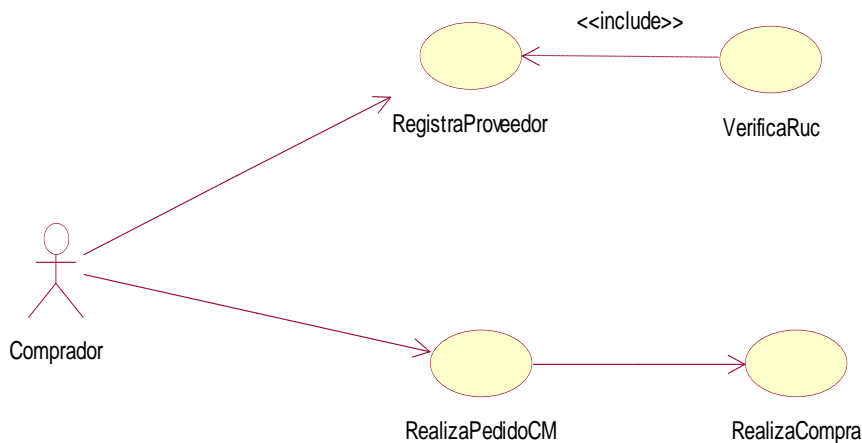


Figura 6: Diagrama de casos de uso del comprador. El actual caso de uso se refiere al registro de proveedores que realiza el comprador, el cual debe verificar que el proveedor posea RUC, también para la requisición de productos los cual la empresa requiera.

Fuente: Maycol Flores

Casos de Uso Bodeguero

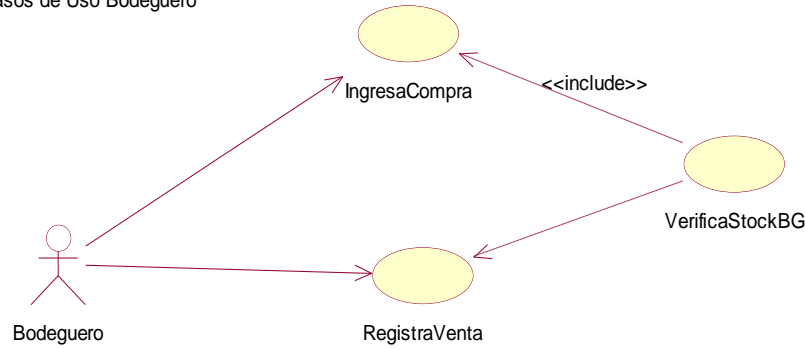


Figura 7: Diagrama de casos de uso del bodeguero. El actual caso de uso se refiere al registro de productos al momento de ingresar y a la salida de productos de la empresa

Fuente: Maycol Flores

Casos de Uso Devoluciones

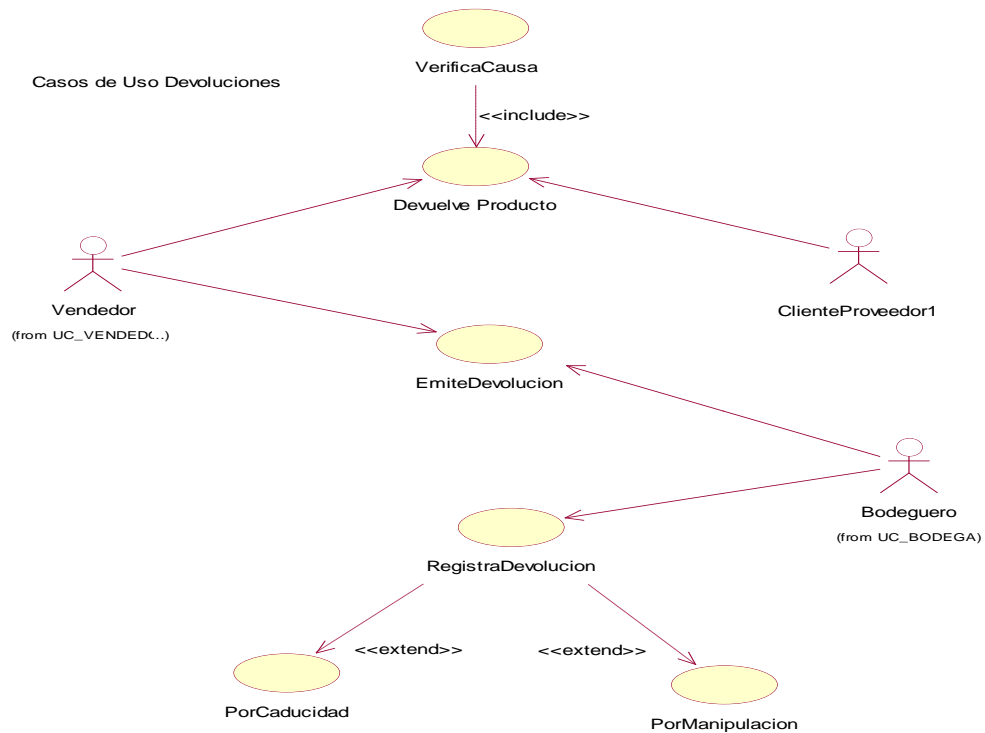


Figura 8: Diagrama de casos de uso de las devoluciones. El actual caso de uso nos indica que el cliente al verificar el producto y encuentra una anomalía retorna el producto al vendedor el cual emitirá una devolución al bodeguero el cual registrara la devolución delimitando la razón por la cual se devolvió el producto sea por caducidad o por alteración del mismo.

Fuente: Maycol Flores

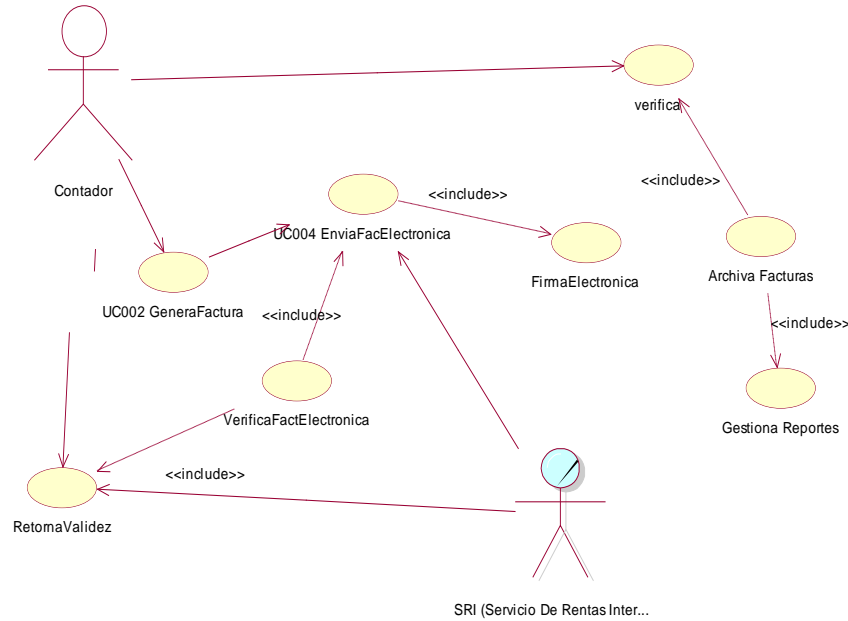


Figura 9: Diagrama de casos de uso del contador. El actual caso de uso nos indica que el contador genera la factura y envía la misma al SRI el cual verifica, firma y retorna validez al contador el cual verifica validez, archiva las facturas y gestiona los reportes.

Fuente: Maycol Flores

3.04 Casos de uso de realización

Caso UCR001.-

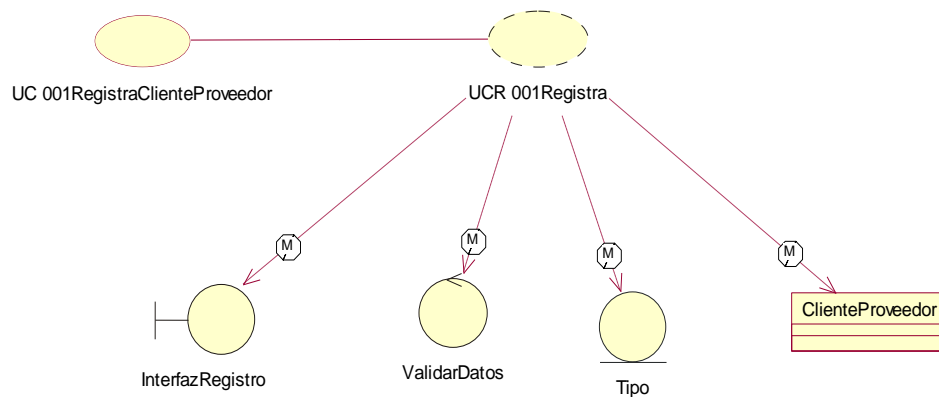


Figura 10: Diagrama de realización registrar cliente o proveedor

Fuente: Maycol Flores

Tabla 14

Especificación de casos de realización Registrar.

Nombre	Registra
Identificador	UCR001
Responsabilidades	Registrar clientes y/o proveedores solo con RUC
Tipo	Sistema
Referencias Casos de Uso	UC 001
Referencias Requisitos	RF001
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> Ingresar los datos informativos del cliente. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> El cliente debe estar registrado para poder generar una venta. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> Cliente Registrado con éxito. Proveedor Registrado con éxito. 	

Notas: En el cuadro de especificaciones podemos resaltar como el vendedor deberá registrar un cliente o el proveedor según sea el caso.

Fuente: Maycol Flores

Caso UCR002.-

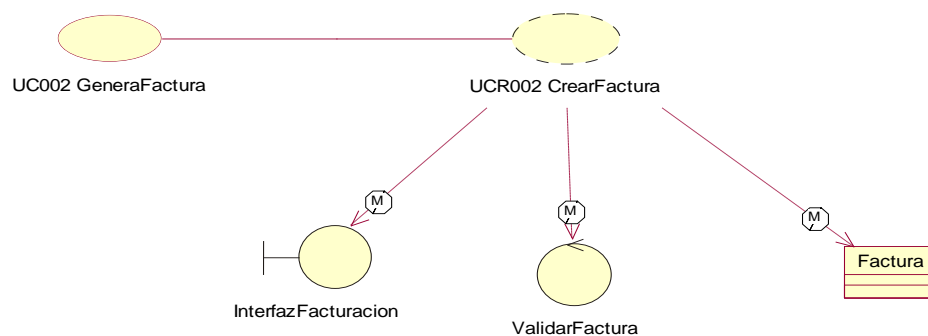


Figura 11: Diagrama de realización el cual muestra cómo se generara una factura.

Fuente: Maycol Flores

Tabla 15

Especificación de casos de realización Generar Factura.

Nombre	Crear Factura
Identificador	UCR002
Responsabilidades	Permitir el control de Facturas.
Tipo	Sistema
Referencias Casos de Uso	UC 002
Referencias Requisitos	RF002
PRECONDICIONES De Instancia	
<ul style="list-style-type: none"> • Ingresar los datos de venta como lo son los productos. 	
POSCONDICIONES De Instancia	
<ul style="list-style-type: none"> • El cliente debe estar registrado para poder generar una venta. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> • Factura emitida con éxito. 	

Notas: En el cuadro de especificaciones vemos como se podrá tener un control de facturas la cual se guardara en la base para mayor seguridad.

Fuente: Maycol Flores

Caso UCR003.-

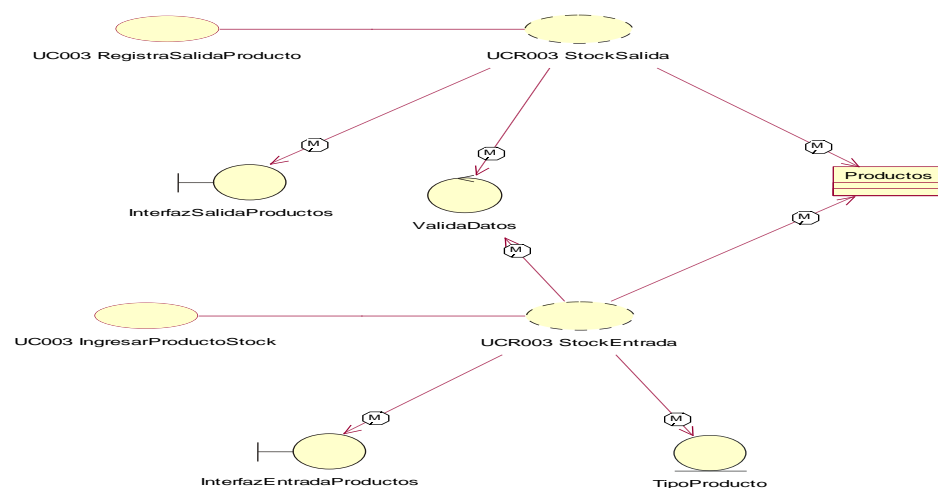


Figura 12: Diagrama de realización del control de salida y entrada de productos.

Fuente: Maycol Flores

Tabla 16

Especificación de casos de realización Stock Salida.

Nombre	Stock Salida
Identificador	UCR003
Responsabilidades	Control de Stock.
Tipo	Sistema
Referencias	UC 003
Referencias	RF003
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> Registra datos de los productos. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> El cliente debe estar registrado para poder generar una venta. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> Stock Actualizado. 	

Notas: En el cuadro de especificaciones resaltamos como se podrá registrar los productos cuando se generen ventas, lo cual nos permitirá tener un stock actualizado.

Fuente: Maycol Flores

Tabla 17

Especificación de casos de realización Stock Entrada.

Nombre	Stock Entrada
Identificador	UCR003
Responsabilidades	Control de Stock.
Tipo	Sistema
Referencias	UC 003
Referencias	RF003
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> Registra datos de los productos. Selecciona proveedor. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> El proveedor debe estar registrado para poder generar un ingreso. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> Producto Registrado. Stock actualizado. 	

Notas: En el cuadro de especificaciones resaltamos como se podrá registrar los productos cuando se ingrese un nuevo número de stock a la empresa, lo cual nos permitirá tener un stock actualizado.

Fuente: Maycol Flores

Caso UCR004.-

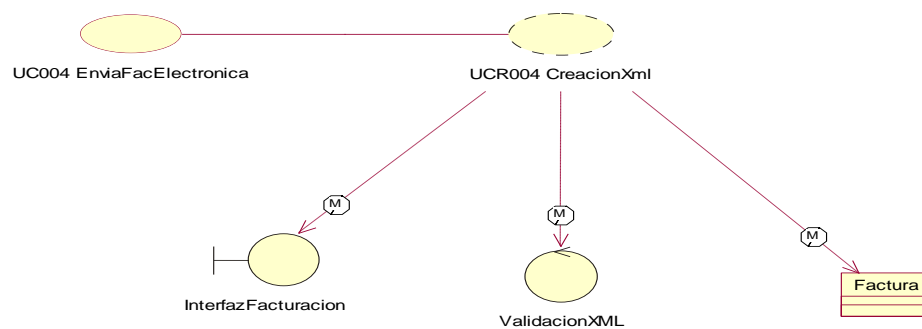


Figura 13: Diagrama de realización de la creación de XML.

Fuente: Maycol Flores

Tabla 18

Especificación de casos de realización Creación XML.

Nombre	Creación XML.
Identificador	UCR004
Responsabilidades	Generación Facturación Electrónica.
Tipo	Sistema
Referencias	UC 004
Referencias	RF004
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> Generará un archivo XML basado a las normas regidas por el SRI. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> Se debe generar una venta para crear el archivo XML de la factura. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> Factura enviada al SRI. Factura generada correctamente / Factura generada con errores. 	

Notas: En el cuadro de especificaciones damos a conocer como se generara la factura electrónica desde la factura normal la cual será enviada al SRI para su verificación.

Fuente: Maycol Flores

Caso UCR005.-

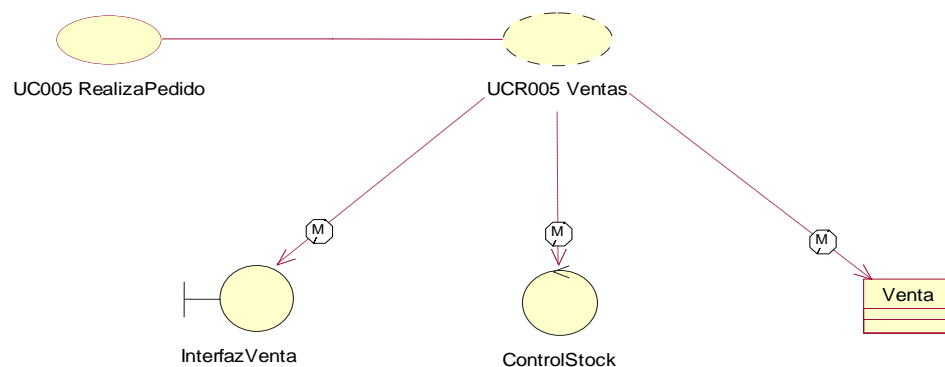


Figura 14: Diagrama de realización de la generación de venta.

Fuente: Maycol Flores

Tabla 19

Especificación de casos de realización Ventas.

Nombre	Ventas
Identificador	UCR005
Responsabilidades	Generación de ventas.
Tipo	Sistema
Referencias	UC 005
Referencias	RF005
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Ingresar los productos. • Ingresar Cliente. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Debe existir el stock necesario para la venta. • El cliente debe estar registrado. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> • Venta generada exitosamente. 	

Notas: En el cuadro de especificaciones vemos como se generara la venta y la orden de venta.

Fuente: Maycol Flores

Caso UCR006.-

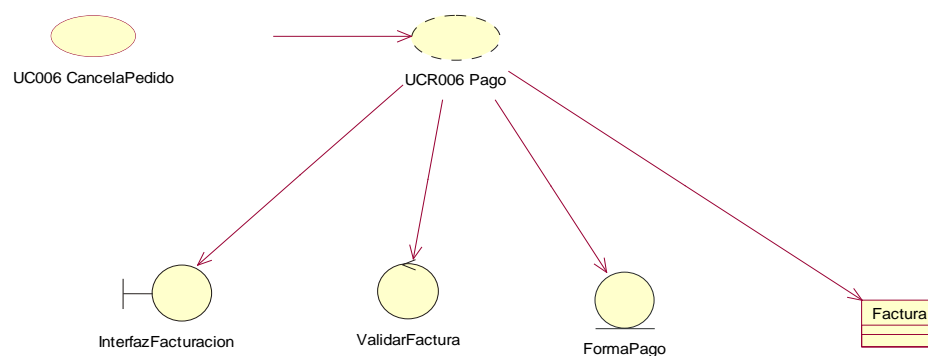


Figura 15: Diagrama de realización del tipo de pago.

Fuente: Maycol Flores

Tabla 20

Especificación de casos de realización Pago.

Nombre	Pago
Identificador	UCR006
Responsabilidades	Seleccionar el tipo de cancelación.
Tipo	Sistema
Referencias	UC 006
Referencias	RF006
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Seleccionar tipo de pago. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Debe existir el stock necesario para la venta. • El cliente debe estar registrado. 	
SALIDAS PANTALLA	

- Venta generada exitosamente.

Notas: En el cuadro de especificaciones vemos como en la factura se deberá seleccionar el tipo de pago que el cliente debe realizar para poder generar la venta.

Fuente: Maycol Flores

Caso UCR007.-

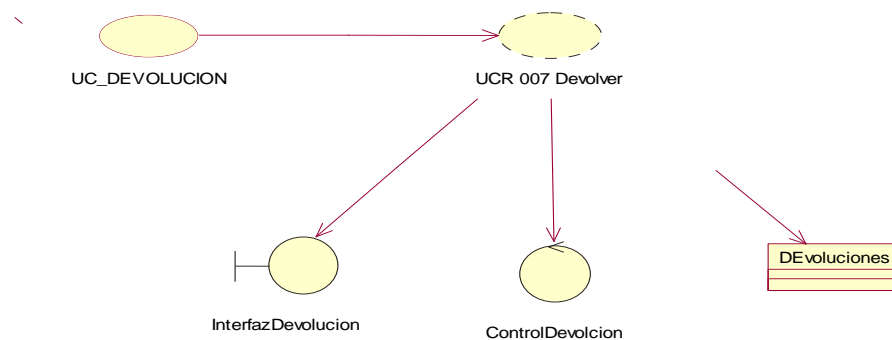


Figura 16: Diagrama de realización devolución.

Fuente: Maycol Flores

Tabla 21

Especificación de casos de realización Devolver.

Nombre	Devolver
Identificador	UCR007
Responsabilidades	Realizar devolución de productos.
Tipo	Sistema
Referencias	UC 007
Referencias	RF007
PRECONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Seleccionar razón de la devolución. 	
POSCONDICIONES	
De Instancia	
<ul style="list-style-type: none"> • Debe de haber existido la venta, para registrar la devolución. 	
SALIDAS PANTALLA	
<ul style="list-style-type: none"> • Devolución Registrada. 	

Notas: En el cuadro de especificaciones vemos como se deberá registrar una devolución, se deberá también seleccionar la razón de la devolución.

Fuente: Maycol Flores

3.05 Diagrama de secuencias del sistema

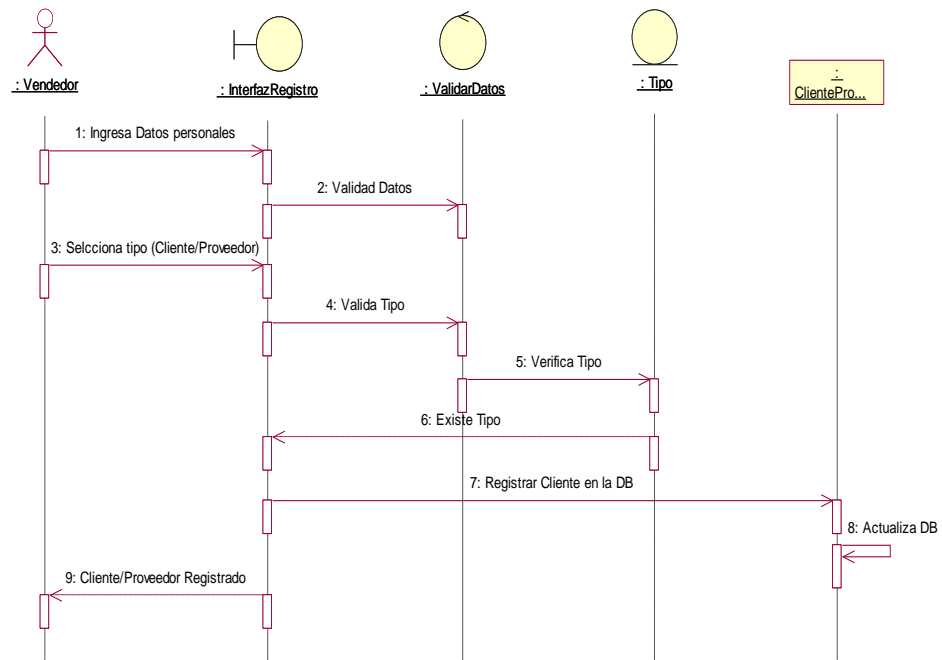


Figura 17: En este diagrama de secuencia vemos como el vendedor registra a un nuevo usuario para guardar sus datos personales en la base de datos.

Fuente: Maycol Flores

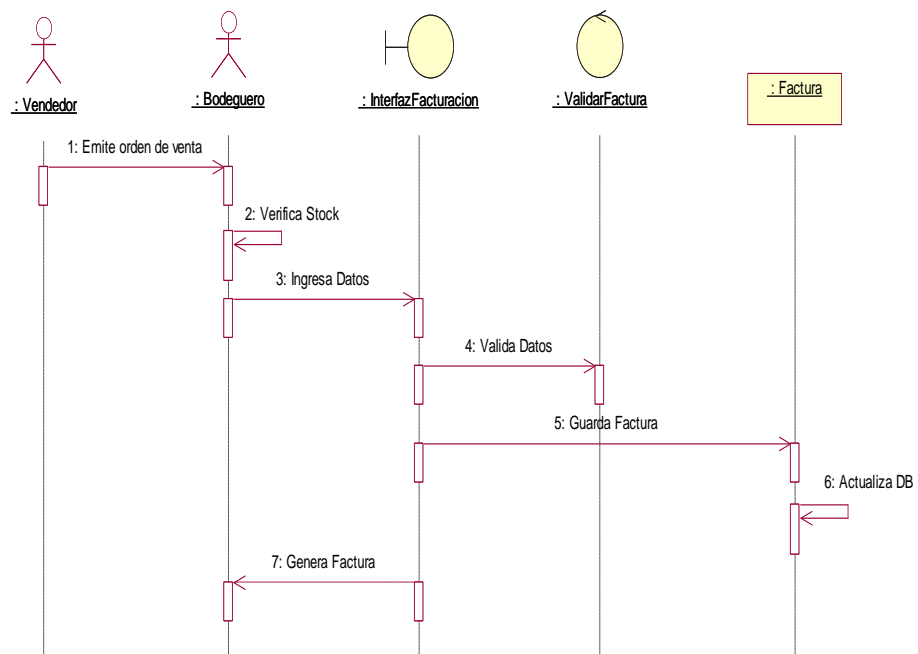


Figura 18: En este diagrama de secuencia vemos como el vendedor emite la orden de venta, el bodeguero verifica el stock y genera la factura y la registra en la base de datos.

Fuente: Maycol Flores

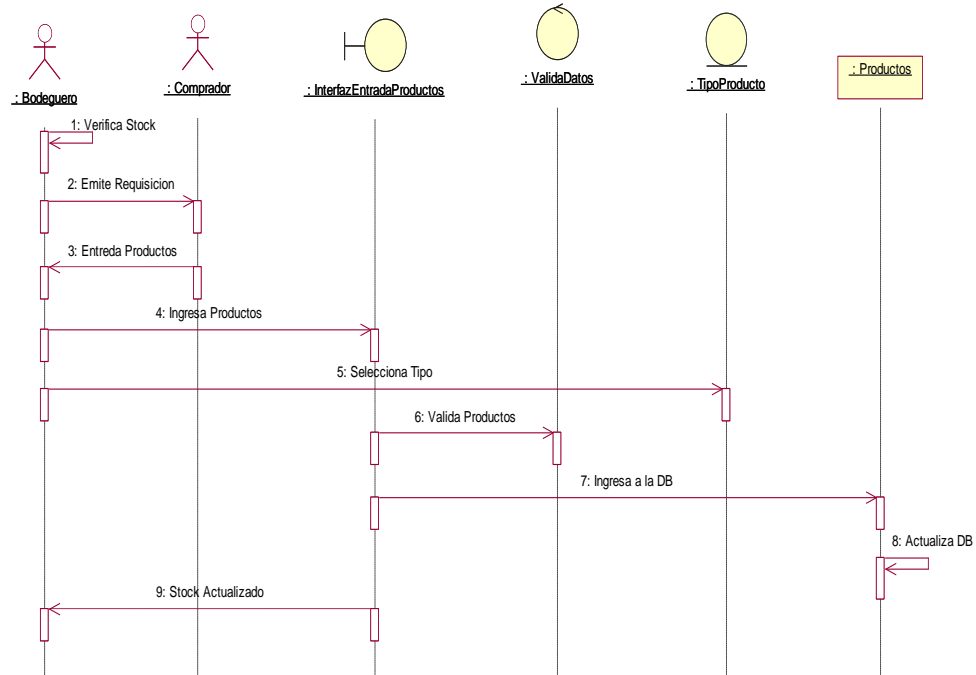


Figura 19: En este diagrama de secuencia vemos como el bodeguero requiere más productos y el comprador compra a los proveedores, el bodeguero procede al ingreso de productos para actualizar el stock.

Fuente: Maycol Flores

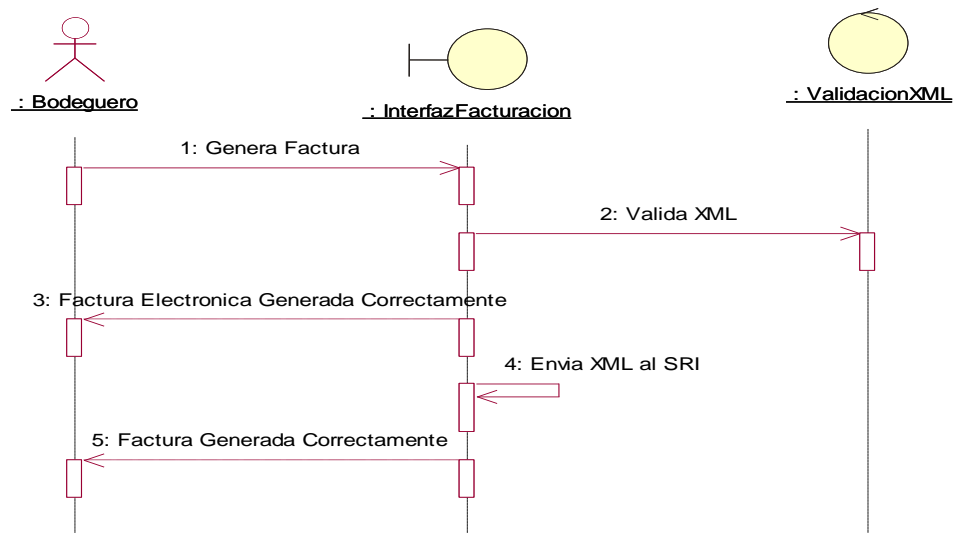


Figura 20: En este diagrama de secuencia vemos como el bodeguero genera el archivo XML desde la misma interfaz de facturación y es enviada al SRI para la verificación de la misma

Fuente: Maycol Flores

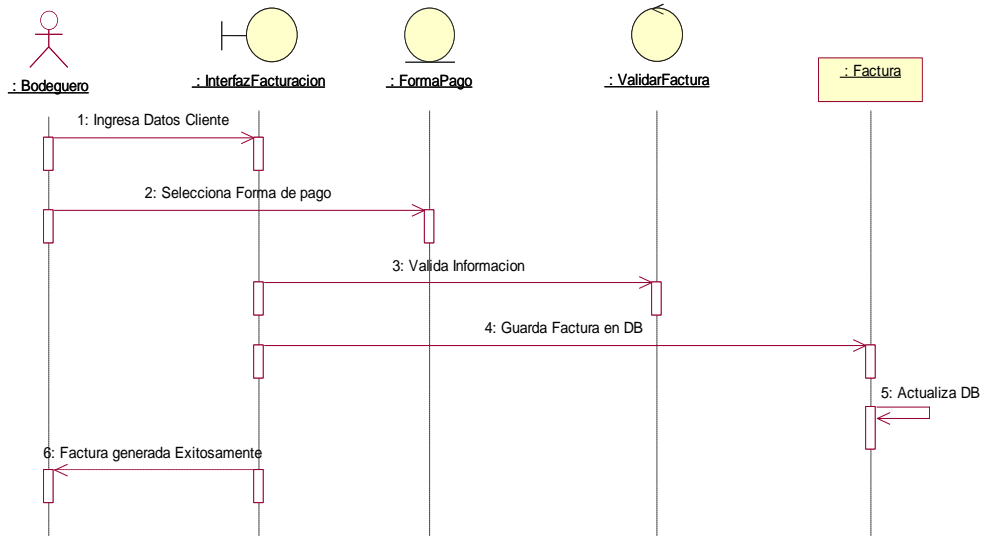


Figura 21: En este diagrama de secuencia vemos como el bodeguero escoge la forma de pago que desea realizar el cliente.

Fuente: Maycol Flores

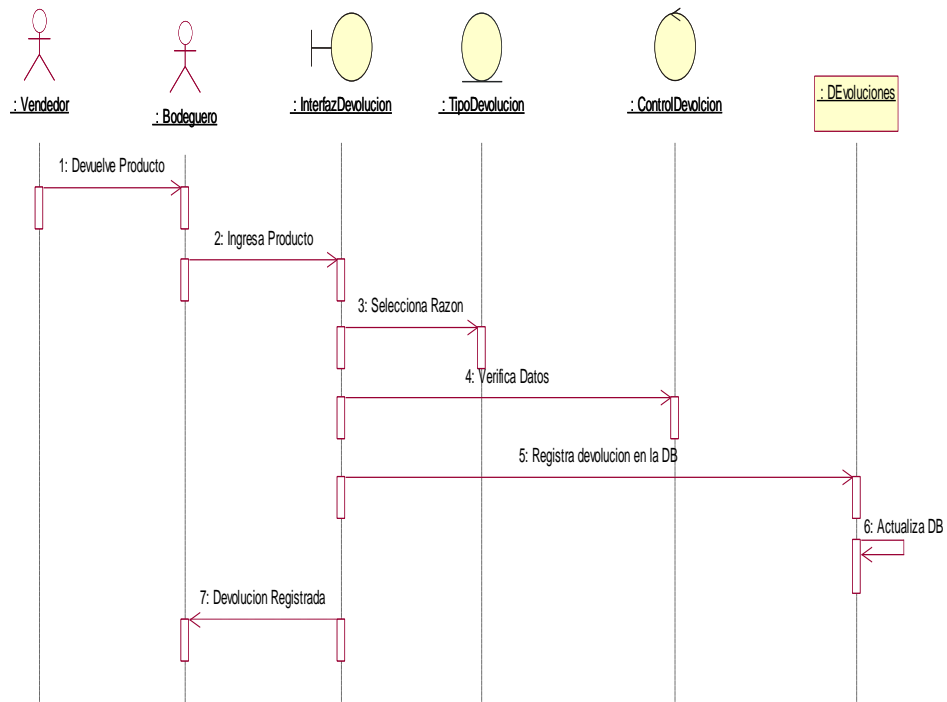


Figura 22: En este diagrama de secuencia vemos como el vendedor devuelve productos y el bodeguero procede a registrar la devolución.

Fuente: Maycol Flores

3.06 Especificación de casos de uso

Tabla 22

Especificación de casos de uso Registra Cliente Proveedor

Caso de Uso		Registra Cliente Proveedor	
Identificador		UC 001	
CURSO TÍPICO DE EVENTOS			
Usuario		Sistema	
1.-El vendedor solamente registrara a los clientes y proveedores que posean RUC.		1.- El sistema mostrara todos los campos obligatorios que se deben llenar.	
CURSOS ALTERNATIVOS			
1.- Si los usuarios del sistema no están registrados no podrán registrar a ningún cliente, ni proveedor.			

Nota: El anterior cuadro muestra cómo se debe registrar de manera correcta a un cliente y/o proveedor.

Fuente: Maycol Flores

Tabla 23

Especificación de casos de uso Genera Factura.

Caso de Uso		Genera Factura	
Identificador		UC 002	
CURSO TÍPICO DE EVENTOS			
Usuario		Sistema	
1.-El bodeguero recibirá la orden de venta y procederá a registrar los datos del cliente y de los productos que requiere.		1.- El sistema deberá contar con el módulo de editar las facturas por si existen errores.	
CURSOS ALTERNATIVOS			
1.- Si los usuarios del sistema no están registrados no podrá generar las facturas			

Notas: El anterior cuadro muestra cómo se debe generar una factura para lo cual la persona encarga de la facturación deberá estar registrada en el sistema.

Fuente: Maycol Flores

Tabla 24

Especificación de casos de uso RegistraSalidaProducto – RegistraEntradaProducto

RegistraSalidaProducto – RegistraEntradaProducto	
Identificador	UC 003
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
1.-El bodeguero en el caso de generar una venta deberá registrar la cantidad de productos que se vendieron.	1.- El sistema deberá contar con el módulo que permita actualizar el stock de productos en el caso de ingresar nueva mercancía.
2.- En el caso de ingresar productos de igual manera deberá registrar los productos para así mantener actualizada la base de datos.	
CURSOS ALTERNATIVOS	
1.- Si los usuarios del sistema no están registrados no podrá ingresar productos al sistema.	

Notas: El anterior cuadro muestra cómo se debe registrar los productos tanto como de salida como para el ingreso de productos.

Fuente: Maycol Flores

Tabla 25

Especificación de casos de uso Envía Factura Electrónica.

Caso de Uso	Envía Factura Electrónica
Identificador	UC 004
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
1.-El bodeguero al momento de generar la factura deberá generar el XML para que sea enviado al SRI.	1.- El sistema deberá contar con un botón el cual permita generar el XML y permita cargarlo para enviarlo al SRI.
2.- El SRI enviara un mensaje de confirmación si la factura consta sin errores.	
CURSOS ALTERNATIVOS	
1.- Si los usuarios del sistema no están registrados no podrá generar facturas.	

Nota: El anterior cuadro muestra cómo se deberá generar el XML y como se podrá saber si está correctamente generado.

Fuente: Maycol Flores

Tabla 26

Especificación de casos de uso Realiza Pedido

Caso de Uso	Realiza Pedido
Identificador	UC 005
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
1.-El vendedor deberá registrar todos los productos que el cliente desee.	1.- El sistema deberá contar con el formulario de venta.
2.- el vendedor emitirá una orden de despacho para el bodeguero	
CURSOS ALTERNATIVOS	
1.- Si los usuarios del sistema no están registrados no podrá generar ventas.	

Notas: El anterior cuadro muestra cómo se deberá generar una venta de la manera más correcta.

Fuente: Maycol Flores

Tabla 27

Especificación de casos de uso Devolver

Caso de Uso	Devolución
Identificador	UC007
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
1.-El vendedor deberá entregar los productos al bodeguero el cual se encargara de registrar la devolución.	1.- El sistema deberá contar con el formulario de devolución de productos.
2.- El bodeguero deberá seleccionar la razón por la cual se ejecutó esta acción.	
CURSOS ALTERNATIVOS	

1.- Si los usuarios del sistema no están registrados no podrá generar devoluciones.

Notas: El anterior cuadro muestra cómo se deberá generar devolución la cual como primer punto debió existir una venta.

Fuente: Maycol Flores

Capítulo IV: Análisis de Alternativas

4.01 Matriz de Análisis de Alternativas

Tabla 28

Matriz de Análisis de Alternativas

MATRIZ DE ANÁLISIS DE ALTERNATIVAS							
OBJETIVO	Impacto sobre el Propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categorías
Analizar el área designada a realizar el sistema.	4	3	4	3	2	16	Alta
Asignar el personal para cada una de las áreas del uso del sistema	3	2	4	4	2	15	Media Alta
Analizar el mejoramiento de la empresa una vez aplicado el sistema	3	3	2	4	4	16	Alta
Concluir con éxito el sistema	2	3	4	2	4	15	Media Alta
Permisos asignados a usuarios adecuadamente.	4	4	3	3	2	16	Alta
Acceso a la información solamente de personal autorizado	4	2	3	4	3	16	Alta
TOTAL	20	17	20	20	19	94	

Notas: La siguiente tabla representa la matriz de alternativas con las variables de factibilidad en base a lo político, social, financiero, técnico y en base al impacto sobre el propósito.

Fuente: Maycol Flores

4.01.01 Análisis Matriz de Alternativas

(León, 2007) “Una vez concluido el análisis de objetivos, ya podemos tener una idea aproximada de los retos que enfrentará el proyecto, asimismo ya se tiene claro que aspectos deben trabajarse para lograr un proyecto con impacto exitoso”.

Al analizar el área designada para el evento se clasifica en categoría alta y media alta ya que las actividades que se realizan en el negocio son muy importantes ya que cada usuario debe tener un conocimiento del área la cual estará designada a su cargo para su mayor funcionalidad.

La toma de decisiones correctas en el momento del evento, se realiza por cualquier altercado que suceda en el mismo la cual se clasifica en categoría alta en donde al culminar el evento con éxito también se clasifica en categoría alta, ya que si se llega a cumplir será de gran beneficio y subirá el prestigio de la empresa lo que ayudara en futuros eventos.

La puntuación tomada en cuenta para la matriz de alternativas son las siguientes:

Malo: 1

Regular: 2

Bueno: 3

Muy Bueno: 4

Excelente: 5

Nivel de factibilidad de los objetivos

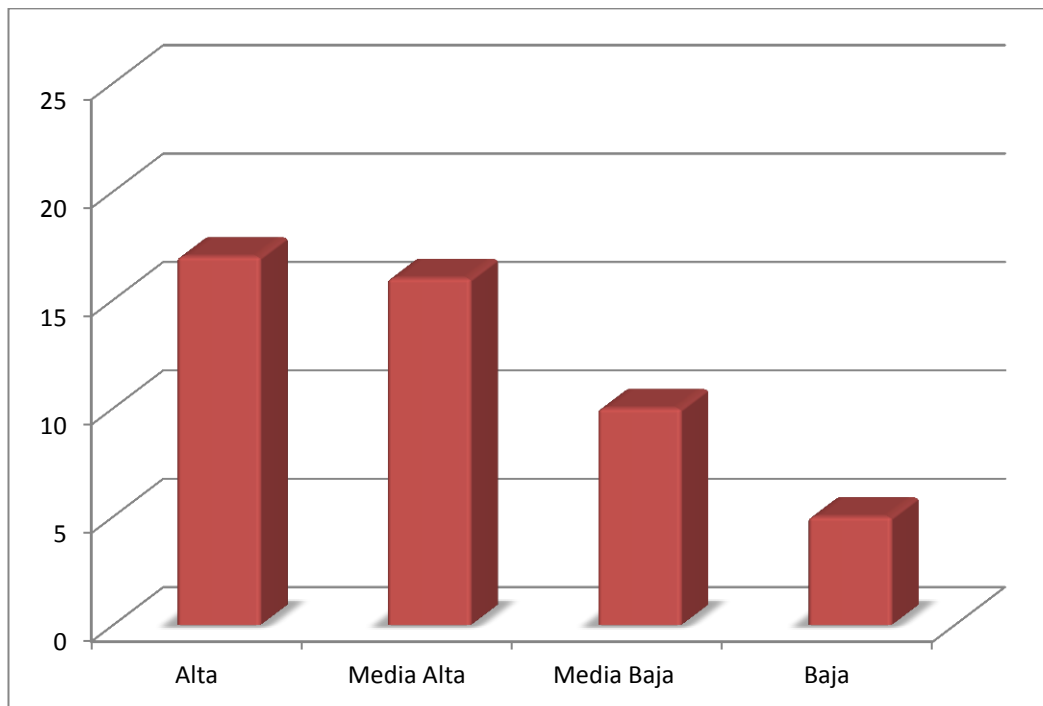


Figura 23: En esta figura muestra el nivel de factibilidad de los objetivos.

Fuente: Maycol Flores

Como podemos observar en la imagen del nivel de factibilidad de los objetivos, tenemos un gráfico estadístico donde tenemos niveles altos, medio altos medios, bajos, y bajos. Ya que el nivel alto tiene un valor de 16, el nivel media alta tiene un valor de 15. Y va una escala de 0 a 25.

4.02 Matriz de Impactos de Objetivos

Tabla 29

Matriz de Impactos de Objetivos

	Factibilidad de Lograse	Impacto En Género	Impacto Ambiental	Relevancia	Sostenibilidad	Total
O B J E T I V O S	Saber si es aceptable y conveniente para la empresa VIASCALA .	En la empresa se cuenta con personal masculino y femenino.	Contribuye a proteger el entorno físico.	Los beneficios son altos para la empresa.	Fortalece a la empresa.	22-32 BAJA
	Saber si es posible producir ganancias con la ayuda del software	Implantar la participación del personal en la sistematización de procesos.	Favorece el uso de los recursos.	Responde a las expectativas planteadas .	Fortalece la participación de los usuarios y clientes.	33-44 MEDIA BAJA
	Saber si es posible reducir fraudes con la ayuda del software		Favorece al entorno ambiental.	Beneficia a grupos de mayor carencia y vulnerabilidad.	Se puede conseguir financiamiento a futuro	45-66 MEDIA ALTA
	Reducción de errores y mayor precisión en los procesos		Mejora el entorno social.	Mejora el entorno laboral y cultural de la empresa.	Los clientes estarán en	57-90 ALTA
	Mejoramiento de los servicios a clientes o usuarios.			Reduce los fallos en el manejo de información.	condiciones y posibilidades de aportar medios.	
Adecuado control en el área de facturación y seguridad en su información	18 puntos	13 puntos	17 puntos	15 puntos	20 puntos	83 puntos

4.03 Estándares para el Diseño de Clases

(Bonaparte, 2012) “UML respaldado por el OMG (Object Management Group), es un lenguaje de modelado de sistemas de software. Diseñado como una herramienta gráfica donde se puede construir, especificar, visualizar y documentar sistemas. Permite representar el modelo de un escenario, donde se describen las entidades intervinientes y sus relaciones. También podemos describir cada entidad, especificar las propiedades y el comportamientos de las mismas.

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Al ser UML un lenguaje, existen reglas para combinar dichos elementos. En conjunto, los diagramas UML brindan diversas perspectivas de un sistema, por ende el modelo. Ahora bien, el modelo UML describe lo que hará el sistema y no como será implementado”.

4.03.01 Bloques de Construcción

Los bloques básicos de construcción de UML son tres, los elementos, las relaciones y los diagramas.

(Arregui, 2004) “**Los elementos:** Son abstracciones que actúan como unidades básicas de construcción. Hay cuatro tipos, los estructurales, los de comportamiento, los de agrupación y los de notación.

En cuanto a los elementos estructurales son las partes estáticas de los modelos y representan aspectos conceptuales o materiales. Los elementos de comportamiento son las partes dinámicas de los modelos y representan comportamientos en el tiempo y en el espacio. Los elementos de agrupación son las partes organizativas de UML, establecen las divisiones en que se puede fraccionar un modelo. Sólo hay un

elemento de agrupación, el paquete, que se emplea para organizar otros elementos en grupos. Los elementos de notación son las partes explicativas de UML, comentarios que pueden describir textualmente cualquier aspecto de un modelo. Sólo hay un elemento de notación principal, la nota.



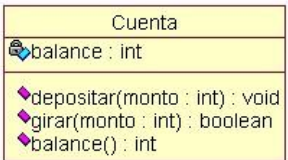




Las relaciones: Son abstracciones que actúan como unión entre los distintos elementos. Hay cuatro tipos, la dependencia, la asociación, la generalización y la realización.


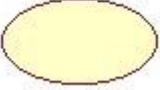
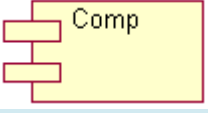
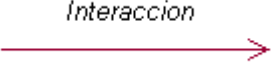
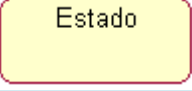
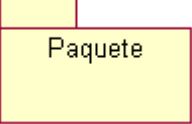
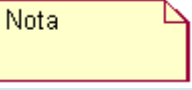

Los diagramas: Son la disposición de un conjunto de elementos, que representan el sistema modelado desde diferentes perspectivas. UML tiene nueve diagramas fundamentales, agrupados en dos grandes grupos, uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico. Los diagramas estáticos son: el de clases, de objetos, de componentes y de despliegue. Los diagramas de comportamiento son: el de Casos de Uso, de secuencia, de colaboración, de estados y de actividades”.

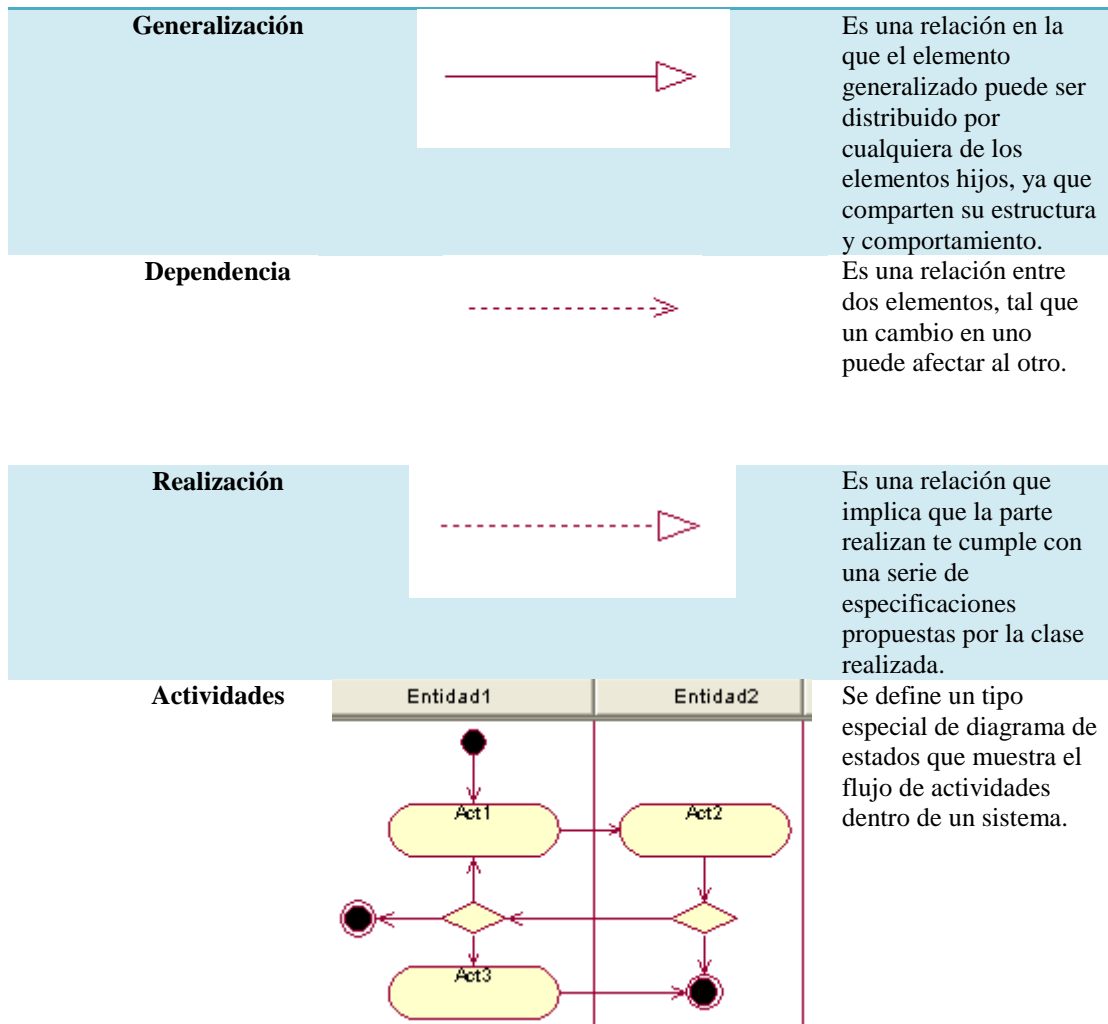
4.03.02 Elementos

Tabla 30

Elementos Estructurales en UML

E S T Á N D A R E S D E U N A C L A S E	Clase		Se define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase. Las clases implementan una o más interfaces.
	Interfaz		Permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.
	Atributos		Los atributos o características de una Clase pueden ser de tres tipos, public, private, protected
	Atributo Public		Indica que el atributo será visible tanto dentro como fuera de la clase
	Atributo privado		Indica que el atributo solo será accesible desde dentro de la clase solo con sus métodos
	Atributo Protected		Indica que el atributo no será accesible desde la fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven.
	Nodo		Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con capacidad de procesar.

Colaboración	 Colaboracion	Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor
Casos de Uso	 Caso de Uso	Es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular.
Diagrama de Componentes	 Comp	Se define una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto.
Interacción	 Interaccion	Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
Estado	 Estado	Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.
Paquete	 Paquete	Se define para organizar otros elementos en grupos.
Nota	 Nota	Se define partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo.
Asociación	 * 0..1	Se define una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.



Nota: La siguiente tabla representa los elementos estructurales en UML.

Fuente: Arregui M

4.04 Diagrama de Clases

El diagrama de clases está compuesto por los principales componentes que son métodos y atributos donde adicionalmente las clases poseen un conjunto de datos primitivos que incluyen:

- Un conjunto completo de tipos de enteros con signo con 1, 2, 4 y 8 bytes de precisión (tinyint, smallint, int y bigint).
- Un tipo de datos bit con valores 0 o 1 (bit).
- Tipo decimal con 38 dígitos de precisión (decimal, numeric).

- Tipos de moneda con precisión de 1/1000 de la unidad monetaria (money, smallmoney).
- Tipos de fecha y hora con una precisión de hasta 3.33 milisegundos (datetime, small-datetime).
- Tipos en coma flotante de precisión sencilla y doble (real, float).
- Tipos de cadenas de caracteres de tamaño fijo y variable de hasta 230-1 caracteres Unicode y no Unicode (char/nchar, varchar/nvarchar, text/ntext).
- Cadenas de bytes de tamaño fijo y variable de hasta 231-1 bytes (binary, varbinary, image).
- Un tipo cursor que permite referencias a un objeto cursor (cursor)
- Tipos de datos de identificadores únicos globales y para la base de datos.”
(Silberschatz, Korth, & Sudarshan, S. (Instituto Indio de Tecnología, 2002)

Véase en el Anexo **A.01**

4.05 Modelo Lógico – Físico

A partir del esquema conceptual y teniendo en cuenta los requisitos de proceso y de entorno, se elabora un esquema lógico en la plataforma Case Rational Rose (Tichelaar & Demeyer, 1999) este diagrama esta generado para que a partir de este diagrama se pueda generar el diagrama físico y posteriormente la Base de Datos.

Véase en el Anexo **A.02** y **A.03**

4.06 Diagrama de Componentes

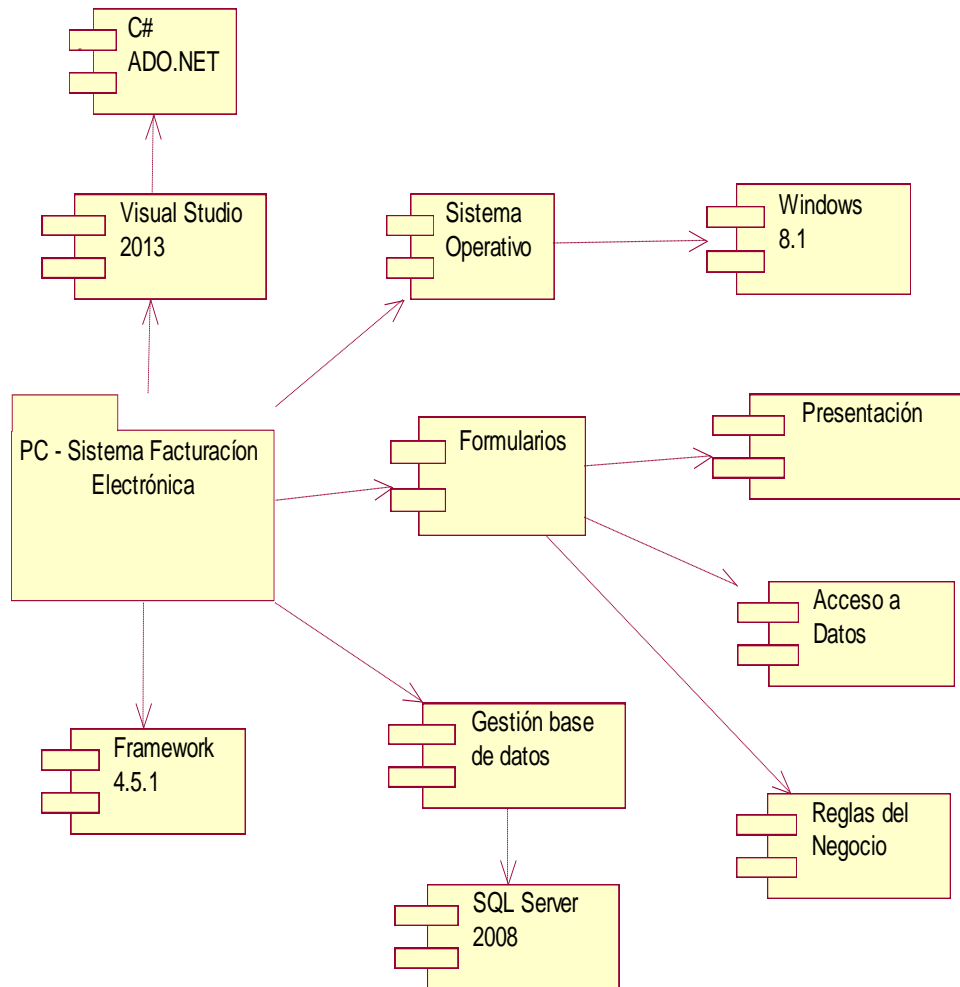


Figura 24: Diagrama de Componentes. Esta figura muestra los principales componentes

Fuente: Maycol Flores

4.07. Diagramas de Estrategias

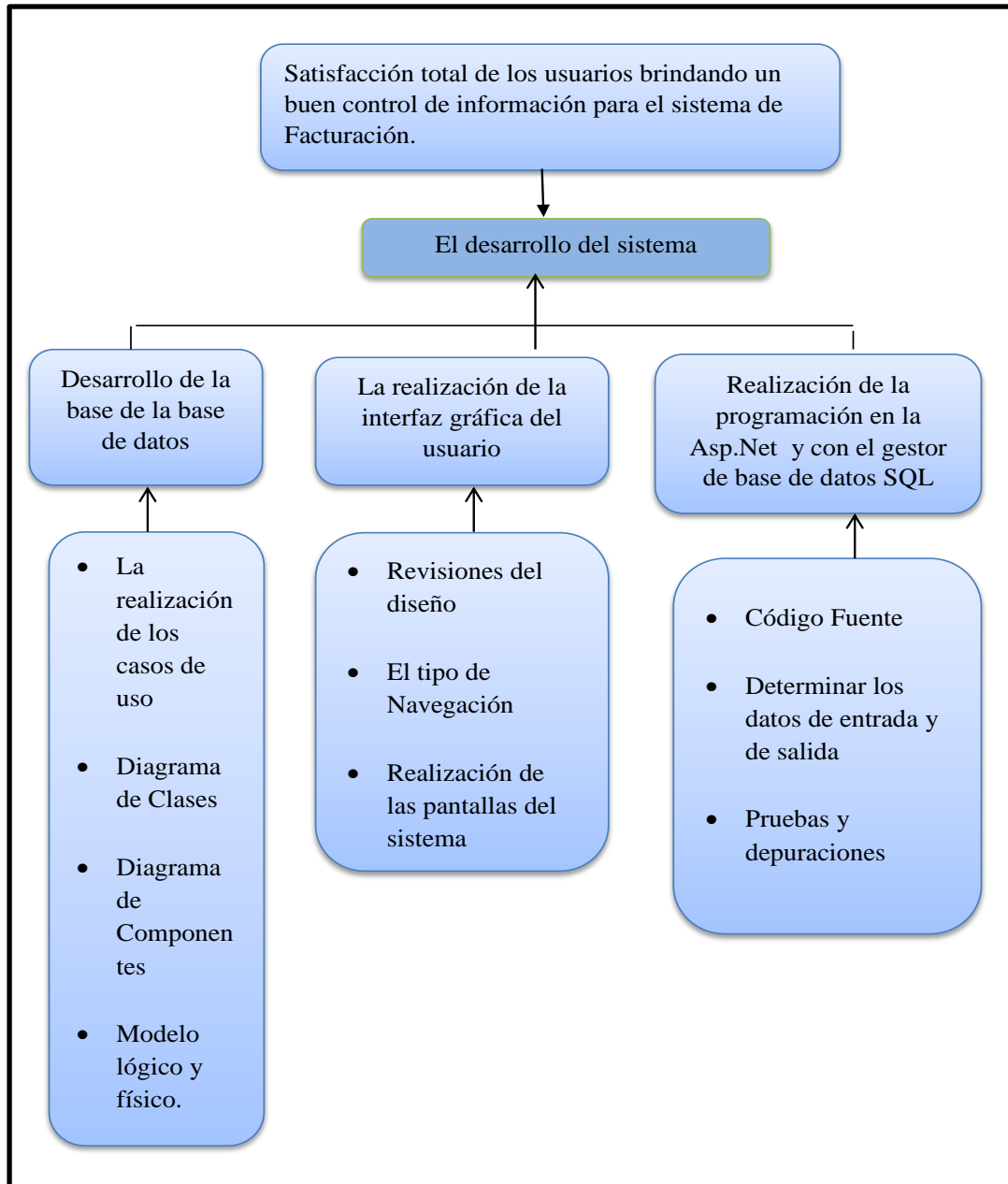


Figura 25: Diagrama de Estrategias.- Esta figura muestra los principales componentes y actividades para alcanzar la finalidad del sistema de Facturación de la empresa Viascala S.A.

Fuente: Maycol Flores

4.08 Matriz de Marco Lógico

El marco lógico ayuda en el proceso de conceptualización, diseño, ejecución y evaluación de proyectos. La siguiente tabla muestra los parámetros iniciales del proyecto, las continuas actividades y procesos que se ejecutaran a fin de lograr un objetivo claro.

Tabla 31

Matriz de Marco Lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
FIN DEL PROYECTO Manejo de la seguridad del sistema de una manera ágil, confidencialidad e integridad.	<ul style="list-style-type: none"> ●Inexistencia de información redundante. ●Pérdidas de información reducidas. ●Manejo de información confiable y segura. 	<ul style="list-style-type: none"> ●Aprobación del sistema por parte del usuario final 	<ul style="list-style-type: none"> ●Ingreso de los servicios de la empresa de acuerdo a las especificaciones del Gerente.
PROPÓSITO DEL PROYECTO Mejorar los procesos que se manejan actualmente de información de Facturación	<ul style="list-style-type: none"> ●Tiempo de respuesta reducido considerablemente. 	<ul style="list-style-type: none"> ● El sistema visualizara reportes conforme a las necesidades del usuario. 	<ul style="list-style-type: none"> ●Apoyo del personal para levantar los servicios que brinda la empresa.
COMPONENTES DEL PROYECTO 1.- Proceso de Facturación 2.- Capacitación a personal encargado de Facturación	<ul style="list-style-type: none"> ●Creación y autenticación de usuarios para el ingreso al sistema. ●Capacitación de los usuarios. 	<ul style="list-style-type: none"> ●Manual técnico, instalación y de usuario. ●Tomar sugerencias de los usuarios. 	<ul style="list-style-type: none"> ●Usuario/contraseña, creado bajo las especificaciones del Gerente. ●Soluciones a las inquietudes y problemas por parte de los usuarios del sistema.
ACTIVIDADES DEL PROYECTO 1.1 Ingreso al sistema por roles y autenticación de usuarios. 1.2 Interfaz amigable e intuitiva	<ul style="list-style-type: none"> ●Eficiencia y eficacia de soporte. ●Acceso confiable al sistema de información. 	<ul style="list-style-type: none"> ●Planificación de un cronograma con fechas para entrega de documentos y capacitaciones. 	<ul style="list-style-type: none"> ●La sesión caducará para mayor seguridad de la información.

4.09 Vistas Arquitectónicas

4.09.01 Vista Lógica

Se muestra un subconjunto significativo del modelo de diseño, es decir algunas clases, subsistemas y paquetes, y las realizaciones de caso de uso.

Véase en el Anexo A.04

4.09.02 Vista Física

Esta vista muestra como el sistema estará desplegada dentro de la empresa, los servicios y los procesos como se ejecutaran desde el usuario hasta la base de datos.

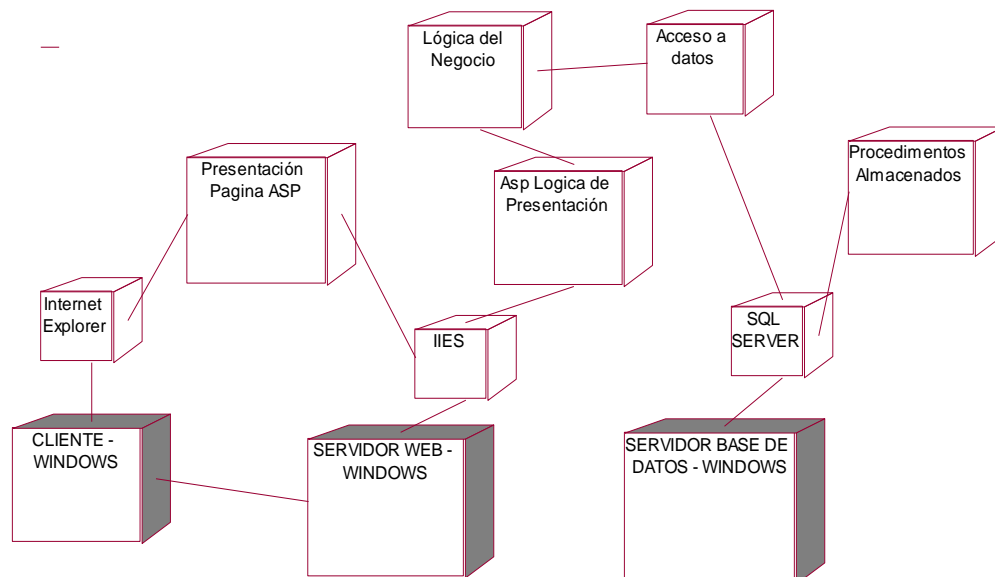


Figura 26: Vista Física.- Esta figura muestra la vista física del sistema de Facturación de la empresa VIASCALA S.A.

Fuente: Maycol Flores

4.09.03 Vista Desarrollo

Esta vista muestra como el sistema está separada por módulos o paquetes en el ambiente de desarrollo.

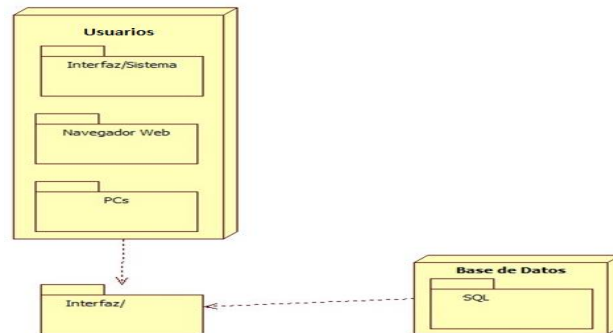


Figura 27: Vista Desarrollo. Esta figura muestra la vista de desarrollo del sistema de Facturación de la empresa VIASCALA S.A.

Fuente: Maycol Flores

4.09.04. Vista de Procesos

Esta vista específica el control de una operación en la vista lógica. Esta vista puede ser descrita como un conjunto de redes lógicas de procesos que son realizadas de forma independiente.

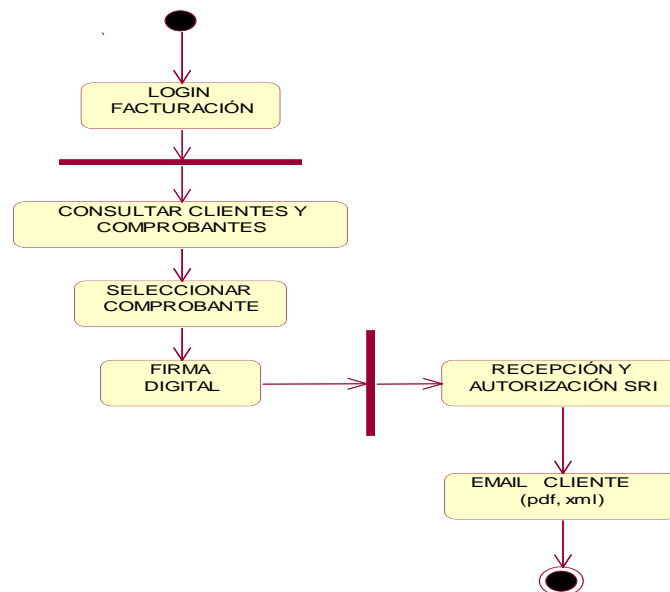


Figura 28: Vista de Procesos. Esta figura muestra la vista de procesos del sistema de Facturación de la empresa VIASCALA S.A.

Fuente: Maycol Flores

Capítulo V: Propuesta

5.01 Especificación de estándares de programación

(AREVALO, 2014)“Un estándar de programación es una forma de normalizar la programación de forma tal que al trabajar en un proyecto, cualquier persona involucrada en el mismo tenga acceso y comprenda el código.

Facilita a un programador la modificación de tu propio código fuente aunque no estés trabajando en el equipo. Definir la forma en que deben ser declaradas las variables, las clases, los comentarios, especificar qué datos deben incluirse acerca del programador y de los cambios realizados al código fuente,

El estándar de codificación completo comprende todos los aspectos de la generación de código si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico.

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software el mantenimiento del código es la facilidad con que el sistema puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuaran luego revisiones de rutina, usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software”.

5.01.01 Objetos para Ejecución

Tabla 32

Objetos para Ejecución

Tipo de Objeto	Extensión
Formulario ASP	.aspx
Biblioteca Clases	.csproj
Clase ASP	.cs
Librería de clases empaquetadas	.jar o .zip
Java Scripts	.js

Nota: La siguiente tabla muestra los objetos utilizados para la ejecución del proyecto.

Fuente: Maycol Flores

5.01.02 Objetos Multimedia

Tabla 33

Objetos Multimedia

Tipo de Objeto	Extensión
Imágenes JPG	.jpg
Imágenes GIF	.gif
Archivos MIF	.mif
Archivos de audio	.aud
Archivos flash	.swf

Nota: La siguiente tabla muestra los objetos multimedia utilizados para el desarrollo del proyecto.

Fuente: Maycol Flores

5.01.03 Objetos Fuentes

Tabla 34

Objetos Fuentes

Tipo de Objeto	Extensión
Código ASP	.asp
Código XML	.xml
Archivo de propiedades	.properties

Nota: La siguiente tabla muestra los objetos fuentes utilizados para el desarrollo del proyecto.

Fuente: Maycol Flores

5.01.04 Arquitectura de Software

(PEDILLA, 2013)“La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementarle en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel.

Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad”.

5.01.05 Arquitectura 3 capas cliente/ servidor

La especialización de la arquitectura cliente-servidor es donde la carga se divide en tres partes(o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentre modelado el negocio) y otra para el almacenamiento (acceso a datos). Una capa solamente tiene relación con la siguiente.

5.01.06 Estándar de codificación ASP.NET

5.01.06.01 ASP.NET

Un estándar de codificación comprende todos los aspectos de la generación de código, al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada con un desarrollo más limpio y entendible dentro de los proyectos a desarrollar para tener un mejor funcionamiento de los mismos.

5.01.06.02 Comentario de las Funciones

Todos los métodos o funciones creados en el sistema deben tener un comentario, antes de su declaración, explicando que hacen para facilitar su búsqueda en caso de alguna modificación.

Ejemplo:

```
public string Encriptar(string texto)
{
    //arreglo de bytes donde guardaremos la llave
    byte[] keyArray;
    //arreglo de bytes donde guardaremos el texto
    //que vamos a encriptar
    byte[] Arreglo_a_Cifrar =
    UTF8Encoding.UTF8.GetBytes(texto);
}
```

5.01.06.03 Estructuras de Control

Permiten tomar decisiones y realizar un proceso repetidas veces. Las estructuras de control a utilizar son las siguientes: **try-catch, if, for, while, switch, do-while, else, return, goto**, las cuales nos servirán de mucha ayuda para realizar condiciones que nosotros mismos establezcamos en el desarrollo del sistema.

Ejemplo:

```
private void numeracion()
{
    try
    {
        SqlConnection cnn = BDComun.ObtenerConexion();

        {
            SqlCommand comando = new SqlCommand(string.Format("Select max(id_factura) from Factura"), cnn);
            int maxId = Convert.ToInt32(comando.ExecuteScalar());
            txtId.Text = maxId.ToString();
            maxId = maxId + 1;

            if (maxId > 9)
            {
                txtNumero.Text = "00000000" + maxId.ToString();
            }
            else
            {
                txtNumero.Text = "00000000" + maxId.ToString();
            }

            cnn.Close();
        }
    }
    catch (Exception)
    {
        int maxId = 1;
        txtNumero.Text = "00000000" + maxId.ToString();
        txtId.Text = txtNumero.Text;
    }
}
```

5.01.06.04 Uso de Variables

Una variable representa un valor que puede ser numérico o una cadena de caracteres que tiene relación con los datos que se desean ingresar, cada uno de estos tipos de datos se pueden almacenar en diferentes variables según la necesidad.

Ejemplo:

String; Representa una cadena de caracteres.

Int: Representa un entero sin decimal.

Float: Representa un número con parte decimal o en coma flotante de 32 bits.

Boolean: Representa un valor lógico.

Double: Representa un valor numérico en coma flotante de 64 bits.

Objeto: Representa un objeto.

5.02 Diseño de Interfaces de Usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Normalmente suelen ser fáciles de entender y fáciles de accionar siendo amigables e intuitivos, son aquellas que incluyen elementos como menús, ventanas y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora brindando tanto comodidad, como eficiencia.

5.02.01 Tipos de Interfaces de Usuario

5.02.01.01 Interfaz de Hardware

Nivel de los dispositivos utilizados para ingresar, procesar y entregar los datos teclado, ratón y pantalla.

5.02.01.02 Interfaz de Software

Destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.

5.02.01.03 Interfaz de Software-Hardware

Establece un puente entre la máquina y las personas, permite a la máquina entender la instrucción y al hombre entender el código binario traducido a información legible.

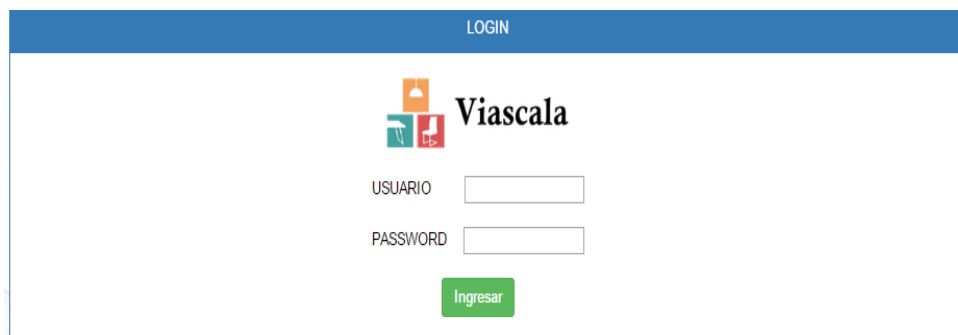


Figura 29: Interfaz de ingreso. Esta figura solicita los datos del usuario para ingresar al sistema.

Fuente: Maycol Flores

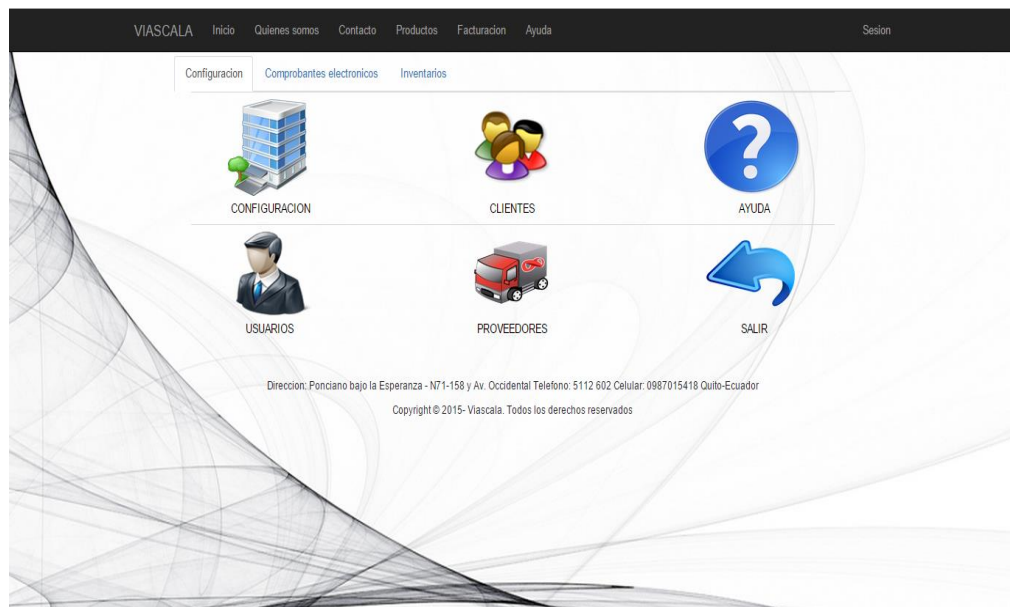
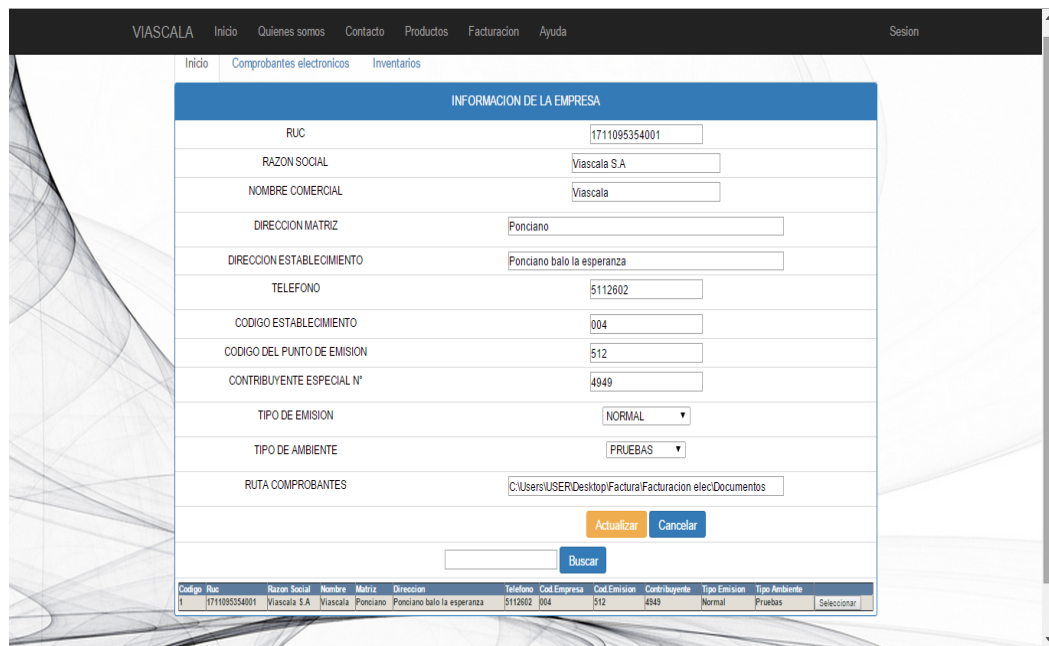


Figura 30: Interfaz de ingreso. Esta figura muestra como resultado el ambiente de trabajo que realizará el usuario.

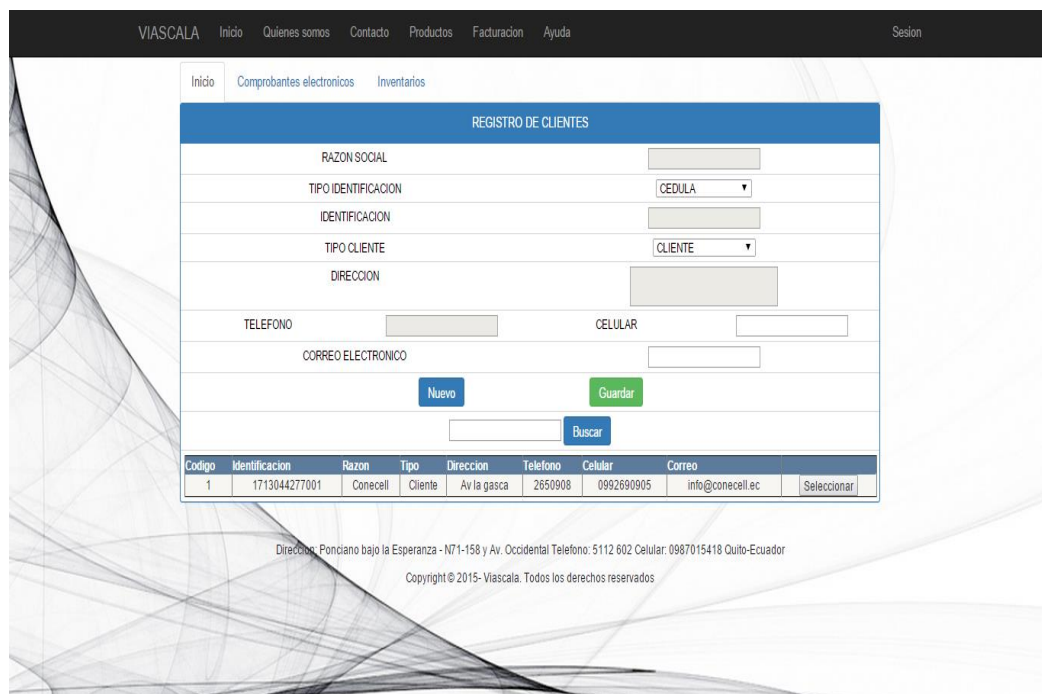
Fuente: Maycol Flores



Código	RUC	Razon Social	Nombre	Matriz	Direccion	Telefono	Cod. Empresa	Cod. Emision	Contribuyente	Tipo Emision	Tipo Ambiente	
1	1711095354001	Viascala S.A	Viascala	Ponciano	Ponciano bajo la esperanza	5112602	004	512	4949	Normal	Pruebas	Seleccionar

Figura 31: Interfaz de ingreso. Esta figura muestra el registro de datos de la empresa.

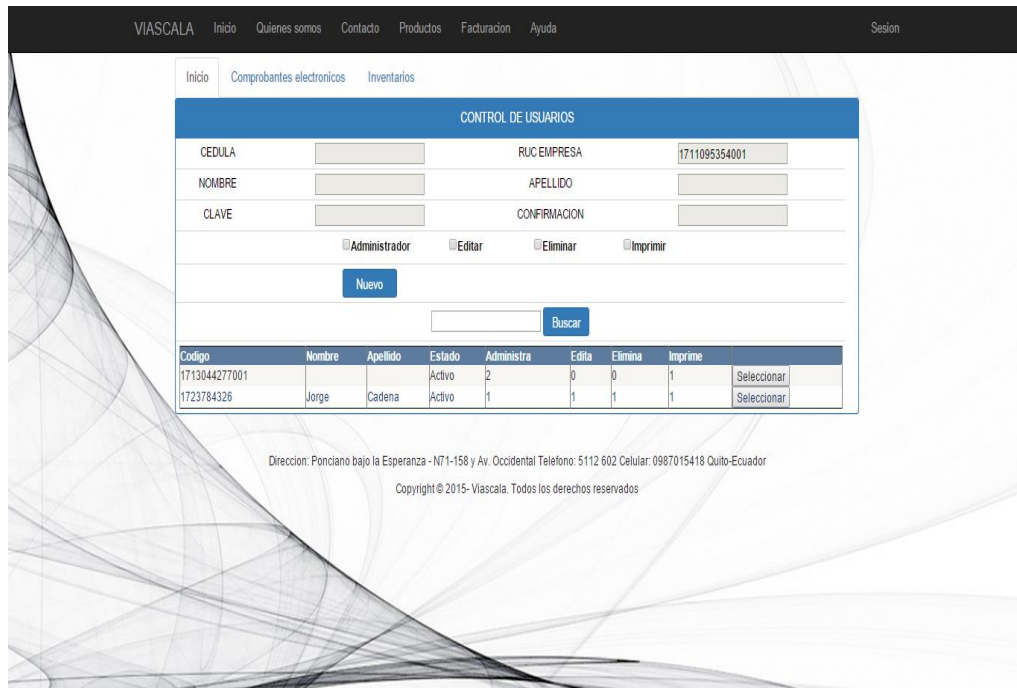
Fuente: Maycol Flores



Código	Identificacion	Razon	Tipo	Direccion	Telefono	Celular	Correo	
1	1713044277001	Conecell	Cliente	Av la gasca	2650908	0992690905	info@conecell.ec	Seleccionar

Figura 32: Interfaz de ingreso. Esta figura muestra el registro de datos de los clientes.

Fuente: Maycol Flores

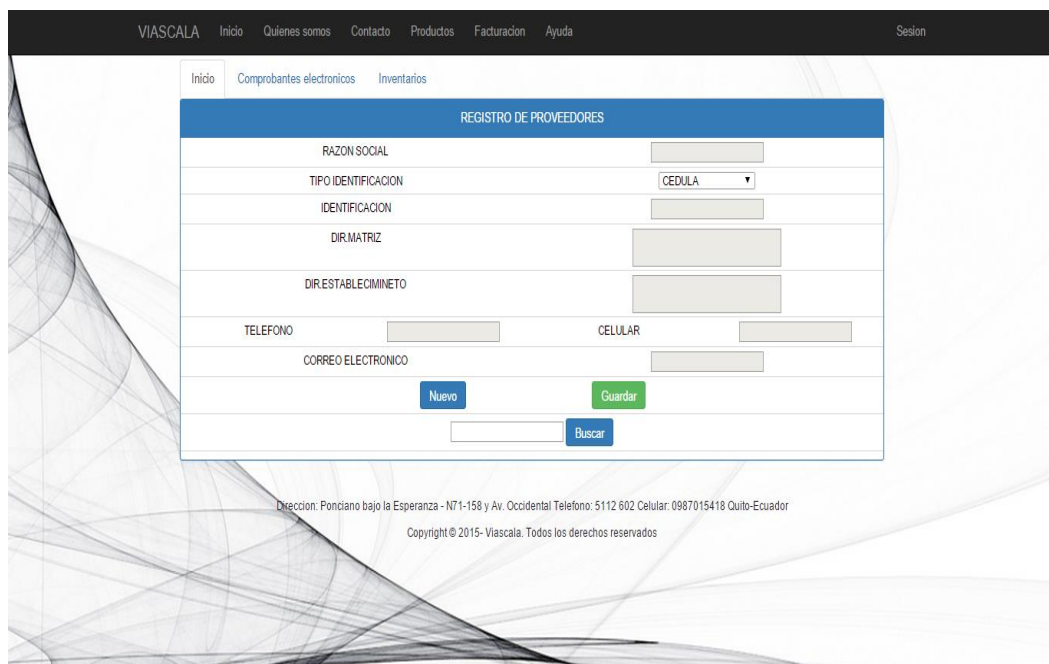


Codigo	Nombre	Apellido	Estado	Administra	Edita	Elimina	Imprime	Seleccionar
1713044277001			Activo	2	0	0	1	Seleccionar
1723784326	Jorge	Cadena	Activo	1	1	1	1	Seleccionar

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 33: Interfaz de ingreso. Esta figura muestra el registro y control de usuarios.

Fuente: Maycol Flores

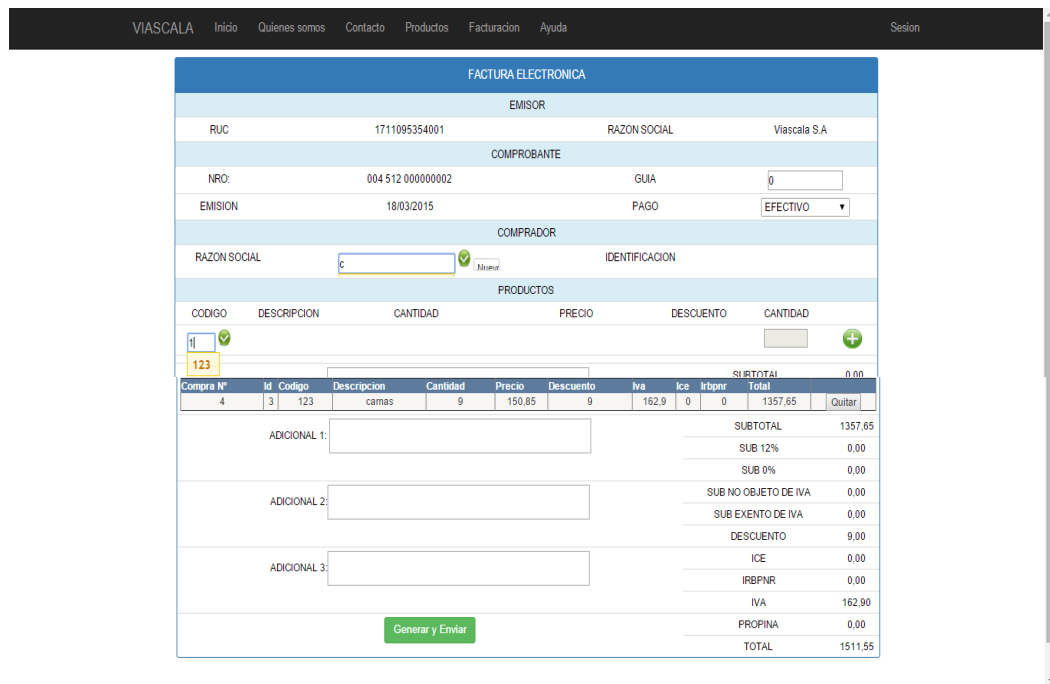


Codigo	Nombre	Apellido	Estado	Administra	Edita	Elimina	Imprime	Seleccionar
1713044277001			Activo	2	0	0	1	Seleccionar
1723784326	Jorge	Cadena	Activo	1	1	1	1	Seleccionar

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 34: Interfaz de ingreso. Esta figura muestra el registro de datos de los proveedores.

Fuente: Maycol Flores



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

FACTURA ELECTRÓNICA

EMISOR

RUC: 1711095354001 RAZÓN SOCIAL: Viascala S.A.

COMPROBANTE

NRO: 004 512 000000002 GUÍA: 0

EMISIÓN: 18/03/2015 PAGO: EFECTIVO

COMPRADOR

RAZÓN SOCIAL: C IDENTIFICACIÓN:

PRODUCTOS

CODIGO	DESCRIPCION	CANTIDAD	PRECIO	DESCUENTO	CANTIDAD
123	camas	9	150,85	9	162,9

SUBTOTAL 1357,65

ADICIONAL 1:

ADICIONAL 2:

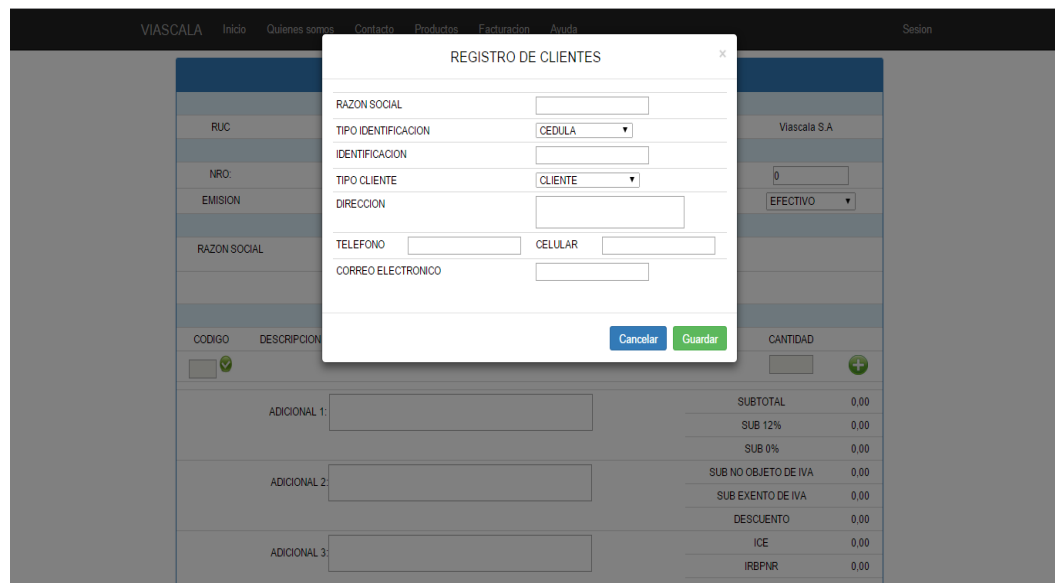
ADICIONAL 3:

Generar y Enviar

Compra N°	Id	Codigo	Descripcion	Cantidad	Precio	Descuento	Iva	Ice	IrBPNR	Total	Quitar
4	3	123	camas	9	150,85	9	162,9	0	0	1357,65	
										SUBTOTAL	1357,65
										SUB 12%	0,00
										SUB 0%	0,00
										SUB NO OBJETO DE IVA	0,00
										SUB EXENTO DE IVA	0,00
										DESCUENTO	9,00
										ICE	0,00
										IRBPNR	0,00
										IVA	162,90
										PROPINA	0,00
										TOTAL	1511,55

Figura 35: Interfaz de ingreso. Esta figura muestra la factura electrónica del sistema.

Fuente: Maycol Flores



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

REGISTRO DE CLIENTES

RAZÓN SOCIAL:

TIPO IDENTIFICACIÓN: CEDULA

IDENTIFICACIÓN:

TIPO CLIENTE: CLIENTE

DIRECCIÓN:

TELÉFONO: CELULAR:

CORREO ELECTRÓNICO:

Cancelar **Guardar**

Figura 36: Interfaz de ingreso. Esta figura muestra el registro de clientes en la factura electrónica en caso de no estar registrado en pantallas modales.

Fuente: Maycol Flores

VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

RETENCIONES

DATOS DEL EMISOR

RUC: 1711095354001 RAZON SOCIAL: Viascala S.A.

IDENTIFICACION COMPROBANTE

NRO: 004 512 000000001 FECHA DE EMISION: 18/03/2015 20:14:53

COMPROBANTE AFECTADO

ID COMPROBANTE:

MOTIVO:

DATOS DEL COMPRADOR

RAZON SOCIAL: IDENTIFICACION

PRODUCTOS

RETENCIONES	SUBTOTAL	0,00
<input type="button" value="Nuevo"/>	SUB 12%	0,00
	SUB 0%	0,00
ADICIONAL: <input type="text"/>	SUB NO OBJETO DE IVA	0,00
	SUB EXENTO DE IVA	0,00
ADICIONAL: <input type="text"/>	DESCUENTO	0,00
	ICE	0,00
ADICIONAL: <input type="text"/>	IRBPNR	0,00
	IVA	0,00
	PROPINA	0,00
	TOTAL	0,00

Figura 37: Interfaz de ingreso. Esta figura muestra las retenciones del sistema.

Fuente: Maycol Flores

VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

DETALLE DE RETENCION

DOC. SUSTENTO: 000000001

TIPO:

DESCRIPCION:

PORCENTAJE:

TOTAL:

PRODUCTOS

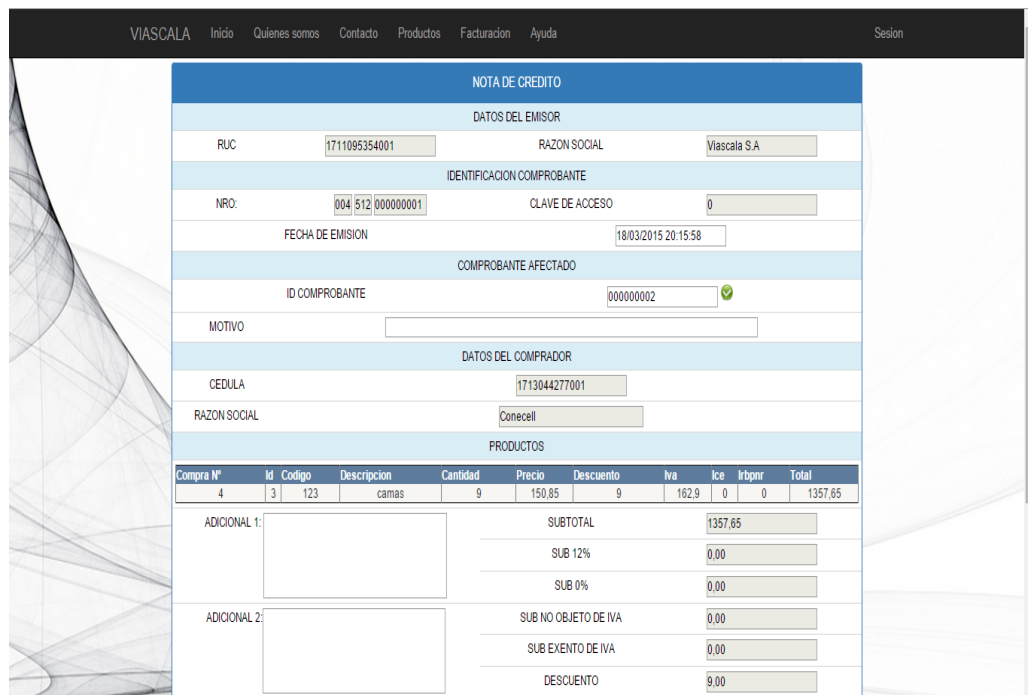
Compra N°	Id	Codigo	Descripcion	Cantidad	Precio	Descuento	Iva	Ice	Irpbpr	Total
2	1	1	camas	1	150,85	1	18,1	0	0	150,85
3	2	1	camas	1	150,85	1	18,1	0	0	150,85

RETENCIONES

RETENCIONES	SUBTOTAL	301,70
<input type="button" value="Nuevo"/>	SUB 12%	0,00
	SUB 0%	0,00
ADICIONAL: <input type="text"/>	SUB NO OBJETO DE IVA	0,00
	SUB EXENTO DE IVA	0,00
ADICIONAL: <input type="text"/>	DESCUENTO	2,00
	ICE	0,00
ADICIONAL: <input type="text"/>	IRBPNR	0,00
	IVA	36,20

Figura 38: Interfaz de ingreso. Esta figura muestra el detalle de retención en pantallas modales.

Fuente: Maycol Flores



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

NOTA DE CREDITO

DATOS DEL EMISOR

RUC: 1711095354001 RAZON SOCIAL: Viascala S.A

IDENTIFICACION COMPROBANTE

NRO: 004 512 000000001 CLAVE DE ACCESO: 0

FECHA DE EMISION: 18/03/2015 20:15:58

COMPROBANTE AFECTADO

ID COMPROBANTE: 000000002

MOTIVO:

DATOS DEL COMPRADOR

CEDULA: 1713044277001 RAZON SOCIAL: Conecell

PRODUCTOS

Compra N°	Id	Código	Descripción	Cantidad	Precio	Descuento	Iva	Ice	Irtpmr	Total
4	3	123	camas	9	150,85	9	162,9	0	0	1357,65

ADICIONAL 1:

ADICIONAL 2:

SUBTOTAL: 1357,65

SUB 12%: 0,00

SUB 0%: 0,00

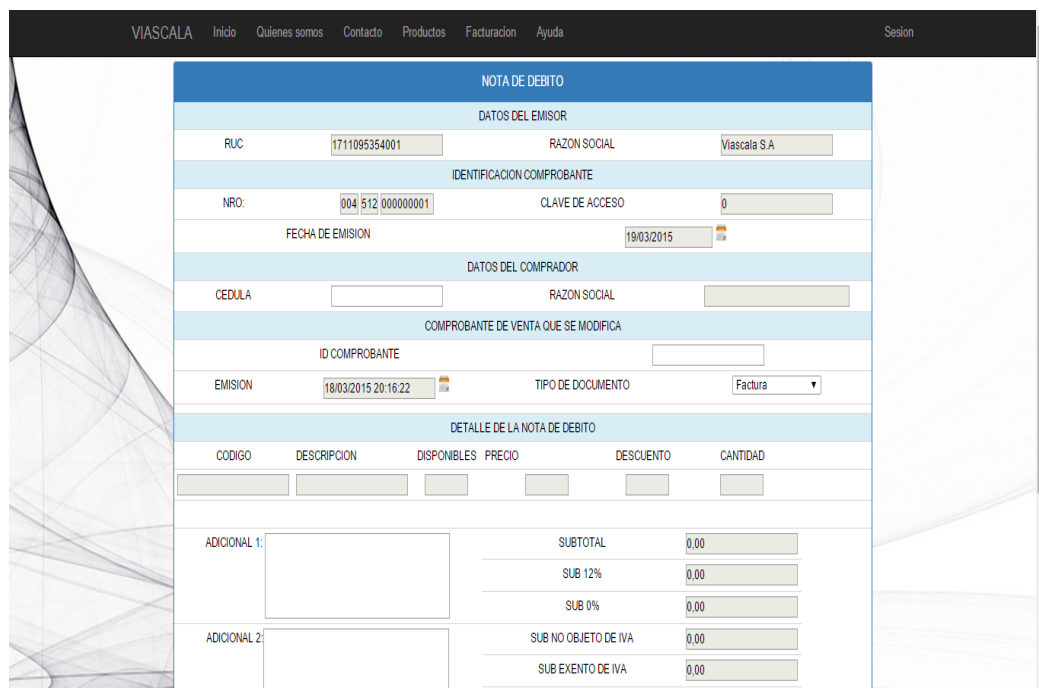
SUB NO OBJETO DE IVA: 0,00

SUB EXENTO DE IVA: 0,00

DESCUENTO: 9,00

Figura 39: Interfaz de ingreso. Esta figura muestra la nota de crédito del sistema.

Fuente: Maycol Flores



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

NOTA DE DEBITO

DATOS DEL EMISOR

RUC: 1711095354001 RAZON SOCIAL: Viascala S.A

IDENTIFICACION COMPROBANTE

NRO: 004 512 000000001 CLAVE DE ACCESO: 0

FECHA DE EMISION: 19/03/2015

DATOS DEL COMPRADOR

CEDULA: RAZON SOCIAL:

COMPROBANTE DE VENTA QUE SE MODIFICA

ID COMPROBANTE:

EMISION: 18/03/2015 20:16:22 TIPO DE DOCUMENTO: Factura

DETALLE DE LA NOTA DE DEBITO

CODIGO	DESCRIPCION	DISPONIBLES	PRECIO	DESCUENTO	CANTIDAD

ADICIONAL 1:

ADICIONAL 2:

SUBTOTAL: 0,00

SUB 12%: 0,00

SUB 0%: 0,00

SUB NO OBJETO DE IVA: 0,00

SUB EXENTO DE IVA: 0,00


Figura 40: Interfaz de ingreso. Esta figura muestra la nota de débito del sistema.

Fuente: Maycol Flores

VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

INVENTARIOS

CODIGO 

DETALLE

PRECIO

CANTIDAD

DESCUENTO 0% ▼

IVA 12% ▼

ICE

IRBPNR

Id	Codigo	Detalle	Fecha	Precio	Cantidad	Descuento	Iva	Irbpnr	Total
1	123	Ingreso	18/03/2015 19:50:09	150.85	99	10	12	0	14934.15
2	1	Salida	18/03/2015 19:50:27	150.85	1	0	0	0	150.85
3	1	Salida	18/03/2015 19:50:44	150.85	1	0	0	0	150.85
4	123	Salida	18/03/2015 20:02:14	150.85	9	0	0	0	1357.65
5	123	Salida	18/03/2015 20:10:55	150.85	1	0	0	0	150.85

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 41: Interfaz de ingreso. Esta figura muestra los inventarios del sistema.

Fuente: Maycol Flores

VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

PRODUCTOS

Numero	Codigo	Detalle	Cantidad	Precio	Descuento	Iva	Ice	Irbpnr	Total
1	123	camas	89	150.85	1	18.1	0	0	14951.25

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 42: Interfaz de ingreso. Esta figura muestra la búsqueda o ingreso de productos en el sistema.

Fuente: Maycol Flores

VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

CONSULTA DE COMPROBANTES


RAZON SOCIAL	<input type="text" value="Estar.net"/>		
TIPO IDENTIFICACION	<input type="text" value="CEDULA"/>		
IDENTIFICACION	<input type="text" value="1720758935001"/>		
TIPO CLIENTE	<input type="text" value="CLIENTE"/>		
DIRECCION	<input type="text" value="El condado"/>		
TELEFONO	<input type="text" value="2650908"/>	CELULAR	<input type="text" value="0987015418"/>
CORREO ELECTRONICO	<input type="text" value="estar@net.ec"/>		
<input type="button" value="Buscar"/>			

Codigo	Ruc	Razon	Tipo	Numero	Fecha	Subtotal	Iva	Total	PDF
5	1711095354001	Viascala S.A	01	000000005	18/03/2015	150,85	18,1	167,95	PDF

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 43: Interfaz de ingreso. Esta figura muestra la consulta de los comprobantes electrónicos.

Fuente: Maycol Flores



Viascala

VIASCALA S.A
VIASCALA
DIRECCIÓN SUCURSAL: Ponciano
CONTRIBUYENTE ESPECIAL No:
OBLIGADO A LLEVAR CONTABILIDAD: Si

RUC: 1711095354001
FACTURA
No: 000000004
NÚMERO DE AUTORIZACIÓN
0
FECHA Y HORA DE AUTORIZACIÓN: 18/03/2015
AMBIENTE: Pruebas
EMISIÓN: Normal
CLAVE DE ACCESO
180320150117110953540012004512000000004010203041K

Razón Social/Nombre y Apellidos: Super Paco
Fecha de emisión: 18/03/2015
RUC/CI: 1723784326

Cod	Cant	Descripción	Precio Unitario	Descuento 1%	Descuento 2%	Descuento 3%	IVA	Precio Total
4	1	camas	150,85				18,10	167,95

INFORMACIÓN ADICIONAL
Dirección: El recreo
Telefono: 2569852
Email: info@paco.com
Usuario: 1723784326
Clave: 1723784326

SUBTOTAL IVA 12% 150,85
SUBTOTAL IVA 0% 0,00
SUBTOTAL NO OBJ IVA 0,00
SUBTOTAL EXENTO IVA 0,00
TOTAL DESCUENTO 1,00
IVA 12% 18,10
VALOR TOTAL 167,95

Figura 44: Interfaz de ingreso. Esta figura muestra el resultado de la generación de la factura electrónica.

Fuente: Maycol Flores

5.03 Especificación de pruebas de unidad

(B., 2010) “Al desarrollar un nuevo software o sistema de información, la primera etapa de pruebas a considerar es la etapa de pruebas unitarias o también llamada pruebas de caja blanca (White Box), estas pruebas también son llamadas pruebas modulares ya que nos permiten determinar si un módulo del programa está listo y correctamente terminado, estas pruebas no se deben confundir con las pruebas informales que realiza el programador mientras está desarrollando el módulo”.

Estas pruebas nos ayudaran a visualizar la funcionalidad de cada módulo probando así el modulo está listo y correctamente terminado a continuación se estableció las pruebas de unidad al aplicativo web en desarrollo y se obtuvieron los siguientes resultados:

Tabla 35

Especificación de pruebas de unidad Ingreso al Sistema

Identificador de la Prueba:	PU001
Método a Probar	Ingreso al Sistema
Objetivo de la Prueba	Verificar que los datos sean ingresados correctamente.
Datos De Entrada	Usuario, Clave
Resultado Esperado	Una vez ingresado los datos correctos el usuario podrá usar el sistema.
Comentarios	Para ingresar al sistema se requiere llenar todos los campos requeridos

Notas: La siguiente tabla muestra la especificación de pruebas de unidad para el ingreso al sistema.

Fuente: Maycol Flores

Tabla 36

Especificación de pruebas de unidad Registro de Usuarios

Identificador de la Prueba:	PU002
Método a Probar	Registro de Usuarios
Objetivo de la Prueba	Verificar que los permisos asignados a los usuarios para el ingreso al sistema estén correctos.
Datos De Entrada	Cedula , Nombre, Apellido, Password, Rol
Resultado Esperado	El sistema solicite que todos los campos estén llenos
Comentarios	Todos los campos son requeridos para el ingreso de información de los usuarios

Notas: La siguiente tabla muestra la especificación de pruebas de unidad para el registro de usuarios.

Fuente: Maycol Flores

Tabla 37

Especificación de pruebas de unidad Registro de Clientes

Identificador de la Prueba:	PU003
Método a Probar	Registro de Clientes
Objetivo de la Prueba	Verificar que todos los campos del sistema estén validados para el ingreso de información de los clientes.
Datos De Entrada	Cedula o Ruc , Nombre, Apellido, Dirección, Teléfono
Resultado Esperado	El sistema solicite que todos los campos estén llenos
Comentarios	Todos los campos son requeridos para el ingreso de información de los clientes

Notas: La siguiente tabla muestra la especificación de pruebas de unidad para el registro de clientes.

Fuente: Maycol Flores

Tabla 38

Especificación de pruebas de unidad Llenado de Comprobantes Electrónicos

Identificador de la Prueba:	PU004
Método a Probar	Llenado de Comprobantes Electrónicos Llenado automático de datos si el cliente está registrado.
Objetivo de la Prueba	
Datos De Entrada	Datos del cliente, empresa, productos
Resultado Esperado	El sistema no permita información mal ingresada
Comentarios	El sistema advierte que el cliente no está registrado en caso de no estarlo, muestra la cantidad de productos existentes en la empresa (stock).

Notas: La siguiente tabla muestra la especificación de pruebas de unidad para el llenado de comprobantes electrónicos.

Fuente: Maycol Flores

Tabla 39

Especificación de pruebas de unidad Factura Electrónica

Identificador de la Prueba:	PU005
Método a Probar	Factura Electrónica (Archivo xml, pdf)
Objetivo de la Prueba	Verificar que los archivos (XML, PDF) generados de la factura sea correctos cumpliendo con los requisitos del SRI.
Datos De Entrada	Datos del cliente, empresa, productos
Resultado Esperado	El sistema genere los archivos (XML, PDF) sin errores
Comentarios	El sistema guardara los archivos (XML, PDF) en una carpeta predeterminada en el ordenador

Notas: La siguiente tabla muestra la especificación de pruebas de la factura electrónica.

Fuente: Maycol Flores

Tabla 40

Especificación de pruebas de unidad Stock de Productos

Identificador de la Prueba:	PU006
Método a Probar	Stock de Productos (Ingresos y Salidas)
Objetivo de la Prueba	Verificar que los productos vendidos sean restados del stock y los productos entrantes a bodega sean sumandos al stock.
Datos De Entrada	Productos
Resultado Esperado	El sistema no tenga errores en las existencias de los productos
Comentarios	El sistema solicitará el código del producto para registrar la salida o entrada del mismo.

Notas: La siguiente tabla muestra la especificación de pruebas de unidad para el stock de productos.

Fuente: Maycol Flores

5.04 Especificación de pruebas de aceptación

(JIMENEZ, 2013) “Una prueba de aceptación es un escenario de utilización del sistema y el comportamiento que de él se espera, visto desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa. Las pruebas de aceptación constituyen el criterio de éxito en cuanto a la implementación de un requisito del sistema. Un mismo requisito del sistema puede presentarse en ejecución como diferentes escenarios (por ejemplo alternativas correspondientes a elecciones que realiza el usuario al interactuar con el sistema).” Con este concepto se ha procedido a realizar las pruebas de aceptación de la aplicación a realizarse para la empresa VIASCALA.

Tabla 41

Especificación de pruebas de aceptación N°1

Identificador de la Prueba	PA001
Caso de uso	UC001
Tipo de usuario	Vendedor, Cliente
Objetivo de la Prueba	Mostrar campos validados para registrar correctamente los datos y ordenes de pedido del cliente
Secuencia de eventos	<ol style="list-style-type: none"> 1. Registro 2. Verificación 3. Emite orden
Resultados Esperados	El sistema registre correctamente la información ingresada
Comentarios	Si hay información mal ingresada el sistema exigirá que se llene los campos vacíos
Estado : Aceptado	

Notas: La siguiente tabla muestra la especificación de pruebas de aceptación N°1

Fuente: Maycol Flores

Tabla 42

Especificación de pruebas de aceptación N°2

Identificador de la Prueba	PA002
Caso de uso	UC002
Tipo de usuario	Encargado de bodega, Cliente
Objetivo de la Prueba	Mostrar campos validados para el correcto despacho de productos y registro de los mismos en el sistema
Secuencia de eventos	<ol style="list-style-type: none"> 1. Registro 2. Actualización 3. Eliminación 4. Consultas
Resultados Esperados	La información ingresada correctamente en el sistema
Comentarios	Si hay información mal ingresada el sistema exigirá que se llene los campos vacíos
Estado : Aceptado	

Notas: La siguiente tabla muestra la especificación de pruebas de aceptación N°2

Fuente: Maycol Flores

Tabla 43

Especificación de pruebas de aceptación N°3

Identificador de la Prueba	PA003
Caso de uso	UC003
Tipo de usuario	Encargado de Bodega
Objetivo de la Prueba	Mostrar campos validados, para el correcto registro de entrada y salida de productos
Secuencia de eventos	
<ol style="list-style-type: none"> 1. Registro 2. Actualización 3. Eliminación 4. Consultas 	
Resultados Esperados	
La información ingresada correctamente en el sistema	
Comentarios	
El sistema advertirá si la información está mal ingresada o existen campos vacíos	
Estado : Aceptado	

Notas: La siguiente tabla muestra la especificación de pruebas de aceptación N°3

Fuente: Maycol Flores

Tabla 44

Especificación de Pruebas de Aceptación N°4

Identificador de la Prueba	PA004
Caso de uso	UC004
Tipo de usuario	Contador
Objetivo de la Prueba	Comprobar que la generación de reportes sea con la información estadística correcta de los comprobantes electrónicos.
Secuencia de eventos	
<ol style="list-style-type: none"> 1. Ingresar 2. consultar 	
Resultados Esperados	
No exista errores en la información generada e ingresada	
Comentarios	
Obtener los mejores resultados para el buen manejo del sistema	
Estado : Aceptado	

Notas: La siguiente tabla muestra la especificación de pruebas de aceptación N°4

Fuente: Maycol Flores

Tabla 45

Especificación de Pruebas de Aceptación

Identificador de la Prueba	PA005
Caso de uso	UC005
Tipo de usuario	Administrador, Usuarios
Objetivo de la Prueba	Verificar el correcto ingreso al sistema por parte de los roles de usuario.
Secuencia de eventos	
1. Ingresar usuario y contraseña correspondientes	
Resultados Esperados	
Considerando los privilegios de usuario verificar la información necesaria y correcta para el acceso al sistema	
Comentarios	
Obtener los mejores resultados para el buen manejo del sistema	
Estado : Aceptado	

Notas: La siguiente tabla muestra la especificación de pruebas de aceptación N°5

Fuente: Maycol Flores

5.05 Especificación de pruebas de carga

(Morrow, 2014)“La carga de trabajo se refiere a la capacidad máxima que tiene un servidor web (hardware y software), para atender a un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación”.

Tabla 46

Especificación de pruebas de carga N°1

Identificador de la prueba	PC001
Tipo de prueba	Desenvolvimiento del sistema con un elevado número de peticiones de usuarios conectados concurrentes.
Objetivo de la prueba	Analizar el comportamiento del sistema con un alto nivel de usuarios en línea
Descripción	

Se utiliza el programa "JMETER", para realizar la prueba

Número de hilos: 20

Periodo de tiempo: 5seg.

Contador de bucle: 1

Resultados Esperados

Conocer la cantidad de usuarios que soporta el software ingresando al sistema al mismo tiempo.

Comentarios

Con los datos ingresados el sistema muestra un límite menor al número ingresado.

Notas: La siguiente tabla muestra la especificación de pruebas de carga N°1.

Fuente: Maycol Flores

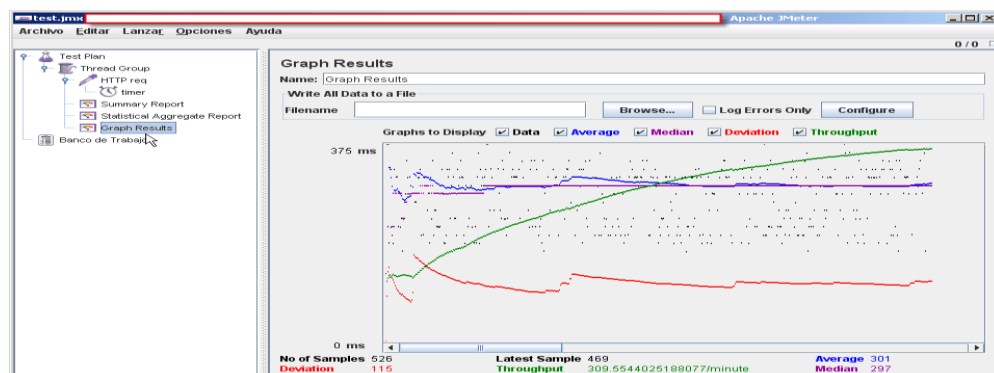


Figura 45: Resultados de pruebas realizadas en JMeter.- En la figura observamos los resultados de las pruebas realizadas en el JMeter

Fuente: Maycol Flores

Tabla 47

*Especificación de pruebas de carga N°2***Identificador****PC002****de la prueba**

Tipo de prueba	Simulación de desempeño con un número alto de peticiones concurrentes.
Objetivo de la prueba	Conocer el comportamiento del sistema en base a número determinado de peticiones y con ello analizar y construir la evolución de los tiempos de respuesta al igual que su rendimiento.
Descripción	

Jmeter – aplicación utilitaria que mide el comportamiento y el rendimiento de aplicaciones web.

Como estrategia se utilizará una función de Jmeter que se denomina Grupo de Hilos (Number of Threads) en la cual diseñaremos y configuraremos un conjunto de peticiones (Http Request).

Los valores que registraremos son:

3 peticiones (hilos)

Nombre de servidor o IP: clothesme.herokuapp.com

método: GET

Resultados Esperados

Rápido procesamiento de información y datos legítimos

Comentarios

Obtener los mejores resultados para el buen manejo del sistema.

Notas: La siguiente tabla muestra la especificación de pruebas de carga N°2.

Fuente: Maycol Flores

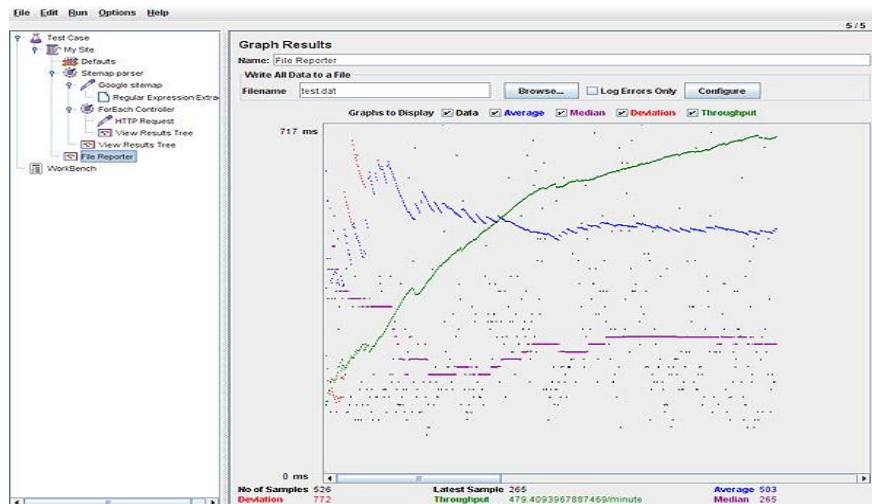


Figura 46: Resultados de pruebas realizadas en JMeter.- En la figura observamos la los resultados de las pruebas realizadas en el JMeter.

Fuente: Maycol Flores

5.06 Configuración del Ambiente mínima/ideal

Para la ejecución de la aplicación web creado para la empresa Viascala se debe tener muy en cuenta algunos puntos tales como las características del servidor que se va a utilizar en la cargar del aplicativo y de la base de datos, se necesita un servidor con las siguientes características.

Tabla 48

Configuración del ambiente mínima / ideal

Servidor web	Cuándo utilizarlo
IIS Express	Utilícelo cuando el servidor web de destino sea IIS 7 pero no desea (o no puede) usar la versión completa de IIS 7. Esto requiere Visual Studio 2010 Service Pack 1 y se debe instalar IIS Express de manera separada. IIS Express hospeda los sitios de una manera muy similar a IIS 7.
Servidor de desarrollo de Visual Studio	Utilícelo cuando trabaje con un proyecto existente o si el sitio tiene como destino una versión anterior de IIS, como IIS 6, y no es importante que el entorno de prueba coincida exactamente con el entorno de producción. Esta opción de servidor es el valor predeterminado en Visual Studio. Sin embargo, el servidor de desarrollo de Visual Studio se ejecuta en un contexto de seguridad diferente al de IIS completo y no revela los errores que se puedan producir durante la implementación en una versión de producción de IIS.
IIS	Utilícelo cuando desee probar la aplicación web con el entorno del servidor que esté más próximo a donde se ejecutará el sitio activo y es práctico instalar y trabajar con IIS en el equipo de desarrollo. Sin embargo, puede ser más complejo configurar la depuración y otras tareas que si utiliza IIS Express o el entorno de desarrollo de Visual Studio. Es necesario que ejecute Visual Studio como administrador.
Importante	
Si utiliza IIS Express o el Servidor de desarrollo de Visual Studio, se recomienda que pruebe la aplicación en el servidor IIS de destino antes de implementarla en un sitio.	

Nota: La siguiente tabla muestra la configuración de ambiente mínima / ideal.

Fuente: Maycol Flores

Para instalar ASP.NET en un servidor en el que se ejecute Windows Server

2003 utilizando el Asistente para configurar su servidor

1. En la barra de tareas, haga clic en el botón Inicio y, a continuación, haga clic en Administrar su servidor y, en la ventana Administrar su servidor, haga clic en Agregar o quitar una función.

2. En el Asistente para configurar su servidor, haga clic en Siguiente, y en el cuadro de diálogo Función del servidor, active Servidor de aplicaciones Web (IIS, ASP.NET) y, después, haga clic en Siguiente.
3. En el cuadro de diálogo Opciones del servidor de aplicaciones Web, haga clic en la casilla de verificación Habilitar ASP.NET, haga clic en Siguiente y, a continuación, vuelva a hacer clic en Siguiente.
4. Si es necesario, inserte el CD de instalación de Windows Server 2003 en la unidad de CD-ROM y, después, haga clic en Siguiente.
5. Cuando finalice la instalación, haga clic en Finalizar.

Para instalar ASP.NET en un servidor en el que se ejecute Windows Server 2003 utilizando el cuadro de diálogo Agregar o quitar programas

1. En la barra de tareas, haga clic en el botón Inicio, apunte a Panel de control y, a continuación, haga clic en Agregar o quitar programas.
2. En el cuadro de diálogo Agregar o quitar programas, haga clic en Agregar o quitar componentes de Windows.
3. En el cuadro Componentes del Asistente para componentes de Windows, haga clic en la casilla de verificación Servidor de aplicaciones Web y, a continuación, haga clic en Siguiente.
4. Una vez que el Asistente para componentes de Windows haya terminado de configurar Windows Server 2003, haga clic en Finalizar.

Para habilitar ASP.NET en el Administrador IIS en la familia de Windows Server 2003

1. En la barra de tareas, haga clic en el botón Inicio y, a continuación, haga clic en Ejecutar.

2. En el cuadro Abrir del cuadro de diálogo Ejecutar, escriba inetmgr y, después, haga clic en Aceptar.
3. En el Administrador IIS, expanda el equipo local y, a continuación, haga clic en Extensiones de servicios Web.
4. En el panel de la derecha, haga clic con el botón secundario del *mouse* (ratón) en ASP.NET y, después, haga clic en Permitir. El estado de ASP.NET cambia a Permitido.

Cuando instala Visual Studio .NET en un equipo en el que se ejecute Windows XP Professional o Windows 2000 Server, .NET Framework y ASP.NET se instalan automáticamente también. Si desea instalar ASP.NET y .NET Framework de forma independiente, puede descargarlos del Web e instalarlos en su servidor. En el procedimiento siguiente se proporcionan instrucciones sobre la forma de hacerlo.

Capítulo VI: Aspectos Administrativos

6.01 Recursos

6.01.01 Recurso Humano

Conjunto de empleados o colaboradores de una empresa los cuales son encargados de realizar las diferentes actividades de cada área que conforma la empresa.

Las personas que intervienen en el sistema son:

- Gerente
- Contador
- Vendedor
- Bodeguero

6.01.02 Recurso Material

Se da a conocer los materiales necesarios que fueron utilizados para la realización del software.

Los materiales utilizados son los siguientes:

- Computadora Escritorio (Intel Core i3 marca samsung)
- Computadora Portátil (Intel Core i5 marca sonyvaio)
- Mouse
- Libros
- Monitor
- Teclado
- Impresora
- Documentación
- Internet

6.01.03 Recurso Tecnológico

Para el desarrollo del software se necesitó de los siguientes programas:

Tabla 49

Recursos Tecnológicos

Programas	Utilidades
Visual Studio 2013 (C# Asp.net)	Programación y codificación del software
Sql Server Management Studio	Servidor de base de datos
Rational Rose Enterprise Edition	Realización de Diagramas

Nota: La siguiente tabla muestra los programas utilizados para el desarrollo del proyecto.

Fuente: Maycol Flores

6.02 Presupuesto

Los gastos que se invirtieron para la realización del proyecto de titulación y aplicada para la empresa VIASCALA fueron las siguientes:

Tabla 50

Presupuesto de Elaboración del Sistema.

DETALLE	TIEMPO	PRECIO UNITARIO	SUB TOTAL	TOTAL RUBRO
BIENES				95,00
Material de Oficina			60,00	
Material de Consulta			20,00	
Varios			15,00	
SERVICIOS				2971.45
Servicios de Internet	6 meses	80,00	480,00	
Transporte	6 meses	2.50	150,00	
Impresiones	No determinado		200,00	
Alimentación	6 meses		350.00	
Tutorías / Seminario	6 meses		722,00	
Universidad	6 meses		549,45	
Servicios Básicos	6 meses		340,00	
TOTAL				3066.45

Nota: La siguiente tabla muestra el presupuesto para la elaboración del proyecto.

Fuente: Maycol Flores

6.03 Cronograma

El cronograma es una herramienta muy importante en la gestión de proyectos. Puede tratarse de un documento impreso o de una aplicación digital; en cualquier caso, el cronograma incluye una lista de actividades o tareas con las fechas previstas de su comienzo y final.

Para poder realizar un cronograma siempre se recomienda tener en cuenta consejos tales como contar con un calendario a mano, disponer de todas las tareas que deben componer a aquel, establecer una línea del tiempo básica.

Véase en el Anexo **A.05**

Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones

1.-Luego de haber realizado el software, llegamos a la conclusión de que la implementación de la Facturación Electrónica es muy beneficiosa para la economía de las empresas debido principalmente a que origina agilidad de los procesos de facturación y pago, minimiza la reducción de errores en los procesos de facturación y su consecuente simplificación en el cumplimiento de deberes tributarios (declaración y pago de impuestos, principalmente), la disminución en riesgos de fraude, la simplificación de la administración, mejora la imagen de la compañía, el ahorro de costos operacionales de la empresa.

2.- El desarrollo del sistema de esta tesis requiere del almacenamiento de la información en una base de datos de SQL Server 2008 y la implementación de procedimientos almacenados en el motor de la base de datos, lo que permite interactuar de una manera fácil y rápida en los procesos de Ingreso, Eliminación, Actualización y Consultas.

3.- La implementación de la interfaz del sistema fue desarrollada bajo la aplicación de Visual Studio 2013 el cual facilitó el trabajo gracias a su gran cantidad de herramientas fáciles de usar, brindando interfaces de calidad intuitivas y novedosas.

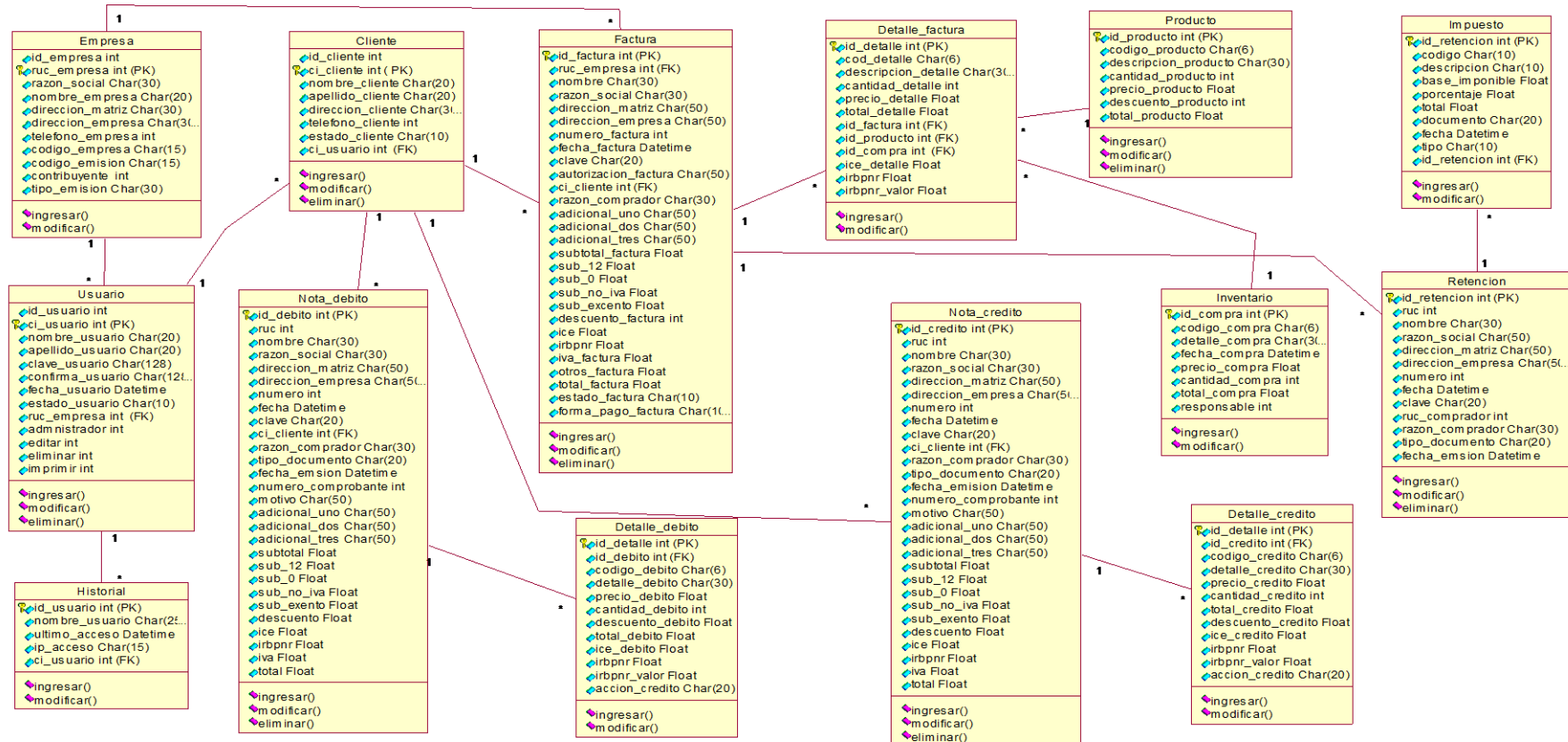
4.-Con la ayuda de la aplicación en la empresa se disminuirá el tiempo en la realización de sus actividades diarias, brindando seguridad y una información actualizada y detallada.

5.- Con las pruebas de carga realizadas se permitió un análisis más detallado a las funcionalidades del sistema para su correcto funcionamiento, cumpliendo con las peticiones y requerimientos de los usuarios.

7.02 Recomendaciones

- 1.-Realizar el mantenimiento preventivo del software para tener una mayor funcionalidad de la aplicación con información actualizada y detallada que se usan en sus actividades diarias.
- 2.- Asignar correctamente permisos (roles) a los empleados para el ingreso al sistema para tener una mayor estabilidad y seguridad de información.
- 3.-Realizar pruebas finales del sistema que permitan determinar si la totalidad de los requerimientos fueron alcanzados, en caso contrario estudiar la culminación de los requerimientos que carece el sistema.
- 4.- Crear un manual claro y sencillo del manejo del sistema en función de las actividades operacionales del Departamento de servicios que sirva como soporte y ayuda a los usuarios para el buen funcionamiento del sistema.

ANEXOS



Fuente: Maycol Flores



Fuente: Maycol Flores

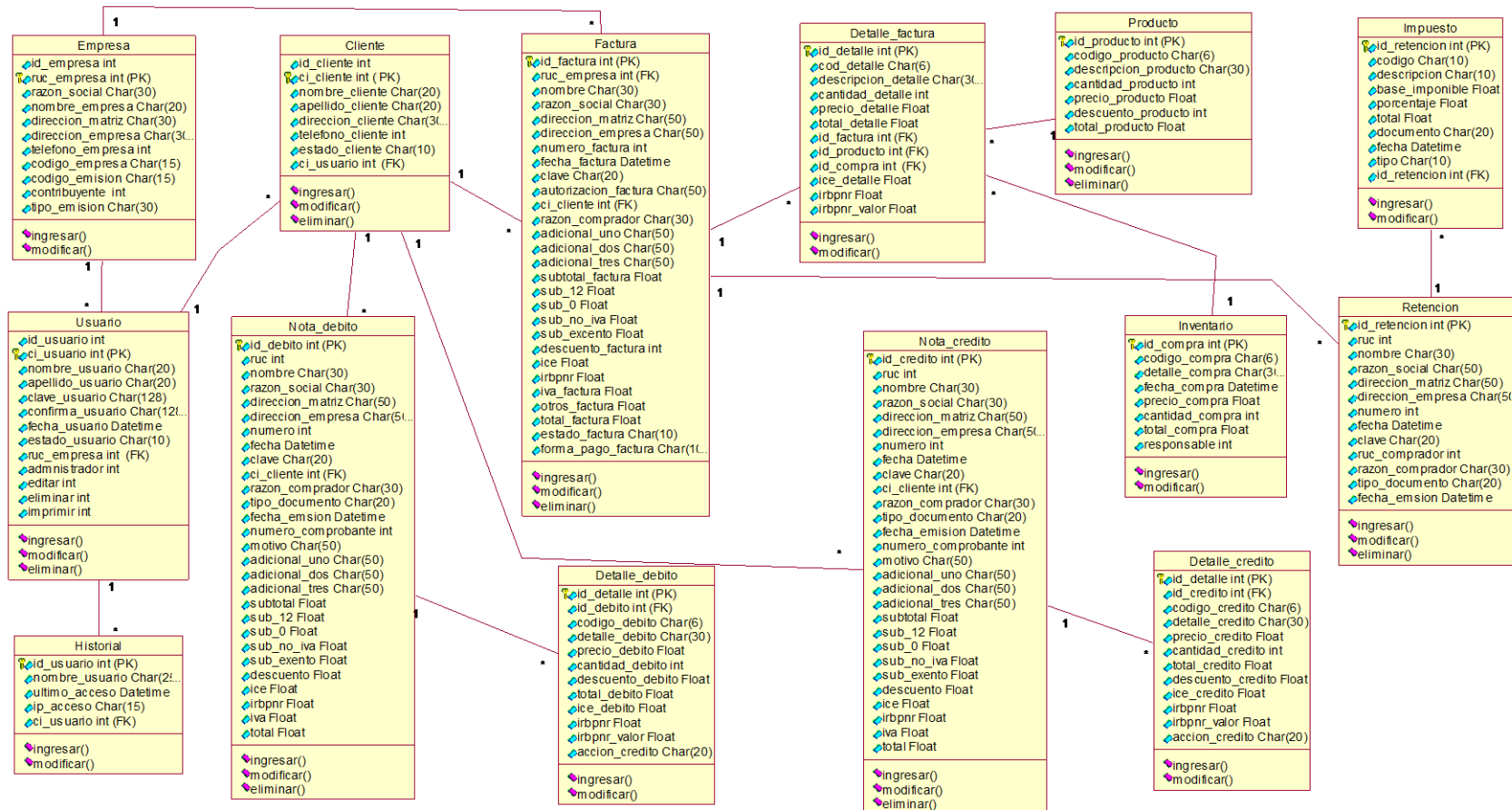


Figura 50: Vista Lógica

Fuente: Maycol Flores

A.05 Cronograma de Actividades

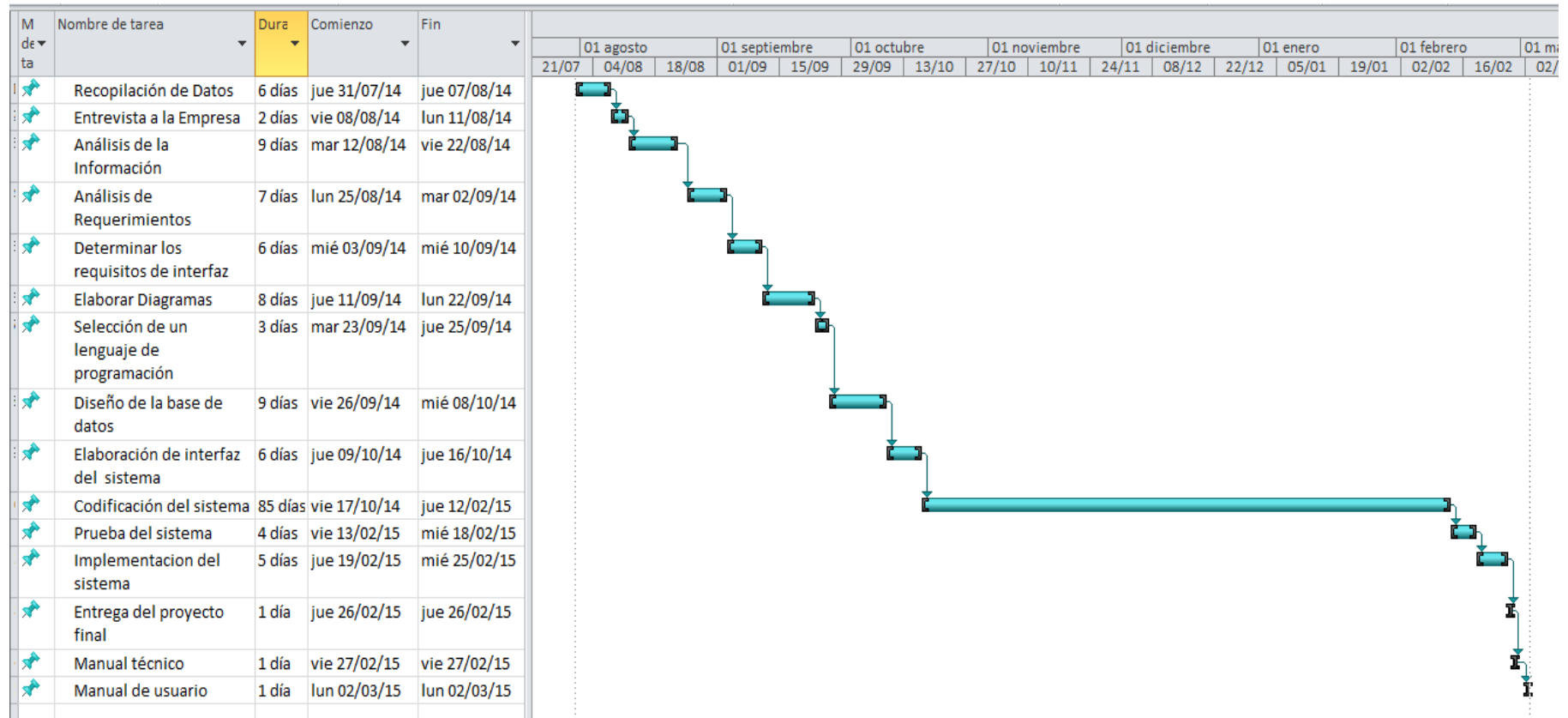


Figura 51: Cronograma de actividades

Fuente: Maycol Flores

A.06 Manual de Usuario

1. Introducción

El presente manual muestra los pasos a seguir para dar uso y administrar de forma correcta a nivel de administrador y usuario el sistema de facturación electrónica de tal forma que los procesos que se llevan a cabo en la aplicación se ejecuten correctamente para evitar complicaciones.

2. Interfaces de Usuario

2.1 Inicio de Sesión

Una vez que ejecutemos el sistema nos re direccionara a la página donde se muestra a continuación lo siguientes controles:

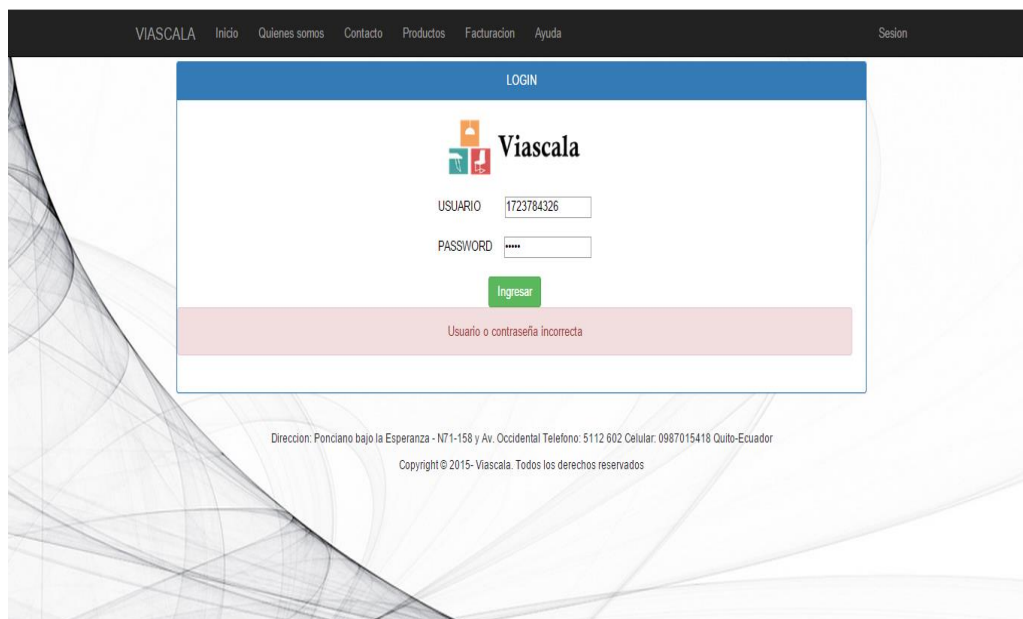


Figura 52: Página de autenticación. La presente imagen muestra la página desde donde se podrá ingresar a la parte administrativa del sistema.

Fuente: Maycol Flores

Campo de usuario: En este campo se tendrá que ingresar el nombre de usuario, el cual se le proporciona al administrador del sistema en el momento que se realiza el respectivo registro.

Campo de contraseña: En este campo se tendrá que ingresar la contraseña de usuario, la cual proporciona el administrador del sistema en el momento del registro.

Botón Ingresar: Una vez llenados los campos anteriores se debe dar clic en este botón para que nos redirija a la página de administración del sistema. Cabe recalcar que dependiendo de los privilegios que tenga el usuario se podrá acceder a la administración del sistema.

2.2 Página Principal:

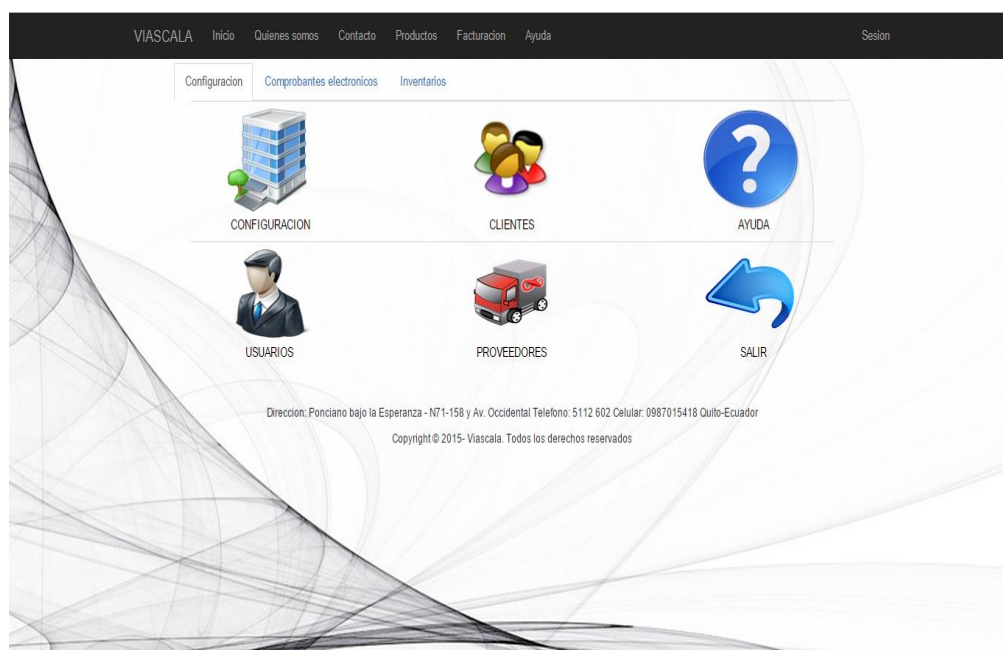


Figura 53: Página principal. La presente imagen muestra los distintos menús del sistema.

Fuente: Maycol Flores

Con este menú se podrá acceder a las distintas opciones para la administración del sistema, las cuales se describen a continuación.

2.3 Página de Configuración:

Esta opción nos permite ingresar información general de la empresa de una manera fácil y sencilla la cual se realizara de la siguiente manera cumpliendo los siguientes pasos.

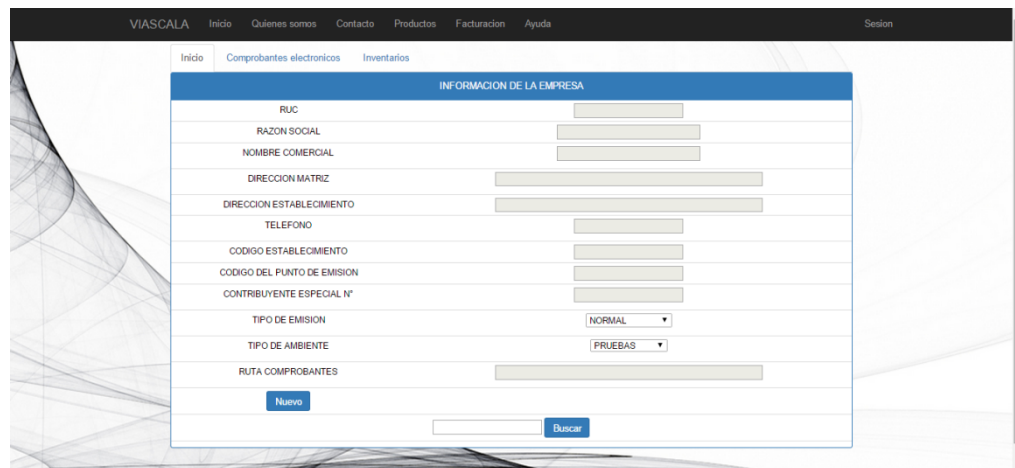


Figura 54: Página de información de la empresa. Dar clic en nuevo.

Fuente: Maycol Flores

Debemos dar clic en el botón nuevo para que se habiliten los campos para poder llenar la información, es necesario llenar todos los campos para poder guardar en el sistema caso contrario saldrá el mensaje de error.

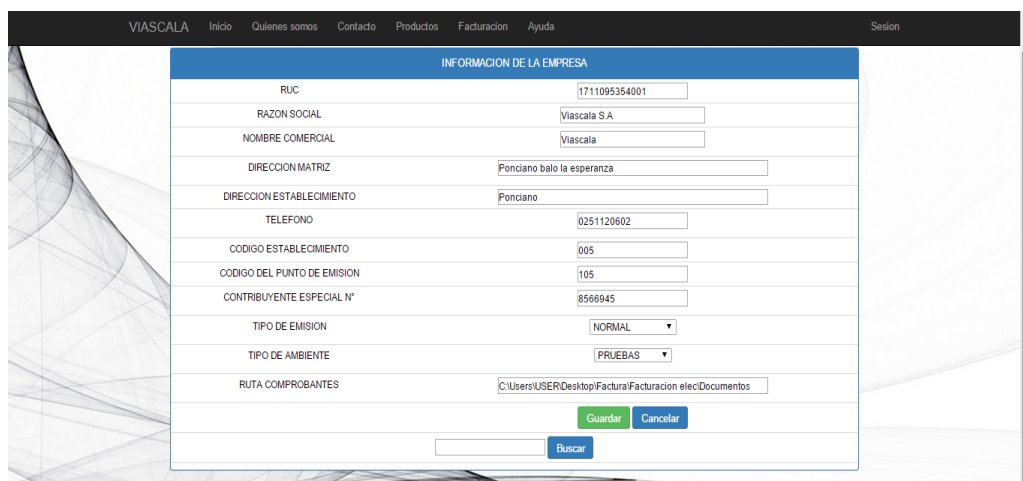


Figura 55: Pagina de información de la empresa. Llenar todos los campos.

Fuente: Maycol Flores

2.4 Página de Usuarios:

Esta opción nos permite ingresar al control de usuarios del sistema para asignar permisos a cada usuario creado, para garantizar un buen uso del sistema.



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

CONTROL DE USUARIOS

CEDULA RUC EMPRESA

NOMBRE APELLIDO

CLAVE CONFIRMACION

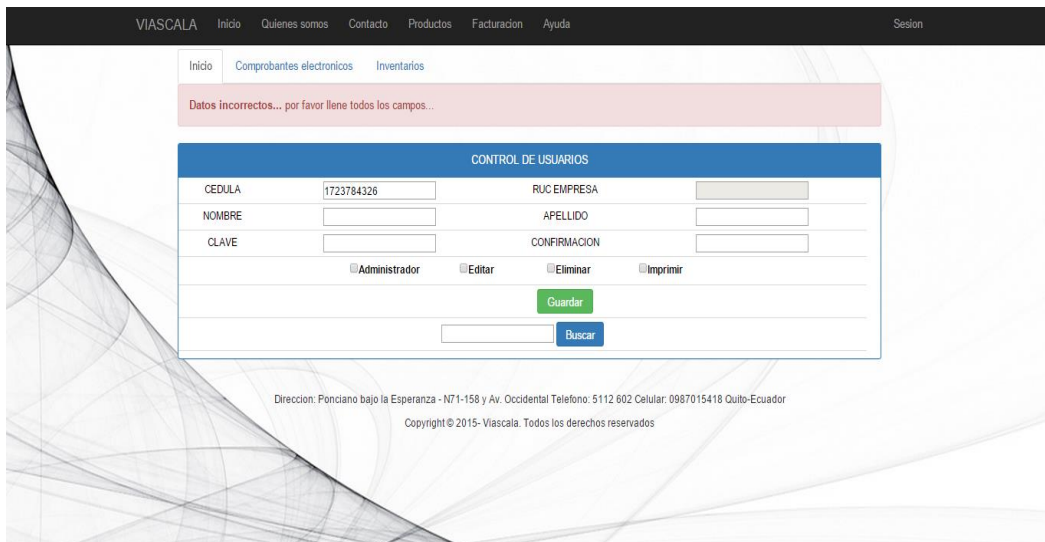
☐ Administrador ☐ Editar ☐ Eliminar ☐ Imprimir

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 56: Pagina de Control de Usuarios

Fuente: Maycol Flores

Es necesario llenar todos los campos para poder crear el usuario, caso contrario nos desplegara el mensaje de error.



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

Datos incorrectos... por favor llene todos los campos...

CONTROL DE USUARIOS

CEDULA RUC EMPRESA

NOMBRE APELLIDO

CLAVE CONFIRMACION

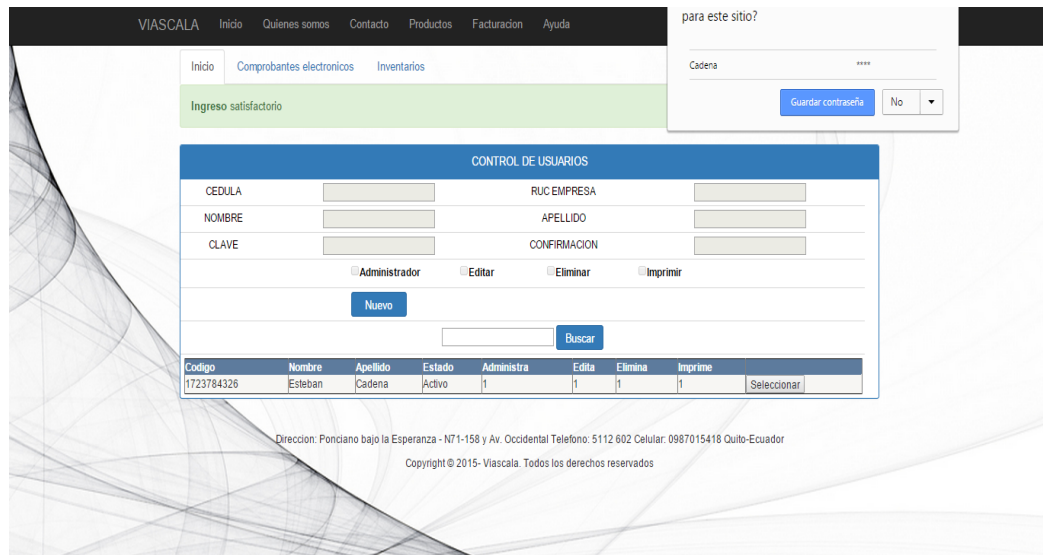
☐ Administrador ☐ Editar ☐ Eliminar ☐ Imprimir

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 57: Pagina de Control de Usuarios. Llenar todos los campos

Fuente: Maycol Flores

Una vez llenos todos los campos dar clic en guardar para crear el usuario en el sistema. A su vez si todo es correcto desplegara el mensaje de ingreso satisfactorio.



para este sitio?

Cadena ****

Guardar contraseña No

CONTROL DE USUARIOS

CEDULA RUC EMPRESA

NOMBRE APELLIDO

CLAVE CONFIRMACION

☐ Administrador ☐ Editar ☐ Eliminar ☐ Imprimir

Nuevo

Buscar

Codigo	Nombre	Apellido	Estado	Administra	Edita	Elimina	Imprime	
1723784326	Esteban	Cadena	Activo	1	1	1	1	Seleccionar

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador

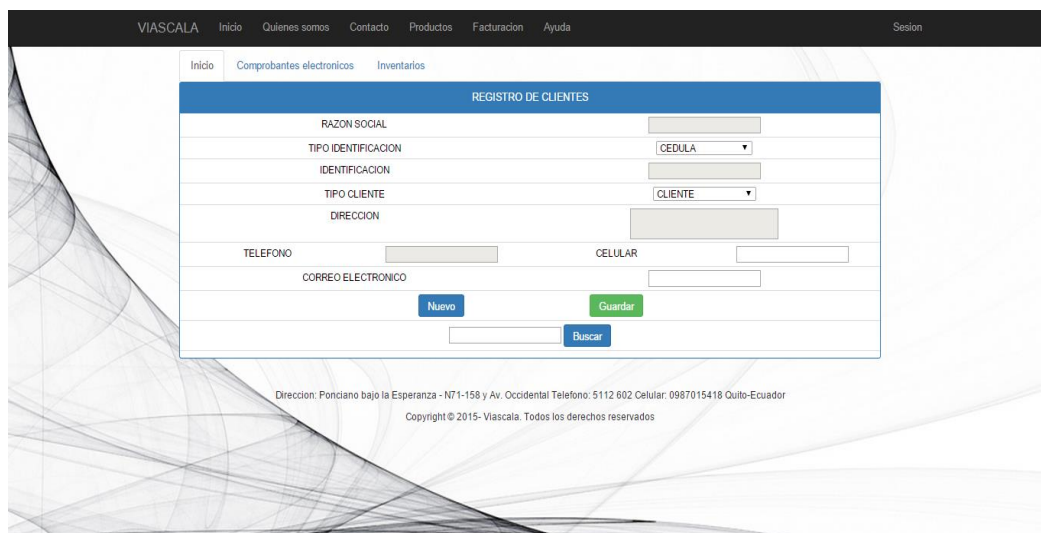
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 58: Control de usuarios. Ingreso Satisfactorio.

Fuente: Maycol Flores

2.5 Página de Clientes:

Esta opción nos permite ingresar al registro de clientes del sistema para registrar datos generales de los mismos el cual será usado al momento de generar facturas de una manera fácil y rápida.



VIASCALA Inicio Quienes somos Contacto Productos Facturacion Ayuda Sesion

Inicio Comprobantes electronicos Inventarios

REGISTRO DE CLIENTES

RAZON SOCIAL

TIPO IDENTIFICACION CEDULA

IDENTIFICACION

TIPO CLIENTE CLIENTE

DIRECCION

TELEFONO CELULAR

CORREO ELECTRONICO

Nuevo Guardar

Buscar

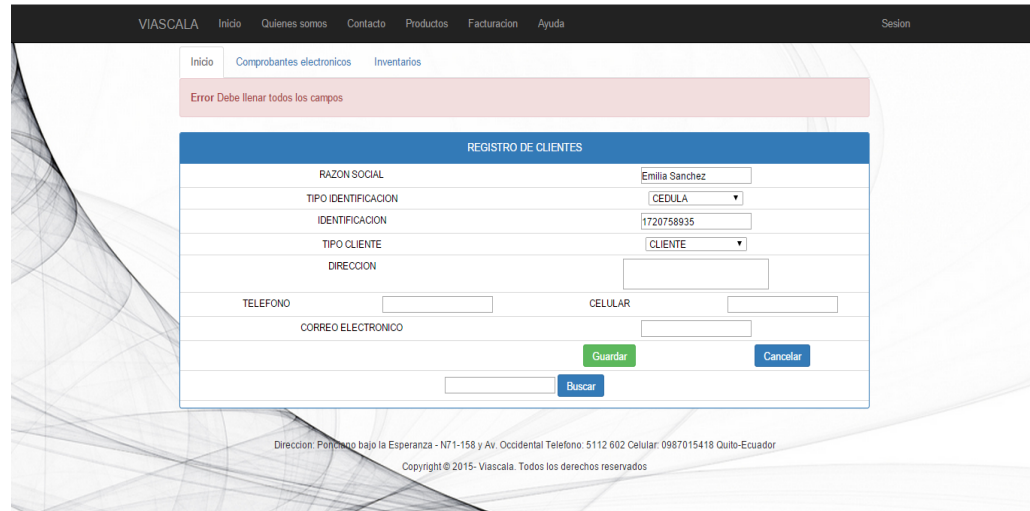
Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador

Copyright © 2015- Viascala. Todos los derechos reservados

Figura 59: Registro de Clientes.

Fuente: Maycol Flores

Es necesario llenar todos los campos para poder registrar los clientes en el sistema, caso contrario nos desplegará el mensaje de error.



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

Error Debe llenar todos los campos

REGISTRO DE CLIENTES

RAZON SOCIAL: Emilia Sanchez

TIPO IDENTIFICACION: CEDULA

IDENTIFICACION: 1720758935

TIPO CLIENTE: CLIENTE

DIRECCION:

TELEFONO: CELULAR:

CORREO ELECTRONICO:

Guardar Cancelar

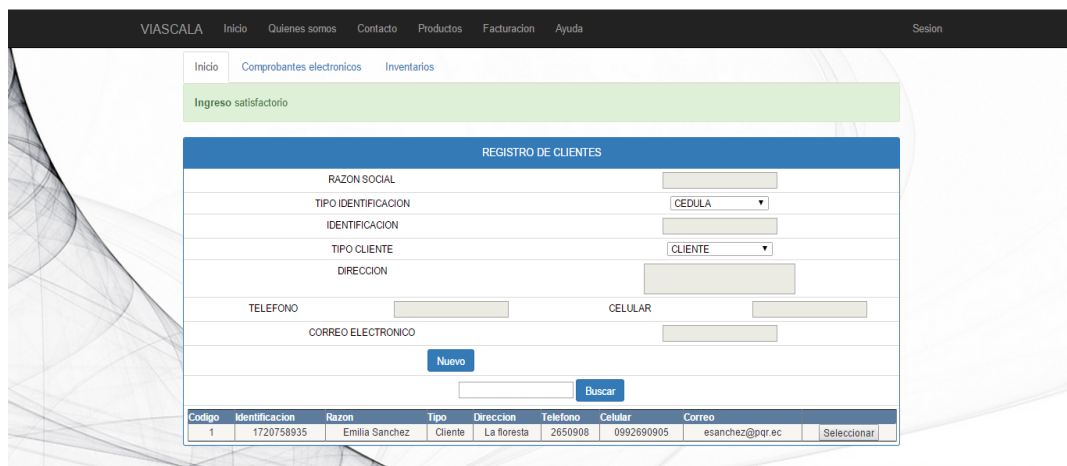
Buscar

Dirección: Pichincha bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 60: Registro de Clientes. Llenar todos los campos.

Fuente: Maycol Flores

Una vez llenos todos los campos dar clic en guardar para registrar los clientes en el sistema. A su vez si todo es correcto se desplegará el mensaje de ingreso satisfactorio.



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

Ingreso satisfactorio

REGISTRO DE CLIENTES

RAZON SOCIAL:

TIPO IDENTIFICACION: CEDULA

IDENTIFICACION:

TIPO CLIENTE: CLIENTE

DIRECCION:

TELEFONO: CELULAR:

CORREO ELECTRONICO:

Nuevo

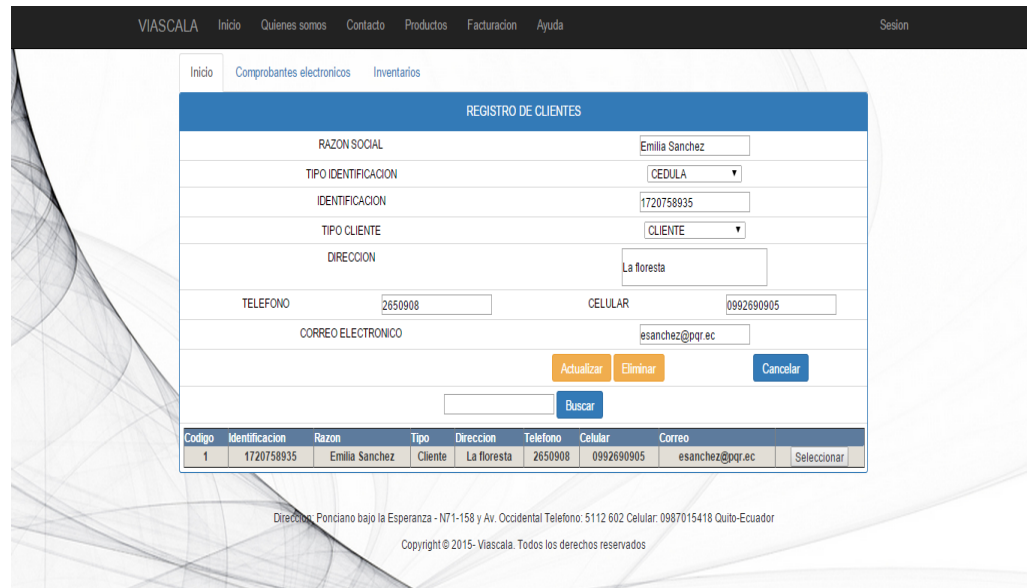
Buscar

Código	Identificación	Razón	Tipo	Dirección	Teléfono	Celular	Correo	
1	1720758935	Emilia Sanchez	Cliente	La floresta	2650908	0992690905	esanchez@pqr.ec	Seleccionar

Figura 61: Registro de Clientes. Ingreso Satisfactorio

Fuente: Maycol Flores

Se puede actualizar y eliminar la información de acuerdo a como el usuario lo requiera.



Código	Identificación	Razon	Tipo	Dirección	Teléfono	Celular	Correo	
1	1720758935	Emilia Sanchez	Cliente	La floresta	2650908	0992690905	esanchez@pqr.ec	Seleccionar

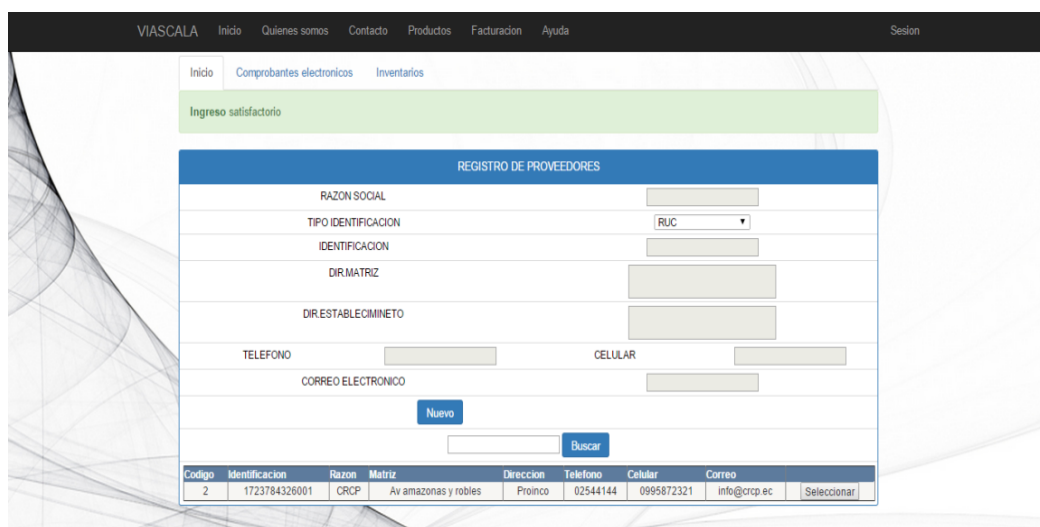
Dirección: Pondano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 62: Registro de Clientes. Actualizar y eliminar la información.

Fuente: Maycol Flores

2.6 Página de Proveedores:

Esta opción nos permite ingresar al registro de proveedores al sistema para registrar datos generales de los mismos el cual será usado al momento de generar facturas y hacer retenciones de una manera fácil y rápida.

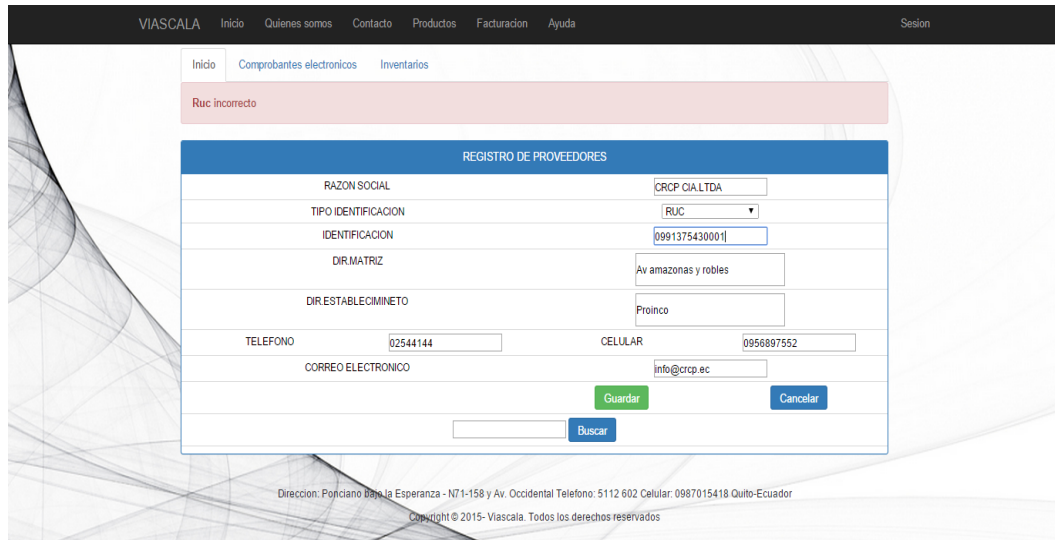


Código	Identificación	Razon	Matriz	Dirección	Teléfono	Celular	Correo	
2	1723784326001	CRCP	Av amazonas y robles	Proinco	02544144	0995872321	info@crp.ec	Seleccionar

Figura 63: Registro de Proveedores

Fuente: Maycol Flores

Es necesario llenar todos los campos con información correcta para poder registrar los proveedores en el sistema, caso contrario nos desplegará el mensaje de error.



The screenshot shows the 'Registro de Proveedores' form with the following data entered:

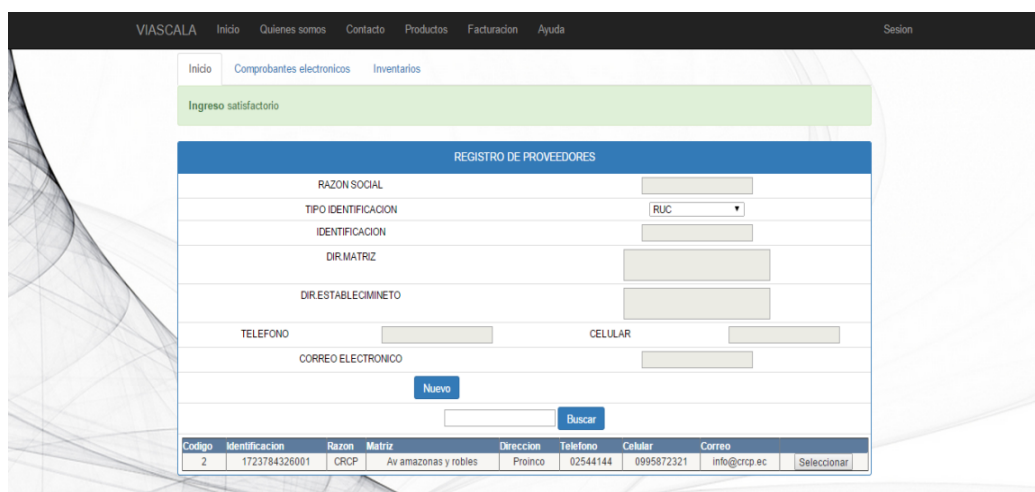
RAZON SOCIAL	CRCP CIA LTDA
TIPO IDENTIFICACION	RUC
IDENTIFICACION	0991375430001
DIR.MATRIZ	Av amazonas y robles
DIRESTABLECIMIENTO	Proinco
TELEFONO	02544144
CELULAR	0956897552
CORREO ELECTRONICO	info@crp.ec

A red error message 'Ruc incorrecto' is displayed at the top. The form includes 'Guardar', 'Cancelar', and 'Buscar' buttons. At the bottom, contact information and a copyright notice are visible.

Figura 64: Registro de Proveedores. Datos incorrectos.

Fuente: Maycol Flores

Una vez llenos todos los campos dar clic en guardar para registrar los proveedores en el sistema. A su vez si todo es correcto se desplegará el mensaje de ingreso satisfactorio.



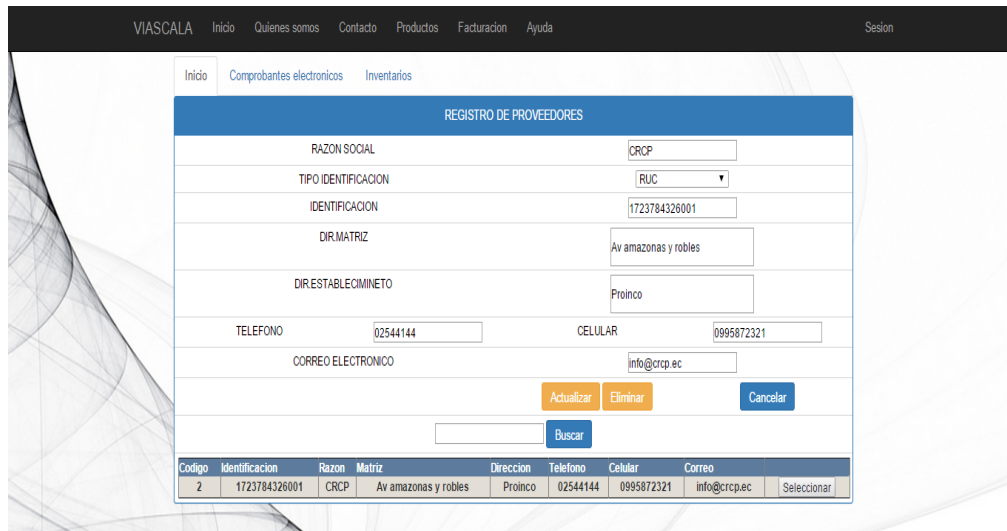
The screenshot shows the 'Registro de Proveedores' form with the same data as Figure 64. A green success message 'Ingreso satisfactorio' is displayed at the top. The form includes 'Nuevo' and 'Buscar' buttons. Below the form, a table lists the registered provider:

Codigo	Identificacion	Razon	Matriz	Direccion	Telefono	Celular	Correo	
2	1723784326001	CRCP	Av amazonas y robles	Proinco	02544144	0995872321	info@crp.ec	Seleccionar

Figura 65: Registro de Proveedores. Ingreso Satisfactorio.

Fuente: Maycol Flores

Se puede actualizar y eliminar la información de acuerdo a como el usuario lo requiera.



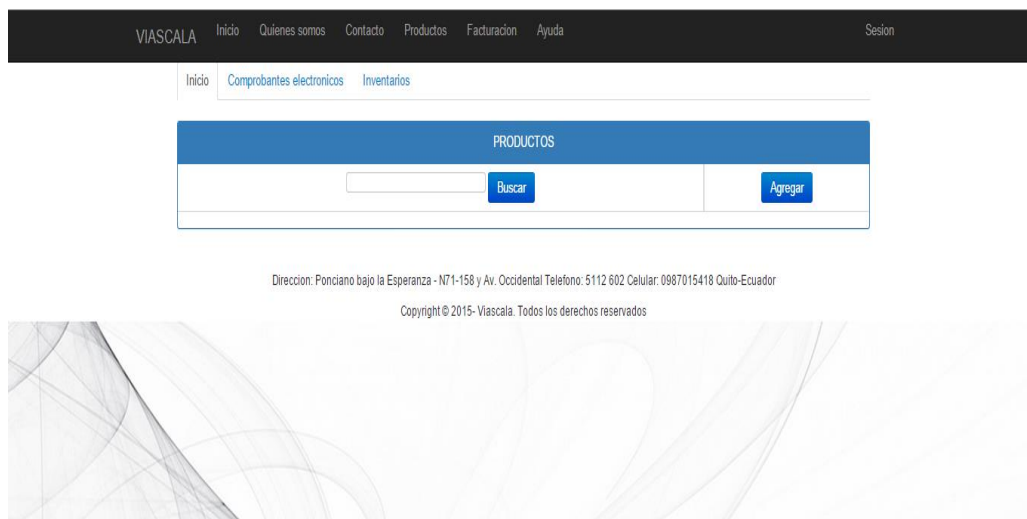
Código	Identificación	Razón	Matriz	Dirección	Teléfono	Celular	Correo	
2	1723784326001	CRCP	Av amazonas y robles	Proinco	02544144	0995872321	info@crp.ec	Seleccionar

Figura 66: Registro de Proveedores. Actualizar y eliminar información.

Fuente: Maycol Flores

2.7 Página de Productos (inventarios):

Esta opción nos permite ingresar productos al sistema para ser visualizados en el stock a la hora de vender y generar facturas de una manera fácil y rápida.



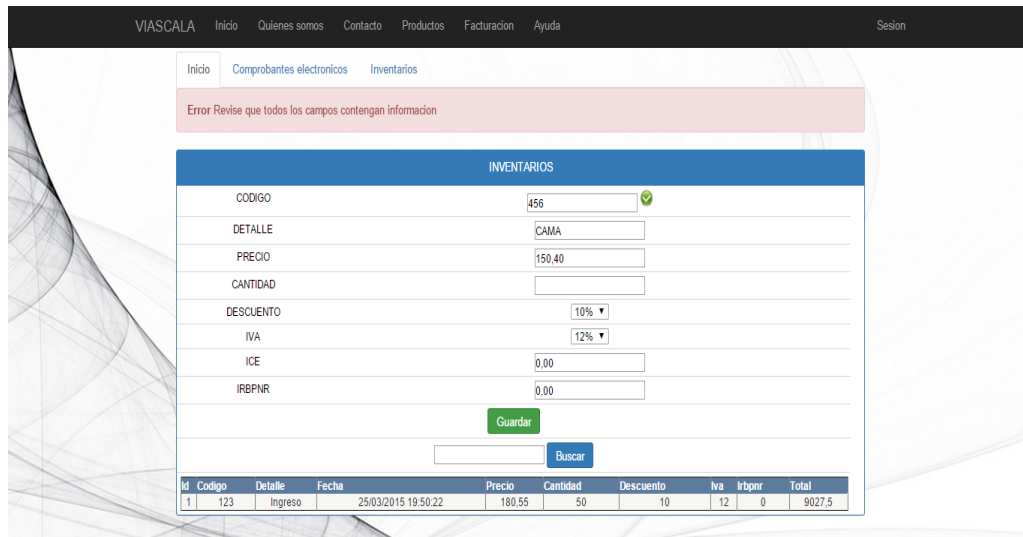
Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador

Copyright © 2015- Viascala. Todos los derechos reservados

Figura 67: Pagina de Productos.

Fuente: Maycol Flores

Es necesario llenar todos los campos para poder registrar los productos en el sistema, en caso de que el producto no este registrado, si ya está registrado se habilita el campo solo para ingresar la cantidad y automáticamente se sumara al stock.



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

Error Revise que todos los campos contengan información

INVENTARIOS

CODIGO: 456 ✓

DETALLE: CAMA

PRECIO: 150.40

CANTIDAD:

DESCUENTO: 10% ▼

IVA: 12% ▼

ICE: 0.00

IRBPNR: 0.00

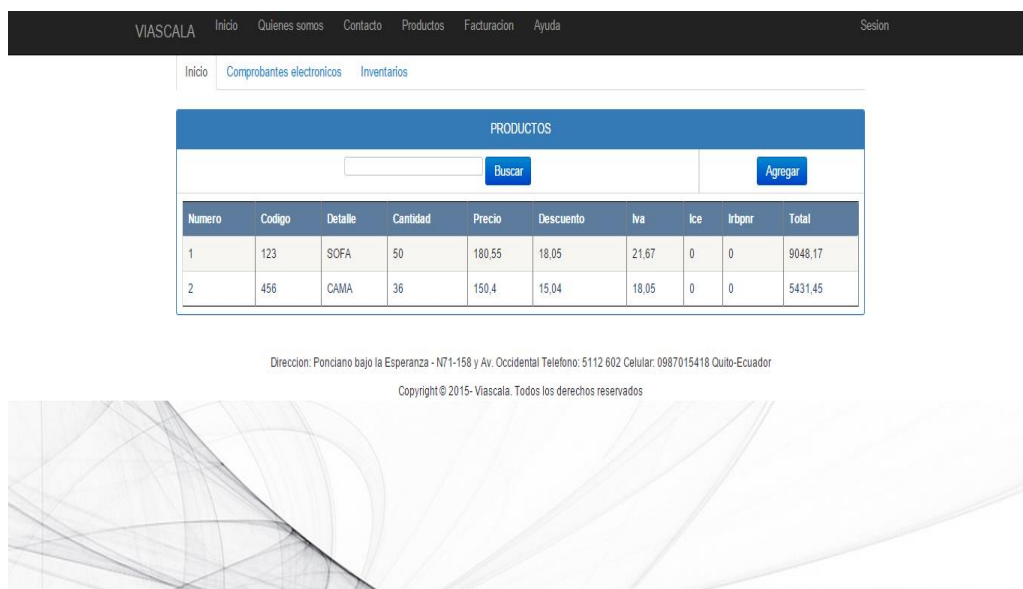
Guardar

Buscar

Id	Codigo	Detalle	Fecha	Precio	Cantidad	Descuento	Iva	Irbpnr	Total
1	123	Ingreso	25/03/2015 19:50:22	180.55	50	10	12	0	9027.5

Figura 68: Pagina Productos. Llenar todos los campos.

Fuente: Maycol Flores



VIASCALA Inicio Quiénes somos Contacto Productos Facturación Ayuda Sesión

Inicio Comprobantes electrónicos Inventarios

PRODUCTOS

Buscar Agregar

Numero	Codigo	Detalle	Cantidad	Precio	Descuento	Iva	Ice	Irbpnr	Total
1	123	SOFA	50	180.55	18.05	21.67	0	0	9048.17
2	456	CAMA	36	150.4	15.04	18.05	0	0	5431.45

Dirección: Ponciano bajo la Esperanza - N71-158 y Av. Occidental Teléfono: 5112 602 Celular: 0987015418 Quito-Ecuador

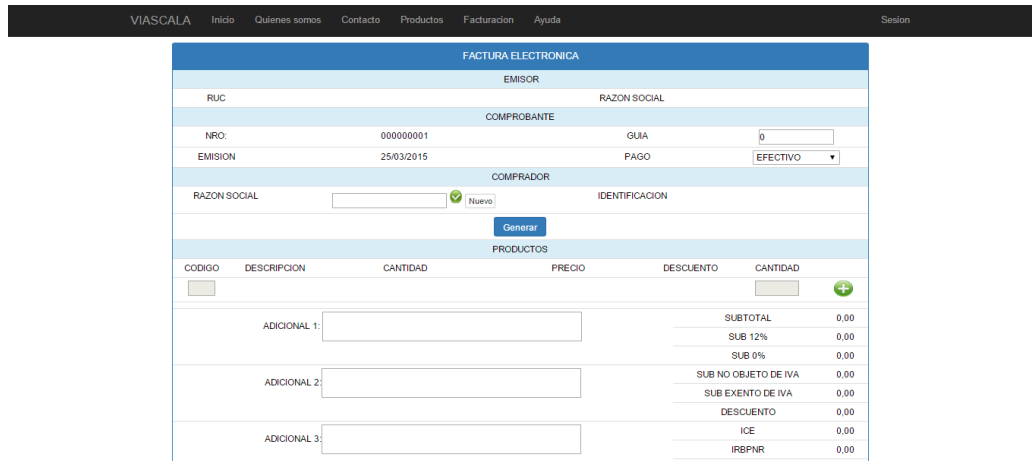
Copyright © 2015- Viascala. Todos los derechos reservados

Figura 69: Pagina Productos. Stock de Productos

Fuente: Maycol Flores

2.8 Página de Factura Electrónica:

Esta opción nos permite generar facturas de una manera fácil y rápida, ya sea a clientes o proveedores, cumpliendo con todos los procesos de facturación como la nota de crédito, nota de débito y retenciones, generando a su vez sus comprobantes electrónicos como son el XML, PDF Y FIRMA DIGITAL.



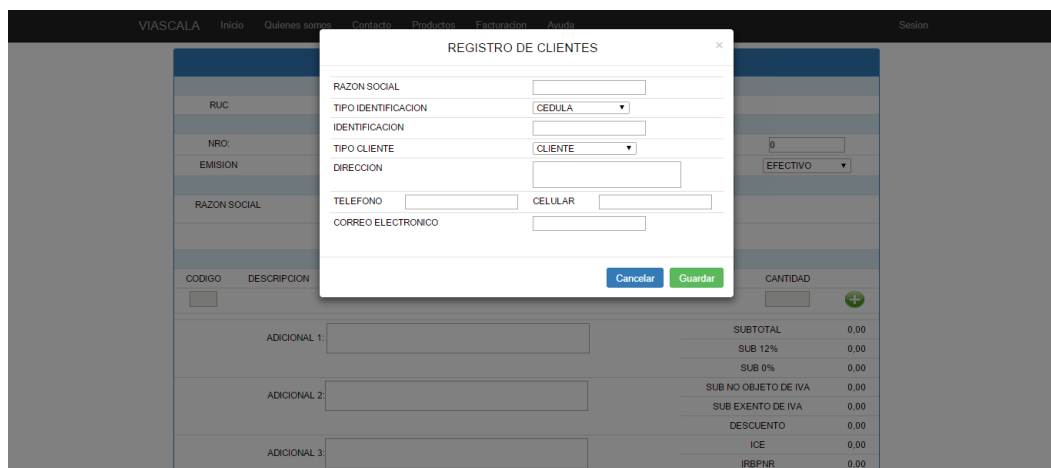
CODIGO	DESCRIPCION	CANTIDAD	PRECIO	DESCUENTO	CANTIDAD
	ADICIONAL 1:				
	ADICIONAL 2:				
	ADICIONAL 3:				

SUBTOTAL	0.00
SUB 12%	0.00
SUB 0%	0.00
SUB NO OBJETO DE IVA	0.00
SUB EXENTO DE IVA	0.00
DESCUENTO	0.00
ICE	0.00
IRBPNR	0.00

Figura 70: Pagina Factura Electrónica.

Fuente: Maycol Flores

Para generar una factura es necesaria la información del cliente, en caso de que el cliente no este registrado en el sistema se habilitara una pantalla modal en la página de la factura para poder registrar el mismo desde ahí sin tener que ir al registro de clientes.



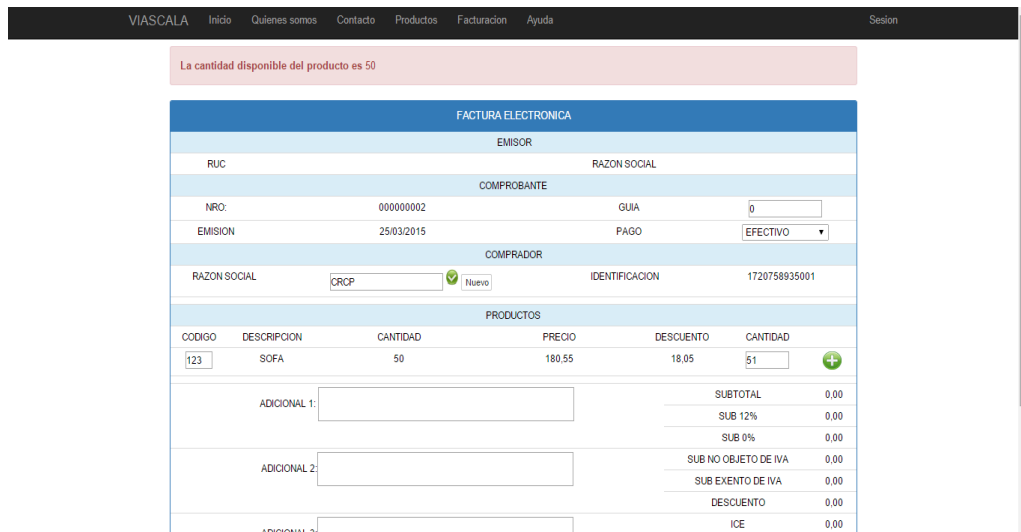
CODIGO	DESCRIPCION	CANTIDAD
	ADICIONAL 1:	
	ADICIONAL 2:	
	ADICIONAL 3:	

SUBTOTAL	0.00
SUB 12%	0.00
SUB 0%	0.00
SUB NO OBJETO DE IVA	0.00
SUB EXENTO DE IVA	0.00
DESCUENTO	0.00
ICE	0.00
IRBPNR	0.00

Figura 71: Pagina Factura Electrónica. Registro de Clientes

Fuente: Maycol Flores

Una vez ingresado los datos del cliente en la factura registramos los productos a vender teniendo en cuenta el stock de productos.



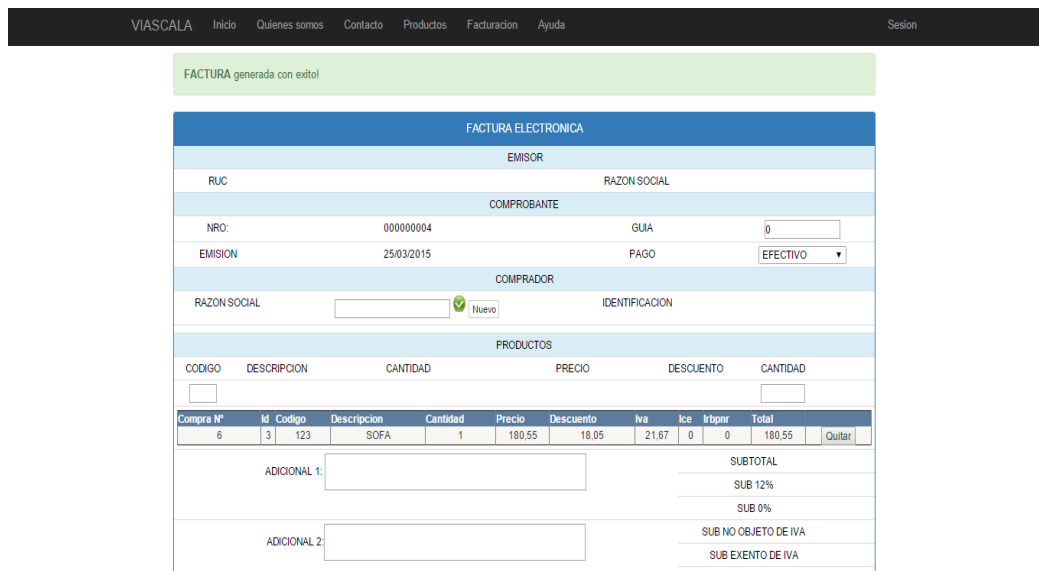
La cantidad disponible del producto es 50

FACTURA ELECTRÓNICA						
EMISOR						
RUC		RAZON SOCIAL				
COMPROBANTE						
NRO.	00000002	GUIA		0		
EMISION	25/03/2015	PAGO		EFECTIVO		
COMPRADOR						
RAZON SOCIAL		IDENTIFICACION		1720758935001		
PRODUCTOS						
CODIGO	DESCRIPCION	CANTIDAD	PRECIO	DESCUENTO	CANTIDAD	
123	SOFA	50	180.55	18.05	51	
ADICIONAL 1:					SUBTOTAL	0.00
ADICIONAL 2:					SUB 12%	0.00
ADICIONAL 3:					SUB 0%	0.00
					SUB NO OBJETO DE IVA	0.00
					SUB EXENTO DE IVA	0.00
					DESCUENTO	0.00
					ICE	0.00

Figura 72: Pagina Factura Electrónica. Verificando stock

Fuente: Maycol Flores

Una vez llenado todos los datos del cliente y productos en venta procedemos a generar la factura.



FACTURA generada con éxito!

FACTURA ELECTRÓNICA											
EMISOR											
RUC		RAZON SOCIAL									
COMPROBANTE											
NRO.	00000004	GUIA		0							
EMISION	25/03/2015	PAGO		EFECTIVO							
COMPRADOR											
RAZON SOCIAL		IDENTIFICACION									
PRODUCTOS											
CODIGO	DESCRIPCION	CANTIDAD	PRECIO	DESCUENTO	CANTIDAD						
Compra N°	Id	Codigo	Descripcion	Cantidad	Precio	Descuento	Iva	Ice	Irbpnr	Total	
6	3	123	SOFA	1	180.55	18.05	21.67	0	0	180.55	Quitar
ADICIONAL 1:					SUBTOTAL						
ADICIONAL 2:					SUB 12%						
					SUB 0%						
					SUB NO OBJETO DE IVA						
					SUB EXENTO DE IVA						

Figura 73: Pagina Factura Electrónica. Generada con Éxito

Fuente: Maycol Flores

Una vez generada la factura electrónica nos dirigimos a la carpeta raíz del sistema y verificaremos los comprobantes electrónicos como son el XML Y PDF.

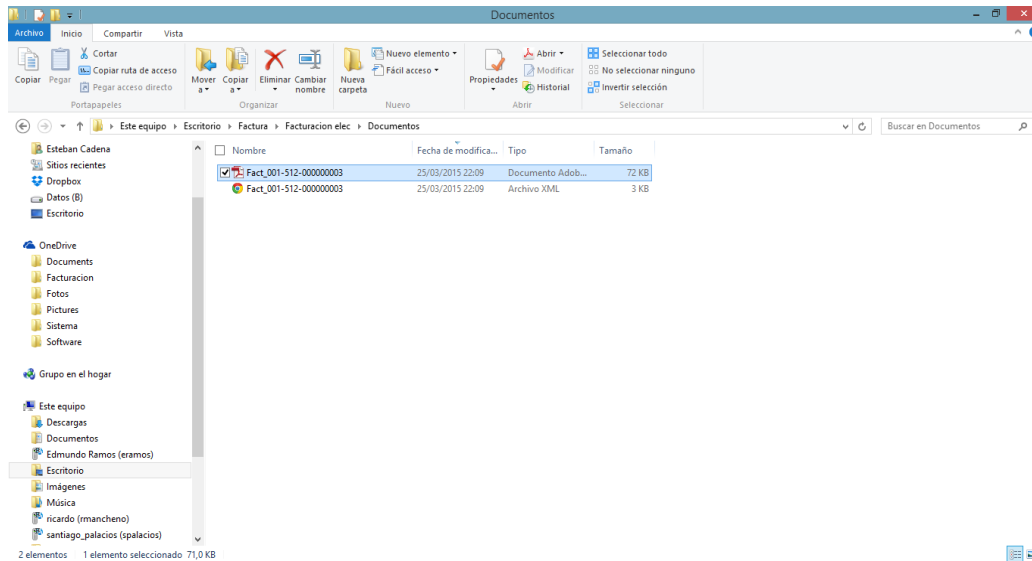


Figura 74: Carpeta raíz del sistema. Comprobantes Electrónicos

Fuente: Maycol Flores



VIASCALA S.A.
VIASCALA
DIRECCIÓN SUCURSAL:
CONTRIBUYENTE ESPECIAL No:
OBLIGADO A LLEVAR CONTABILIDAD: SI

RUC: 253201150120000000020102030416
RUC/CE: 1720758935001

Cod	Cant	Descripción	Precio Unitario	Desc 1%	Desc 2%	Desc 5	IVA	Precio Total
1	1	SOFA	180,00				21,67	180,00

INFORMACIÓN ADICIONAL
Dirección: Av Amazonas
Teléfono: 02544144
Email: info@crep.ec
Usuario: 1720758935001
Clave: 1720758935001

SUBTOTAL IVA 12% 180,00
SUBTOTAL IVA 0% 0,00
SUBTOTAL NO OBJ IVA 0,00
SUBTOTAL EXENTO IVA 0,00
TOTAL DESCUENTO 18,00
IVA 12% 21,67
VALOR TOTAL 184,17

Figura 75: Comprobantes Electrónicos. PDF

Fuente: Maycol Flores

```
<?xml version="1.0" encoding="UTF-8"?>
<factura id="comprobante" version="1.0.0">
  <infoTributaria>
    <ambiente/>
    <tipoEmision/>
    <razonSocial/>
    <ruc/>
    <claveAcceso>250320150120000000020102030418</claveAcceso>
    <codDoc>01</codDoc>
    <estab/>
    <ptoEmi/>
    <secuencial>000000004</secuencial>
    <dirMatriz/>
  </infoTributaria>
  <infoFactura>
    <fechaEmision>25/03/2015</fechaEmision>
    <dirEstablecimiento/>
    <obligadoContabilidad>Si</obligadoContabilidad>
    <tipoIdentificacionComprador>04</tipoIdentificacionComprador>
    <razonSocialComprador>CRCP</razonSocialComprador>
    <identificacionComprador>1720758935001</identificacionComprador>
    <totalSinImpuestos>180,55</totalSinImpuestos>
    <totalDescuento>18,05</totalDescuento>
  </infoFactura>
  <totalImpuestos>
    <codigo>2</codigo>
    <codigoPorcentaje>2</codigoPorcentaje>
    <baseImponible>180,55</baseImponible>
    <valor>21,67</valor>
  </totalImpuestos>
  <totalConImpuestos>
    <propina>0,00</propina>
    <importeTotal>184,17</importeTotal>
    <moneda>Dolar</moneda>
  </totalConImpuestos>
  <infoAdicional>
    <campoAdicional nombre="email">info@crp.ec</campoAdicional>
    <campoAdicional nombre="usuario">1720758935001</campoAdicional>
    <campoAdicional nombre="clave">1720758935001</campoAdicional>
  </infoAdicional>
</factura>
```

Figura 76: Comprobantes Electrónicos. XML

Fuente: Maycol Flores

Una vez generada facturas a proveedores podemos realizar las correspondientes retenciones en caso de ser necesarias o requeridas por el usuario.

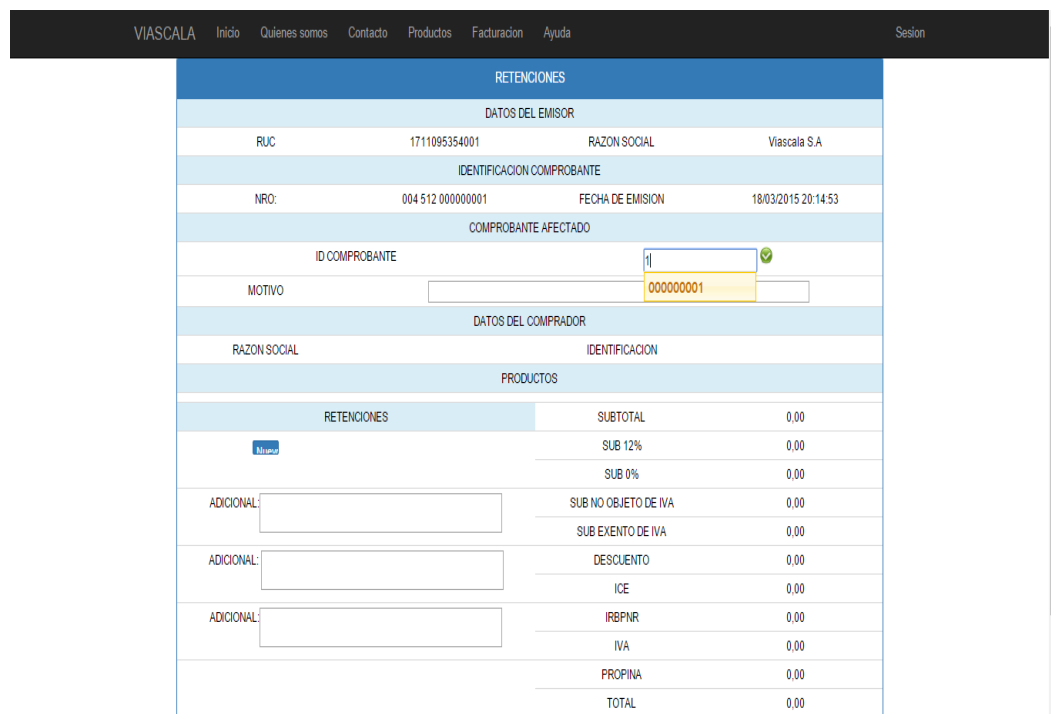
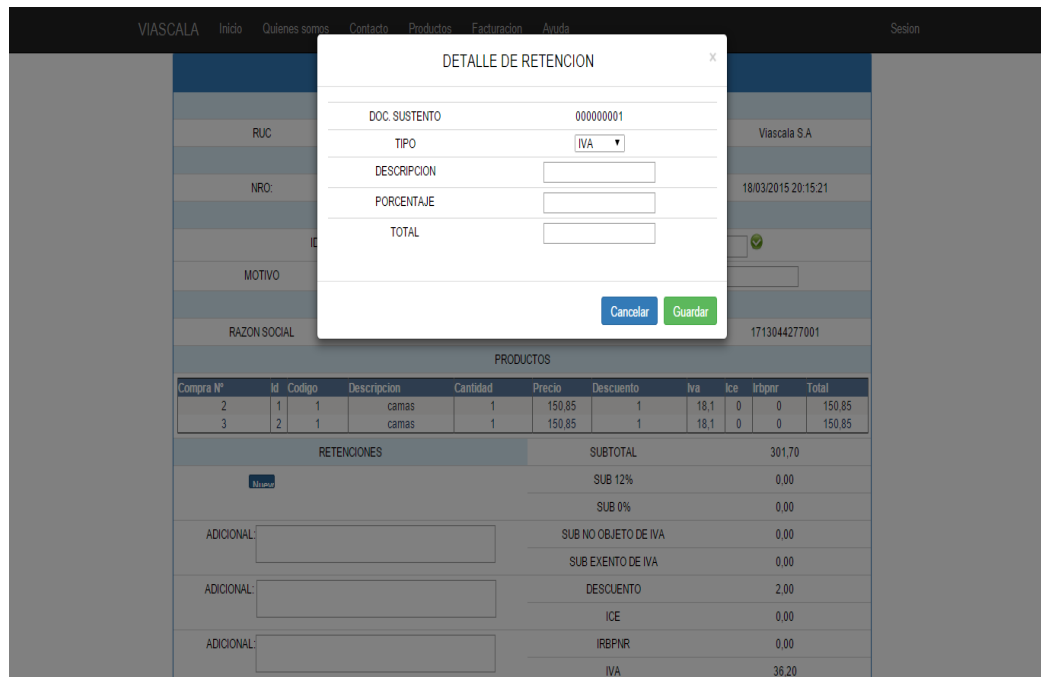


Figura 77: Página de Retenciones. Retención a Proveedores.

Fuente: Maycol Flores



DETALLE DE RETENCION

DOC. SUSTENTO: 000000001

TIPO: IVA

DESCRIPCION:

PORCENTAJE:

TOTAL:

Cancelar Guardar

PRODUCTOS

Compra N°	Id	Codigo	Descripcion	Cantidad	Precio	Descuento	Iva	Ice	Irtpnr	Total
2	1	1	camas	1	150.85	1	18,1	0	0	150.85
3	2	1	camas	1	150.85	1	18,1	0	0	150.85

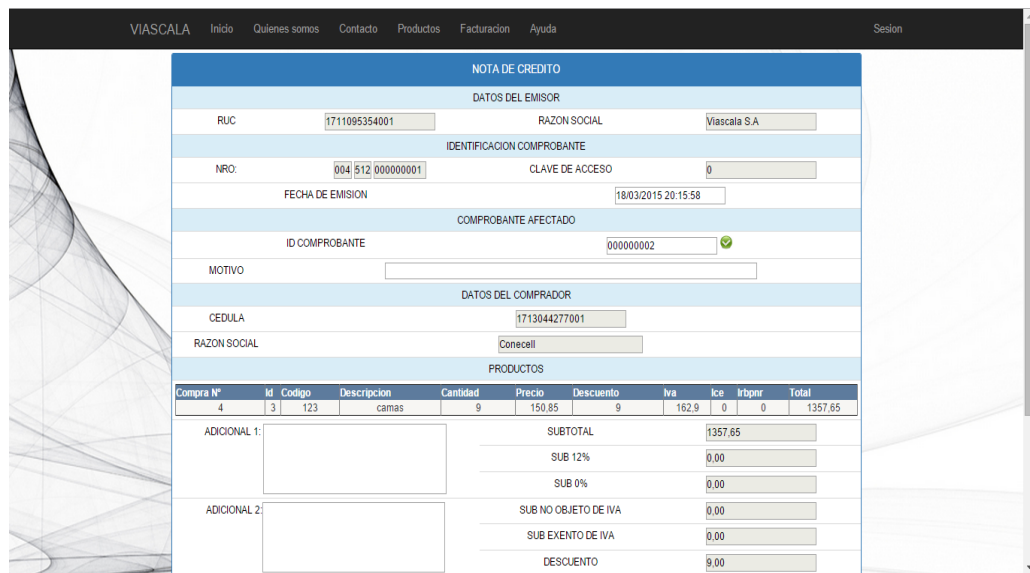
RETENCIONES

	SUBTOTAL	
	301,70	
	SUB 12%	0,00
	SUB 0%	0,00
ADICIONAL:	SUB NO OBJETO DE IVA	0,00
ADICIONAL:	SUB EXENTO DE IVA	0,00
ADICIONAL:	DESCUENTO	2,00
ADICIONAL:	ICE	0,00
ADICIONAL:	IRBPNR	0,00
ADICIONAL:	IVA	36,20

Figura 78: Página de Retenciones. Detalle Retención.

Fuente: Maycol Flores

Se pueden realizar devoluciones en caso de que el cliente lo requiera necesario.



NOTA DE CREDITO

DATOS DEL EMISOR

RUC: 1711095354001 RAZON SOCIAL: Viascala S.A.

IDENTIFICACION COMPROBANTE

NRO: 004 512 000000001 CLAVE DE ACCESO: 0

FECHA DE EMISION: 18/03/2015 20:15:58

COMPROBANTE AFECTADO

ID COMPROBANTE: 000000002

MOTIVO:

DATOS DEL COMPRADOR

CEDULA: 1713044277001 RAZON SOCIAL: Conecell

PRODUCTOS

Compra N°	Id	Codigo	Descripcion	Cantidad	Precio	Descuento	Iva	Ice	Irtpnr	Total
4	3	123	camas	9	150.85	9	162,9	0	0	1357.65

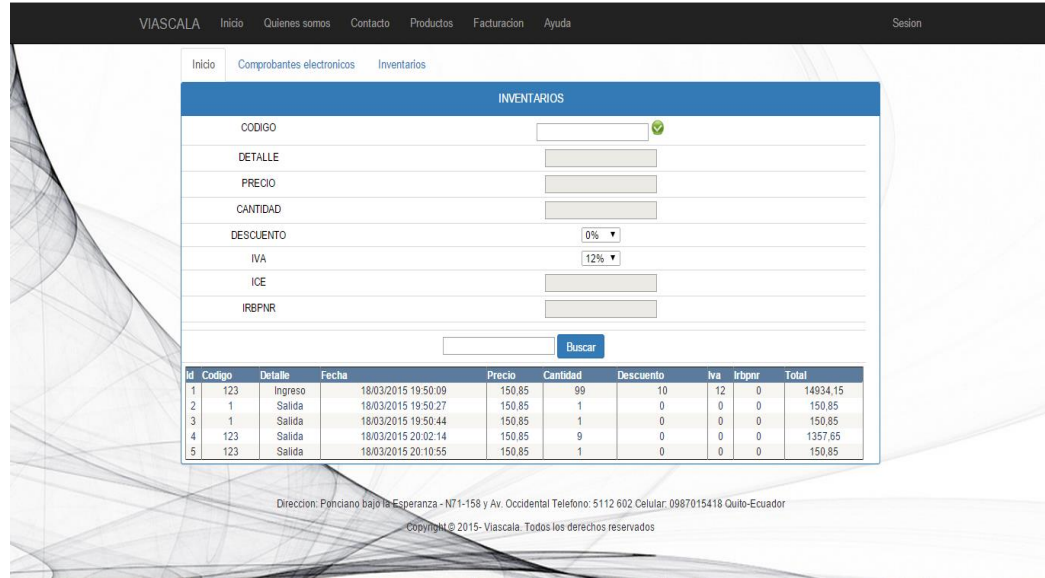
ADICIONAL 1:

	SUBTOTAL	
	1357,65	
	SUB 12%	0,00
	SUB 0%	0,00
ADICIONAL 2:	SUB NO OBJETO DE IVA	0,00
ADICIONAL 2:	SUB EXENTO DE IVA	0,00
ADICIONAL 2:	DESCUENTO	9,00

Figura 79: Página de Devoluciones. Nota de Crédito.

Fuente: Maycol Flores

Luego de hacer una devolución de productos podemos observar que los productos devueltos son sumados nuevamente al stock para tener un mejor control de productos y llevar una mejor contabilidad.

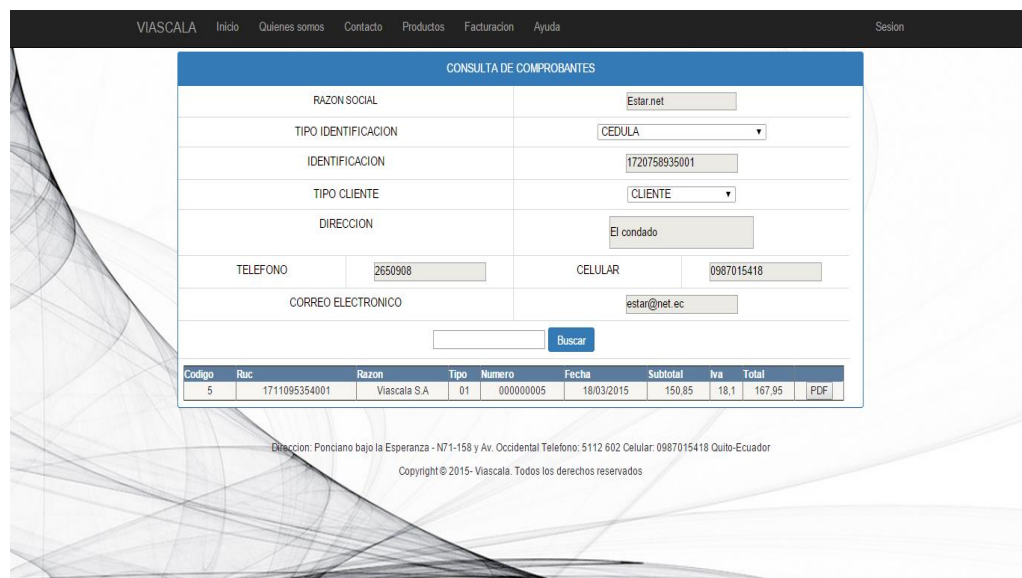


ID	Codigo	Detalle	Fecha	Precio	Cantidad	Descuento	Iva	Irbpnr	Total
1	123	Ingreso	18/03/2015 19:50:09	150.85	99	10	12	0	14934.15
2	1	Salida	18/03/2015 19:50:27	150.85	1	0	0	0	150.85
3	1	Salida	18/03/2015 19:50:44	150.85	1	0	0	0	150.85
4	123	Salida	18/03/2015 20:02:14	150.85	9	0	0	0	1357.65
5	123	Salida	18/03/2015 20:10:55	150.85	1	0	0	0	150.85

Figura 80: Página de Inventarios. Entradas y Salidas

Fuente: Maycol Flores

Los clientes tendrán acceso al sistema solamente para consultar su comprobante electrónico y posterior a eso podrá descargarse o imprimir los documentos.



Codigo	Ruc	Razon	Tipo	Numero	Fecha	Subtotal	Iva	Total
5	1711095354001	Viascala S.A	01	000000005	18/03/2015	150.85	18.1	167.95

Figura 81: Pagina Consulta de Comprobantes.

Fuente: Maycol Flores

A.07 Manual Técnico

1. Introducción

La finalidad de este manual técnico es la de proporcionar al lector la lógica con la que se desarrolló la aplicación, la cual se sabe que es propia de cada programador, por lo que se considera necesario ser documentada.

2. Objetivo

Proporcionar una guía para el lector, del desarrollo de interfaz y también de la lógica con la que el sistema funciona.

3. Contenido

3.1 Declarar entidades

Código utilizado para la creación de métodos get y set del sistema.

```
public class clienteENT
{
//metodos de atributos
private int id_cliente;
public int iid_cliente
{
get { return id_cliente;
set { id_cliente = value; }
}
//metodos de atributos
private string tipo_identificacion;
public string itipo_identificacion
{
get { return tipo_identificacion; }
set { tipo_identificacion = value; }
}
//metodos de atributos
private string ci_cliente;
public string ici_cliente
{
get { return ci_cliente; }
set { ci_cliente = value; }
}
//metodos de atributos
private string razon_cliente;
public string irazon_cliente
{
```

```
get { return razon_cliente; }
set { razon_cliente = value; }
}
//metodos de atributos
private string tipo_cliente;
public string itipo_cliente
{
    get { return tipo_cliente; }
    set { tipo_cliente = value; }
}
//metodos de atributos
private string direccion_cliente;
public string idireccion_cliente
{
    get { return direccion_cliente; }
    set { direccion_cliente = value; }
}
//métodos de atributos
private string telefono_cliente;
public string itelefono_cliente
{
    get { return telefono_cliente; }
    set { telefono_cliente = value; }
}
//metodos de atributos
private string celular_cliente;
public string icelular_cliente
{
    get { return celular_cliente; }
    set { celular_cliente = value; }
}
//metodos de atributos
private string estado_cliente;
public string iestado_cliente
{
    get { return estado_cliente; }
    set { estado_cliente = value; }
}
//metodos de atributos
private string correo_cliente;
public string icorreo_cliente
{
    get { return correo_cliente; }
    set { correo_cliente = value; }
}
//metodos de atributos
private int ci_usuario;
public int ici_usuario
{
    get { return ci_usuario; }
```



```
set { ci_usuario = value; }  
}  
}
```

3.2 Insertar cliente

Código utilizado para el registro de clientes en el sistema utilizando procedimientos almacenados.

```
public Boolean insertarCliente(clienteENT usuario)  
{  
    SqlConnection cnn = new SqlConnection(conexionD.conexionC());  
    cnn.Open();  
    SqlCommand cmd = new SqlCommand("inscliente", cnn);  
    cmd.Parameters.Add("@razon_cliente", SqlDbType.NVarChar, 50).Value =  
        usuario.irazon_cliente;  
    cmd.Parameters.Add("@tipo_identificacion", SqlDbType.NVarChar, 30).Value =  
        usuario.itipo_identificacion;  
    cmd.Parameters.Add("@ci_cliente", SqlDbType.NVarChar, 15).Value =  
        usuario.ici_cliente;  
    cmd.Parameters.Add("@tipo_cliente", SqlDbType.NVarChar, 20).Value =  
        usuario.itipo_cliente;  
    cmd.Parameters.Add("@direccion_cliente", SqlDbType.NVarChar, 50).Value =  
        usuario.idireccion_cliente;  
    cmd.Parameters.Add("@telefono_cliente", SqlDbType.NVarChar, 15).Value =  
        usuario.itelefono_cliente;  
    cmd.Parameters.Add("@celular_cliente", SqlDbType.NVarChar, 15).Value =  
        usuario.icelular_cliente;  
    cmd.Parameters.Add("@estado_cliente", SqlDbType.NVarChar, 15).Value =  
        usuario.iestado_cliente;  
    cmd.Parameters.Add("@correo_cliente", SqlDbType.NVarChar, 30).Value =  
        usuario.icorreo_cliente;  
    cmd.Parameters.Add("@ci_usuario", SqlDbType.Int).Value =  
        Convert.ToInt32(usuario.ici_usuario);  
  
    cmd.CommandType = CommandType.StoredProcedure;  
    int exito = cmd.ExecuteNonQuery();  
    if (exito == 0)  
    {  
        return false;  
    }  
    else  
    {  
        return true;  
    }  
}
```

3.3 Actualizar clientes

Código utilizado para actualizar información de clientes en el sistema utilizando procedimientos almacenados.

```
public Boolean updateCliente(clienteENT usuario)
{
    SqlConnection cnn = new SqlConnection(conexionD.conexionC());
    cnn.Open();
    SqlCommand cmd = new SqlCommand("updcliente", cnn);
    cmd.Parameters.Add("@razon_cliente", SqlDbType.NVarChar, 50).Value =
        usuario.irazon_cliente;
    cmd.Parameters.Add("@tipo_identificacion", SqlDbType.NVarChar, 30).Value =
        usuario.itipo_identificacion;
    cmd.Parameters.Add("@ci_cliente", SqlDbType.NVarChar, 15).Value =
        usuario.ici_cliente;
    cmd.Parameters.Add("@tipo_cliente", SqlDbType.NVarChar, 20).Value =
        usuario.itipo_cliente;
    cmd.Parameters.Add("@direccion_cliente", SqlDbType.NVarChar, 50).Value =
        usuario.idireccion_cliente;
    cmd.Parameters.Add("@telefono_cliente", SqlDbType.NVarChar, 15).Value =
        usuario.itelefono_cliente;
    cmd.Parameters.Add("@celular_cliente", SqlDbType.NVarChar, 15).Value =
        usuario.icelular_cliente;
    cmd.Parameters.Add("@estado_cliente", SqlDbType.NVarChar, 15).Value =
        usuario.iestado_cliente;
    cmd.Parameters.Add("@correo_cliente", SqlDbType.NVarChar, 30).Value =
        usuario.icorreo_cliente;
    cmd.Parameters.Add("@ci_usuario", SqlDbType.Int).Value =
        Convert.ToInt32(usuario.ici_usuario);

    cmd.CommandType = CommandType.StoredProcedure;
    int exito = cmd.ExecuteNonQuery();
    if (exito == 0)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

3.4 Eliminar cliente

Código utilizado para eliminar información de clientes en el sistema utilizando procedimientos almacenados.

```
public Boolean eliminarCliente(clienteENT cliente)
{
    SqlConnection cnn = new SqlConnection(conexionD.conexionC());
    cnn.Open();
    SqlCommand cmd = new SqlCommand("delcliente", cnn);
    cmd.Parameters.Add("@ci_cliente", SqlDbType.Int).Value = cliente.ici_cliente;
    cmd.Parameters.Add("@estado_cliente", SqlDbType.NVarChar, 10).Value =
    cliente.iestado_cliente;

    cmd.CommandType = CommandType.StoredProcedure;
    int exito = cmd.ExecuteNonQuery();
    if (exito < 1)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

3.5 Listar cliente en grid view

Código utilizado para listar información de clientes en el sistema utilizando procedimientos almacenados.

```
public List<clienteENT> listarCliente(clienteENT cliente)
{
    List<clienteENT> ListaCliente = new List<clienteENT>();

    SqlConnection cnn = new SqlConnection(conexionD.conexionC());
    {
        cnn.Open();
        SqlCommand comando = new SqlCommand(string.Format("SELECT * FROM
        Cliente Where estado_cliente = '{0}'", "Activo"), cnn);
        SqlDataReader lector = comando.ExecuteReader();
        while (lector.Read())
        {

```

```
clienteENT clienteE = new clienteENT();
clienteE.iid_cliente = int.Parse(lector[0].ToString());
clienteE.irazon_cliente = lector[1].ToString();
clienteE.itipo_identificacion = lector[2].ToString();
clienteE.ici_cliente = lector[3].ToString();
clienteE.itipo_cliente = lector[4].ToString();
clienteE.idireccion_cliente = lector[5].ToString();
clienteE.itelefono_cliente = lector[6].ToString();
clienteE.icelular_cliente = lector[7].ToString();
clienteE.iestado_cliente = lector[8].ToString();
clienteE.icorreo_cliente = lector[9].ToString();
ListaCliente.Add(clienteE);
    }
cnn.Close();
}

return ListaCliente;
}
```

3.6 Capa de negocios

Código utilizado para declarar variables en el sistema utilizando procedimientos almacenados.

```
public class cliente
{
    clienteDAL clienteD = new clienteDAL();
    public Boolean insertarCliente(clienteENT cliente)
    {
        return clienteD.insertarCliente(cliente);
    }
    public List<clienteENT> listarCliente(clienteENT cliente)
    {
        return clienteD.listarCliente(cliente);
    }
    public Boolean updateCliente(clienteENT cliente)
    {
        return clienteD.updateCliente(cliente);
    }
    public Boolean eliminarCliente(clienteENT cliente)
    {
        return clienteD.eliminarCliente(cliente);
    }
}
```

3.7 Validación de campos

Código utilizado para validar campos en el sistema, para tener un mejor registro de información y mejor funcionamiento del sistema.

```
function validar_numeros(e) {
    tecla = (document.all) ? e.keyCode : e.which;
    if (tecla == 8) return true;
    patron =
    /[qwertyuiopasdfghjklñzxcvbnmQWERTYUIOPÑLKJHGFDSA ZXCVBNM]/;
    te = String.fromCharCode(tecla);
    if (patron.test(te)) {
        alert('DEBE INGRESAR SOLO NÚMEROS');
        return false;
    }
}

function validar_letras(e) {
    tecla = (document.all) ? e.keyCode : e.which;
    if (tecla == 8) return true;
    patron = /[0123456789]/;
    te = String.fromCharCode(tecla);
    if (patron.test(te)) {
        alert('DEBE INGRESAR SOLO LETRAS');
        return false;
    }
}

function alerta(a) {
    alert('REGISTRO GUARDADO !!');
}

function eliminar(a) {
    alert('REGISTRO ELIMINADO OK');
}

function mensaje(a) {
    alert('NO SELECCIONNO EL REGISTRO !!');
}
```

3.8 Autocomplete

Código utilizado para realizar búsquedas rápidas de información en el sistema.

```
private void autocomplete()
{
    string queryString = "SELECT * FROM Proveedor ORDER BY razon_proveedor";
    using (SqlConnection connection = BDComun.ObtnerCOnexion())
```

```
{
using (SqlCommand command = new SqlCommand(queryString, connection))
{
//connection.Open();

using (SqlDataReader reader = command.ExecuteReader())
{
while (reader.Read())
{

if (string.IsNullOrEmpty(SuggestionList))
{
SuggestionList += "\"" + reader["razon_proveedor"].ToString() + "\"";
}
else
{
SuggestionList += ", \"" + reader["razon_proveedor"].ToString() + "\"";
}
}
}
$(function () {
var availableTags = [ <%= SuggestionList %>];

$("#<%= txtNom.ClientID %>").autocomplete({
source: availableTags,
});
});
```

3.9 Operaciones automáticas de cálculo en comprobantes

Código utilizado para realizar los cálculos respectivos para generar una factura.

```
private void operaciones()
{
SqlConnection cnn = BDComun.ObtnerCONexion();
{
SqlCommand comando = new SqlCommand(string.Format("Select
sum(total_detalle)from Detalle_facturap where id_facturap= '{0}'", txtnum.Text),
cnn);
TextTotal.Text = comando.ExecuteScalar().ToString();
SqlCommand coma = new SqlCommand(string.Format("Select
sum(iva_detalle)from Detalle_facturap where id_facturap= '{0}'", txtnum.Text),
cnn);
txtIvaT.Text = coma.ExecuteScalar().ToString();
SqlCommand com = new SqlCommand(string.Format("Select
sum(irbpnr_valor)from Detalle_facturap where id_facturap= '{0}'", txtnum.Text),
cnn);
txtIc.Text = com.ExecuteScalar().ToString();
}
```

```
SqlCommand co = new SqlCommand(string.Format("Select sum(ice_detalle)from
Detalle_facturap where id_facturap= '{0}'", txtnum.Text), cnn);
txtIc.Text = co.ExecuteScalar().ToString();
SqlCommand c = new SqlCommand(string.Format("Select sum(des_detalle)from
Detalle_facturap where id_facturap= '{0}'", txtnum.Text), cnn);
TextDescuento.Text = c.ExecuteScalar().ToString();

float des = float.Parse(TextDescuento.Text);
float iva = float.Parse(txtIvaT.Text);
float total = float.Parse(TextTotal.Text);
float sub = (float.Parse(TextTotal.Text) + float.Parse(TextDescuento.Text)) -
float.Parse(txtIvaT.Text);
txtSub.Text = sub.ToString();
cnn.Close();
}}
```

3.10 Validar cedula

Código utilizado para validar el número de cedula de los clientes en el sistema para un mejor registro de información y mejor funcionamiento del sistema.

```
public int validadorDeCedula(String ced)
{ int resultado = -1;
int isNumeric;
var total = 0;
const int tamanoLongitudCedula = 10;
int[] coeficientes = { 2, 1, 2, 1, 2, 1, 2, 1, 2 };
const int numeroProvincias = 24;
const int tercerDigito = 6;
if (int.TryParse(ced, out isNumeric) && ced.Length == tamanoLongitudCedula)
{
var provincia = Convert.ToInt32(string.Concat(ced[0], ced[1], string.Empty));
var digitoTres = Convert.ToInt32(ced[2] + string.Empty);
if ((provincia > 0 && provincia <= numeroProvincias) && digitoTres <
tercerDigito)
{
var digitoVerificadorRecibido = Convert.ToInt32(ced[9] + string.Empty);
for (var k = 0; k < coeficientes.Length; k++)
{
var valor = Convert.ToInt32(coeficientes[k] + string.Empty) *
Convert.ToInt32(ced[k] + string.Empty);
total = valor <= 10 ? total + (valor - 9) : total + valor;
}
var digitoVerificadoObtenido = total >= 10 ? (total % 10) != 0 ?
10 - (total % 10) : (total % 10) : total;
//return digitoVerificadoObtenido = digitoVerificadorRecibido;
```

```
return 50;
}
}
return resultado;
}
```

3.11 Validar Ruc

Código utilizado para validar el número de Ruc de los clientes en el sistema para un mejor registro de información y mejor funcionamiento del sistema.

```
public int validadorDeRuc(String ruc)
{
    int resultado = -1;
    long isNumeric;
    const int tamanoLongitudRuc = 13;
    //const string establecimiento = "001";
    if (long.TryParse(ruc, out isNumeric) && ruc.Length.Equals(tamanoLongitudRuc))
    {
        var numeroProvincia = Convert.ToInt32(String.Concat(ruc[0] + string.Empty, ruc[1]
        + string.Empty));
        var personaNatural = Convert.ToInt32(ruc[2] + string.Empty);
        if((numeroProvincia >= 1 && numeroProvincia <= 24) && (personaNatural >= 0
        && personaNatural <6))
        {
            resultado = 50;
        }
    }
    return resultado;
}
```

3.12 Encriptar y desencriptar contraseña

Código utilizado para encriptar y desencriptar contraseñas en el sistema para una mejor seguridad de información y mejor funcionamiento del sistema.

```
public string Encriptar(string texto)
{
    //arreglo de bytes donde guardaremos la llave
    byte[] keyArray;
    //arreglo de bytes donde guardaremos el texto
    //que vamos a encriptar
    byte[] Arreglo_a_Cifrar =
    UTF8Encoding.UTF8.GetBytes(texto);
```



```
//se utilizan las clases de encriptación
//provistas por el Framework
//Algoritmo MD5
MD5CryptoServiceProvider hashmd5 =
new MD5CryptoServiceProvider();
//se guarda la llave para que se le realice
//hashing
keyArray = hashmd5.ComputeHash(
UTF8Encoding.UTF8.GetBytes(key));
hashmd5.Clear();
//Algoritmo 3DAS
TripleDESCryptoServiceProvider tdes =
new TripleDESCryptoServiceProvider();
tdes.Key = keyArray;
tdes.Mode = CipherMode.ECB;
tdes.Padding = PaddingMode.PKCS7;

//se empieza con la transformación de la cadena
ICryptoTransform cTransform =
tdes.CreateEncryptor();
//arreglo de bytes donde se guarda la
//cadena cifrada
byte[] ArrayResultado =
cTransform.TransformFinalBlock(Arreglo_a_Cifrar,
0, Arreglo_a_Cifrar.Length);
tdes.Clear();
//se regresa el resultado en forma de una cadena
return Convert.ToBase64String(ArrayResultado,0, ArrayResultado.Length);
}
public string Descriptar(string textoEncriptado)
{
byte[] keyArray;
//convierte el texto en una secuencia de bytes
byte[] Array_a_Descifrar =Convert.FromBase64String(textoEncriptado);
//se llama a las clases que tienen los algoritmos
//de encriptación se le aplica hashing
//algoritmo MD5
MD5CryptoServiceProvider hashmd5 =
new MD5CryptoServiceProvider();
keyArray = hashmd5.ComputeHash(
UTF8Encoding.UTF8.GetBytes(key));
hashmd5.Clear();
TripleDESCryptoServiceProvider tdes =new TripleDESCryptoServiceProvider();
tdes.Key = keyArray;
tdes.Mode = CipherMode.ECB;
tdes.Padding = PaddingMode.PKCS7;
ICryptoTransform cTransform =tdes.CreateDecryptor();
byte[] resultArray =cTransform.TransformFinalBlock(Array_a_Descifrar,0,
Array_a_Descifrar.Length);
tdes.Clear();
```

```
//se regresa en forma de cadena
return UTF8Encoding.UTF8.GetString(resultArray);
}
```

3.13 Conexión a la base de datos

Código utilizado para conectar el sistema a la base de datos .

```
public class BDComun
public static SqlConnection ObtnerCONexion()
{
    SqlConnection Conn = new SqlConnection("Data source=ESTEBAN; Initial
    Catalog= Viascala; Integrated Security = true");
    Conn.Open();
    return Conn;
}
}
```

3.14 Pdf y Xml

Código utilizado para generar los comprobantes electrónicos de los clientes en el sistema.

```
private void comprobantes()
{
    string val = tId.Text;
    XElement xml =
    new XElement("factura", new XAttribute("id", "comprobante"), new
    XAttribute("version", "1.0.0"),
    new XElement("infoTributaria",
    new XElement("ambiente", facturaE.iambiente),
    new XElement("tipoEmision", facturaE.iemision),
    new XElement("razonSocial", facturaE.irazon_social),
    new XElement("ruc", facturaE.iruc_empresa),
    new XElement("claveAcceso", ""),
    new XElement("codDoc", "01"),
    new XElement("estab", txtCE.Text),
    new XElement("ptoEmi", txtE.Text),
    new XElement("secuencial", txtNumero.Text),
    new XElement("dirMatriz", txtMatriz.Text)),
    new XElement("infoFactura",
    new XElement("fechaEmision", txtDate.Text),
    new XElement("dirEstablecimiento", txtEst.Text),
    new XElement("obligadoContabilidad", "Si"),
    new XElement("tipoIdentificacionComprador", "04"),
```

```

new XElement("razonSocialComprador", txtNom.Text),
new XElement("identificacionComprador", txtCliente.Text),
new XElement("totalSinImpuestos", txtSub.Text),
new XElement("totalDescuento", TextDescuento.Text),
new XElement("totalConImpuestos",
new XElement("totalImpuesto",
new XElement("codigo", "2"),
new XElement("codigoPorcentaje", "2"),
new XElement("baseImponible", txtSub.Text),
new XElement("valor", txtIva.Text))),
new XElement("propina", TextOtros.Text),
new XElement("importeTotal", ""),
new XElement("moneda", "Dolar")),
new XElement("detalles",
new XElement("codigoPrincipal", ""),
new XElement("codigoAuxiliar", ""),
new XElement("descripcion", ""),
new XElement("cantidad", ""),
new XElement("precioUnitario", ""),
new XElement("descuento", ""),
new XElement("precioTotalSinImpuestos", ""),
new XElement("detallesAdicionales",
new XElement("descuento", "")),
new XElement("Impuestos",
new XElement("codigo", ""),
new XElement("codigoPorcentaje", ""),
new XElement("tarifa", ""),
new XElement("baseImponible", ""),
new XElement("valor", ""))),
new XElement("infoAdicional",
new XElement("campoAdicional", new XAttribute("nombre", "email"),
txtCorreo.Text),
new XElement("campoAdicional", new XAttribute("nombre", "usuario"),
txtCliente.Text),
new XElement("campoAdicional", new XAttribute("nombre", "clave"),
txtCliente.Text))
);
XDocument facturaXml = new XDocument(xml);
Prueba rpt = new Prueba();
rpt.SetParameterValue("id@", val);
string sec;
string nom;
if (Convert.ToInt32(val) > 999 && Convert.ToInt32(val) < 10000)
{
sec = "000000" + val;
nom = "Fact_" + txtCE.Text + "-" + txtE.Text + "-" + sec;
rpt.ExportToDisk(ExportFormatType.PortableDocFormat,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".pdf");
}

```

```
facturaXml.Save(@"C:\Users\USER\Desktop\Factura\Facturacion
elec\Documentos\" + nom + ".xml");
//rpt.ExportToDisk(ExportFormatType.Xml,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".xml");
}
if (Convert.ToInt32(val) > 99 && Convert.ToInt32(val) < 1000)
{
sec = "00000000" + val;
nom = "Fact_" + txtCE.Text + "-" + txtE.Text + "-" + sec;
rpt.ExportToDisk(ExportFormatType.PortableDocFormat,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".pdf");
facturaXml.Save(@"C:\Users\USER\Desktop\Factura\Facturacion
elec\Documentos\" + nom + ".xml");
//rpt.ExportToDisk(ExportFormatType.Xml,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".xml");
}
if (Convert.ToInt32(val) > 9 && Convert.ToInt32(val) < 100)
{
sec = "000000000" + val;
nom = "Fact_" + txtCE.Text + "-" + txtE.Text + "-" + sec;
rpt.ExportToDisk(ExportFormatType.PortableDocFormat,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".pdf");
facturaXml.Save(@"C:\Users\USER\Desktop\Factura\Facturacion
elec\Documentos\" + nom + ".xml");
// rpt.ExportToDisk(ExportFormatType.Xml,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".xml");
}
if (Convert.ToInt32(val) > 0 && Convert.ToInt32(val) < 9)
{
sec = "0000000000" + val;
nom = "Fact_" + txtCE.Text + "-" + txtE.Text + "-" + sec;
rpt.ExportToDisk(ExportFormatType.PortableDocFormat,
@"C:\Users\USER\Desktop\Factura\Facturacion elec\Documentos\" + nom +
".pdf");
facturaXml.Save(@"C:\Users\USER\Desktop\Factura\Facturacion
elec\Documentos\" + nom + ".xml");
}
```

3.15 Script de base de datos

El siguiente código es de un archivo .SQL que se genera desde el administrador de base de datos de SQL server 2008, el mismo que es un lenguaje declarativo de acceso a base de datos o también conocido SQL, mediante el cual se elabora la base de datos y cada una de sus tablas con sus respectivos procedimientos almacenados.

```
CREATE TABLE [Empresa]
(
    [id_empresa] Int IDENTITY(1,1) NOT NULL,
    [ruc_empresa] Varchar(15) NOT NULL,
    [razon_social] Varchar(50) NOT NULL,
    [nombre_empresa] Varchar(30) NOT NULL,
    [direccion_matriz] Varchar(50) NOT NULL,
    [direccion_empresa] Varchar(50) NOT NULL,
    [telefono_empresa] Varchar(15) NOT NULL,
    [codigo_empresa] Varchar(15) NOT NULL,
    [codigo_emision] Varchar(15) NOT NULL,
    [contribuyente] Varchar(15) NOT NULL,
    [tipo_emision] Varchar(30) NOT NULL,
    [tipo_ambiente] Varchar(30) NOT NULL,
    [dir_comprobante] Varchar(300) NULL
)
go

-- Add keys for table Empresa

ALTER TABLE [Empresa] ADD CONSTRAINT [ruc_empresa] PRIMARY KEY
([ruc_empresa])
go

-- Table Usuario

CREATE TABLE [Usuario]
(
    [id_usuario] Int IDENTITY(1,1) NOT NULL,
    [ci_usuario] Varchar(15) DEFAULT 0000000000 NOT NULL,
    [nombre_usuario] Varchar(30) NOT NULL,
    [apelllido_usuario] Varchar(30) NOT NULL,
    [clave_usuario] Varchar(15) NOT NULL,
    [confirma_usuario] Varchar(15) NOT NULL,
    [fecha_usuario] Datetime NOT NULL,
    [estado_usuario] Varchar(15) NOT NULL,
    [ruc_empresa] Varchar(15) NULL,
    [administrador] Int NULL,
    [editar] Int NULL,
```

```
[eliminar] Int NULL,  
[imprimir] Int NULL  
)  
go
```

```
-- Add keys for table Usuario
```

```
ALTER TABLE [Usuario] ADD CONSTRAINT [ci_usuario] PRIMARY KEY  
([ci_usuario])  
go
```

```
-- Table Cliente
```

```
CREATE TABLE [Cliente]  
(  
[id_cliente] Int IDENTITY(1,1) NOT NULL,  
[razon_cliente] Varchar(50) NOT NULL,  
[tipo_identificacion] Varchar(30) NOT NULL,  
[ci_cliente] Varchar(15) NOT NULL,  
[tipo_cliente] Varchar(30) NOT NULL,  
[direccion_cliente] Varchar(50) NOT NULL,  
[telefono_cliente] Varchar(15) NOT NULL,  
[celular_cliente] Varchar(15) NOT NULL,  
[estado_cliente] Varchar(15) NOT NULL,  
[correo_cliente] Varchar(30) NOT NULL,  
[ci_usuario] Varchar(15) DEFAULT 0000000000 NULL  
)  
go
```

```
-- Add keys for table Cliente
```

```
ALTER TABLE [Cliente] ADD CONSTRAINT [ci_cliente] PRIMARY KEY  
([ci_cliente])  
go
```

```
-- Table Factura
```

```
CREATE TABLE [Factura]  
(  
[id_factura] Int IDENTITY(1,1) NOT NULL,  
[ruc] Varchar(15) NOT NULL,  
[nombre] Varchar(50) NOT NULL,  
[razon_social] Varchar(50) NOT NULL,  
[direccion_matriz] Varchar(50) NOT NULL,  
[direccion_empresa] Varchar(50) NOT NULL,  
[tipo_comprobante] Varchar(30) NULL,  
[ambiente] Varchar(30) NULL,  
[emision] Varchar(30) NULL,  
[numero_factura] Varchar(30) NOT NULL,  
[fecha_factura] Varchar(30) NOT NULL,
```

```
[guia_remision] Varchar(50) NULL,  
[clave] Varchar(60) NULL,  
[autorizacion_factura] Varchar(50) NOT NULL,  
[ci_cliente] Varchar(15) NOT NULL,  
[razon_comprador] Varchar(30) NOT NULL,  
[correo] Varchar(30) NULL,  
[telefono] Varchar(30) NULL,  
[direccion] Varchar(30) NULL,  
[adicional_uno] Varchar(50) NULL,  
[adicional_dos] Varchar(50) NULL,  
[adicional_tres] Varchar(50) NULL,  
[subtotal_factura] Numeric(14,2) NOT NULL,  
[sub_12] Numeric(14,2) NOT NULL,  
[sub_0] Numeric(14,2) NOT NULL,  
[sub_no_iva] Numeric(14,2) NOT NULL,  
[sub_excento] Numeric(14,2) NOT NULL,  
[descuento_factura] Numeric(14,2) NOT NULL,  
[ice] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[iva_factura] Numeric(14,2) NOT NULL,  
[otros_factura] Numeric(14,2) NOT NULL,  
[total_factura] Numeric(14,2) NOT NULL,  
[estado_factura] Varchar(15) NOT NULL,  
[forma_pago_factura] Varchar(15) NOT NULL  
)  
go
```

-- Add keys for table Factura

```
ALTER TABLE [Factura] ADD CONSTRAINT [id_factura] PRIMARY KEY  
([id_factura])  
go
```

-- Table FacturaProveedores

```
CREATE TABLE [FacturaProveedores]  
(  
[id_facturaP] Int IDENTITY(1,1) NOT NULL,  
[ci_proveedor] Varchar(15) NULL,  
[razon_proveedor] Varchar(30) NOT NULL,  
[direccion_matriz] Varchar(50) NOT NULL,  
[direccion_empresa] Varchar(50) NOT NULL,  
[tipo_comprobante] Varchar(30) NULL,  
[numero_factura] Varchar(30) NOT NULL,  
[fecha_factura] Varchar(30) NOT NULL,  
[ruc] Varchar(15) NOT NULL,  
[razon_social] Varchar(50) NOT NULL,  
[adicional_uno] Varchar(50) NULL,  
[adicional_dos] Varchar(50) NULL,  
[adicional_tres] Varchar(50) NULL,  
)
```



```
[subtotal_factura] Numeric(14,2) NOT NULL,  
[sub_12] Numeric(14,2) NOT NULL,  
[sub_0] Numeric(14,2) NOT NULL,  
[sub_no_iva] Numeric(14,2) NOT NULL,  
[sub_excento] Numeric(14,2) NOT NULL,  
[descuento_factura] Numeric(14,2) NOT NULL,  
[ice] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[iva_factura] Numeric(14,2) NOT NULL,  
[otros_factura] Numeric(14,2) NOT NULL,  
[total_factura] Numeric(14,2) NOT NULL,  
[estado_factura] Varchar(15) NOT NULL,  
[forma_pago_factura] Varchar(15) NOT NULL  
)  
go
```

```
-- Add keys for table FacturaProveedores
```

```
ALTER TABLE [FacturaProveedores] ADD CONSTRAINT [Key4] PRIMARY  
KEY ([id_facturaP])  
go
```

```
-- Table Detalle_factura
```

```
CREATE TABLE [Detalle_factura]  
(  
[id_detalle] Int IDENTITY(1,1) NOT NULL,  
[cod_detalle] Varchar(15) NOT NULL,  
[descripcion_detalle] Varchar(30) NOT NULL,  
[cantidad_detalle] Int NOT NULL,  
[precio_detalle] Numeric(14,2) NOT NULL,  
[des_detalle] Numeric(14,2) NOT NULL,  
[id_factura] Int NOT NULL,  
[id_producto] Int NOT NULL,  
[id_compra] Int NOT NULL,  
[iva_detalle] Numeric(14,2) NOT NULL,  
[ice_detalle] Numeric(14,2) NOT NULL,  
[irbpnr_valor] Numeric(14,2) NOT NULL,  
[total_detalle] Numeric(14,2) NOT NULL  
)  
go
```

```
-- Add keys for table Detalle_factura
```

```
ALTER TABLE [Detalle_factura] ADD CONSTRAINT [id_detalle] PRIMARY  
KEY ([id_detalle])  
go
```

```
-- Table Detalle_facturap
```



```
CREATE TABLE [Detalle_facturap]
(
    [id_detalle] Int IDENTITY(1,1) NOT NULL,
    [id_facturaP] Int NULL,
    [cod_detalle] Varchar(15) NOT NULL,
    [descripcion_detalle] Varchar(30) NOT NULL,
    [cantidad_detalle] Int NOT NULL,
    [precio_detalle] Numeric(14,2) NOT NULL,
    [des_detalle] Numeric(14,2) NOT NULL,
    [iva_detalle] Numeric(14,2) NOT NULL,
    [ice_detalle] Numeric(14,2) NOT NULL,
    [irbpnr_valor] Numeric(14,2) NOT NULL,
    [total_detalle] Numeric(14,2) NOT NULL
)
go

-- Add keys for table Detalle_facturap

ALTER TABLE [Detalle_facturap] ADD CONSTRAINT [id_detalle] PRIMARY
KEY ([id_detalle])
go

-- Table Producto

CREATE TABLE [Producto]
(
    [id_producto] Int IDENTITY(1,1) NOT NULL,
    [codigo_producto] Varchar(6) NOT NULL,
    [descripcion_producto] Varchar(30) NOT NULL,
    [precio_producto] Numeric(14,2) NOT NULL,
    [cantidad_producto] Int NOT NULL,
    [descuento_producto] Numeric(14,2) NOT NULL,
    [iva] Numeric(14,2) NOT NULL,
    [ice] Numeric(14,2) NOT NULL,
    [irbpnr] Numeric(14,2) NOT NULL,
    [total_producto] Numeric(14,2) NOT NULL
)
go

-- Add keys for table Producto

ALTER TABLE [Producto] ADD CONSTRAINT [Key6] PRIMARY KEY
([id_producto])
go

-- Table Historial

CREATE TABLE [Historial]
(
    [id_usuario] Int IDENTITY(1,1) NOT NULL,
```

```
[nombre_usuario] Varchar(50) NOT NULL,  
[ultimo_acceso] Datetime NOT NULL,  
[ip_acceso] Varchar(15) NOT NULL,  
[ci_usuario] Varchar(15) DEFAULT 0000000000 NULL  
)  
go
```

-- Add keys for table Historial

```
ALTER TABLE [Historial] ADD CONSTRAINT [Key7] PRIMARY KEY  
([id_usuario])  
go
```

-- Table Inventario

```
CREATE TABLE [Inventario]  
(  
[id_compra] Int IDENTITY(1,1) NOT NULL,  
[codigo_compra] Varchar(6) NOT NULL,  
[detalle_compra] Varchar(30) NOT NULL,  
[fecha_compra] Datetime NOT NULL,  
[precio_compra] Numeric(14,2) NOT NULL,  
[cantidad_compra] Int NOT NULL,  
[descuento_compra] Numeric(14,2) NOT NULL,  
[iva] Numeric(14,2) NOT NULL,  
[ice] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[total_compra] Numeric(14,2) NOT NULL,  
[responsable] Int NOT NULL  
)  
go
```

-- Add keys for table Inventario

```
ALTER TABLE [Inventario] ADD CONSTRAINT [Key10] PRIMARY KEY  
([id_compra])  
go
```

-- Table Nota_debito

```
CREATE TABLE [Nota_debito]  
(  
[id_debito] Int IDENTITY(1,1) NOT NULL,  
[ruc] Varchar(15) NOT NULL,  
[nombre] Varchar(30) NOT NULL,  
[razon_social] Varchar(30) NOT NULL,  
[direccion_matriz] Varchar(50) NOT NULL,  
[direccion_empresa] Varchar(50) NOT NULL,  
[numero] Int NOT NULL,  
[fecha] Varchar(20) NOT NULL,
```

```
[clave] Varchar(20) NULL,  
[ci_cliente] Varchar(15) NOT NULL,  
[razon_comprador] Varchar(30) NOT NULL,  
[tipo_documento] Varchar(20) NOT NULL,  
[fecha_emision] Varchar(20) NOT NULL,  
[numero_comprobante] Varchar(30) NOT NULL,  
[adicional_uno] Varchar(50) NULL,  
[adicional_dos] Varchar(50) NULL,  
[adicional_tres] Varchar(50) NULL,  
[subtotal] Numeric(14,2) NOT NULL,  
[sub_12] Numeric(14,2) NOT NULL,  
[sub_0] Numeric(14,2) NOT NULL,  
[sub_no_iva] Numeric(14,2) NOT NULL,  
[sub_exento] Numeric(14,2) NOT NULL,  
[descuento] Numeric(14,2) NOT NULL,  
[ice] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[iva] Numeric(14,2) NOT NULL,  
[total] Numeric(14,2) NOT NULL  
)  
go
```

-- Add keys for table Nota_debito

```
ALTER TABLE [Nota_debito] ADD CONSTRAINT [Key11] PRIMARY KEY  
([id_debito])  
go
```

-- Table Detalle_debito

```
CREATE TABLE [Detalle_debito]  
(  
[id_detalle] Int IDENTITY(1,1) NOT NULL,  
[id_debito] Int NOT NULL,  
[codigo_debito] Varchar(15) NOT NULL,  
[detalle_debito] Varchar(30) NOT NULL,  
[precio_debito] Numeric(14,2) NOT NULL,  
[cantidad_debito] Int NOT NULL,  
[descuento_debito] Numeric(14,2) NOT NULL,  
[total_debito] Numeric(14,2) NOT NULL,  
[ice_debito] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[irbpnr_valor] Numeric(14,2) NOT NULL,  
[accion_credito] Varchar(20) NOT NULL  
)  
go
```

-- Add keys for table Detalle_debito

```
ALTER TABLE [Detalle_debito] ADD CONSTRAINT [Key13] PRIMARY KEY  
([id_detalle])  
go
```

```
-- Table Detalle_credito
```

```
CREATE TABLE [Detalle_credito]  
(  
    [id_detalle] Int IDENTITY(1,1) NOT NULL,  
    [id_credito] Int NOT NULL,  
    [codigo_credito] Varchar(15) NOT NULL,  
    [detalle_credito] Varchar(30) NOT NULL,  
    [precio_credito] Numeric(14,2) NOT NULL,  
    [cantidad_credito] Int NOT NULL,  
    [total_credito] Numeric(14,2) NOT NULL,  
    [descuento_credito] Numeric(14,2) NOT NULL,  
    [ice_credito] Numeric(14,2) NOT NULL,  
    [irbpnr] Numeric(14,2) NOT NULL,  
    [irbpnr_valor] Numeric(14,2) NOT NULL,  
    [accion_credito] Nvarchar(20) NOT NULL  
)  
go
```

```
-- Add keys for table Detalle_credito
```

```
ALTER TABLE [Detalle_credito] ADD CONSTRAINT [Key14] PRIMARY KEY  
([id_detalle])  
go
```

```
-- Table Nota_credito
```

```
CREATE TABLE [Nota_credito]  
(  
    [id_credito] Int IDENTITY(1,1) NOT NULL,  
    [ruc] Varchar(15) NOT NULL,  
    [nombre] Varbinary(50) NOT NULL,  
    [razon_social] Varchar(50) NOT NULL,  
    [direccion_matriz] Varchar(50) NOT NULL,  
    [direccion_empresa] Varchar(50) NOT NULL,  
    [numero] Int NOT NULL,  
    [fecha] Varchar(20) NOT NULL,  
    [clave] Varchar(20) NULL,  
    [ci_cliente] Varchar(15) NULL,  
    [razon_comprador] Varchar(30) NOT NULL,  
    [tipo_documento] Varchar(30) NOT NULL,  
    [fecha_emision] Varchar(20) NOT NULL,  
    [numero_comprobante] Varchar(30) NOT NULL,  
    [motivo] Varchar(50) NOT NULL,  
    [adicional_uno] Varchar(50) NULL,  
    [adicional_dos] Varchar(50) NULL,  
)
```

```
[adicional_tres] Varchar(50) NULL,  
[subtotal] Numeric(14,2) NOT NULL,  
[sub_12] Numeric(14,2) NOT NULL,  
[sub_0] Numeric(14,2) NOT NULL,  
[sub_no_iva] Numeric(14,2) NOT NULL,  
[sub_exento] Numeric(14,2) NOT NULL,  
[descuento] Numeric(14,2) NOT NULL,  
[ice] Numeric(14,2) NOT NULL,  
[irbpnr] Numeric(14,2) NOT NULL,  
[iva] Numeric(14,2) NOT NULL,  
[total] Numeric(14,2) NOT NULL  
)  
go
```

-- Add keys for table Nota_credito

```
ALTER TABLE [Nota_credito] ADD CONSTRAINT [id_credito] PRIMARY KEY  
([id_credito])  
go
```

-- Table Retencion

```
CREATE TABLE [Retencion]  
(  
[id_retencion] Int NOT NULL,  
[ruc] Varbinary(15) NOT NULL,  
[nombre] Varchar(30) NOT NULL,  
[razon_social] Varchar(50) NOT NULL,  
[direccion_matriz] Varchar(50) NOT NULL,  
[direccion_empresa] Varchar(50) NOT NULL,  
[numero] Int NOT NULL,  
[fecha] Datetime NOT NULL,  
[clave] Char(20) NULL,  
[ruc_comprador] Varchar(15) NOT NULL,  
[razon_comprador] Varchar(30) NOT NULL,  
[tipo_documento] Varchar(20) NOT NULL,  
[fecha_emision] Datetime NOT NULL,  
[numero_comprobante] Int NOT NULL,  
[impuesto] Numeric(18,0) NOT NULL,  
[base_imponible] Numeric(18,0) NOT NULL,  
[porcentaje] Numeric(18,0) NOT NULL,  
[valor_retenido] Numeric(18,0) NOT NULL,  
[adicional_uno] Varchar(50) NULL,  
[adicional_dos] Varchar(50) NULL,  
[adicional_tres] Varchar(50) NULL,  
[id_facturaP] Int NULL  
)  
go
```

-- Add keys for table Retencion

```
ALTER TABLE [Retencion] ADD CONSTRAINT [Key15] PRIMARY KEY  
([id_retencion])  
go
```

-- Table Impuesto

```
CREATE TABLE [Impuesto]  
(  
    [id_impuesto] Int NOT NULL,  
    [codigo] Varchar(15) NOT NULL,  
    [descripcion] Varchar(50) NOT NULL,  
    [base_imponible] Numeric(14,2) NOT NULL,  
    [porcentaje] Numeric(14,2) NOT NULL,  
    [total] Numeric(14,2) NOT NULL,  
    [documento] Varchar(20) NOT NULL,  
    [fecha] Datetime NOT NULL,  
    [tipo] Varchar(15) NOT NULL,  
    [id_retencion] Int NULL  
)  
go
```

-- Add keys for table Impuesto

```
ALTER TABLE [Impuesto] ADD CONSTRAINT [Key16] PRIMARY KEY  
([id_impuesto])  
go
```

-- Table Proveedor

```
CREATE TABLE [Proveedor]  
(  
    [id_proveedor] Int IDENTITY(1,1) NOT NULL,  
    [razon_proveedor] Varchar(50) NOT NULL,  
    [tipo_identificacion] Varchar(30) NOT NULL,  
    [ci_proveedor] Varchar(15) NOT NULL,  
    [direccion_proveedor] Varchar(50) NOT NULL,  
    [establecimiento] Varchar(50) NOT NULL,  
    [telefono_proveedor] Varchar(15) NOT NULL,  
    [celular_proveedor] Varchar(15) NOT NULL,  
    [estado_proveedor] Varchar(15) NOT NULL,  
    [correo_proveedor] Varchar(30) NOT NULL  
)  
go
```

-- Add keys for table Proveedor

```
ALTER TABLE [Proveedor] ADD CONSTRAINT [ci_proveedor] PRIMARY KEY  
([ci_proveedor])  
go
```

-- Create relationships section -----

```
ALTER TABLE [Usuario] ADD CONSTRAINT [Relationship4] FOREIGN KEY
([ruc_empresa]) REFERENCES [Empresa] ([ruc_empresa])
go
ALTER TABLE [Cliente] ADD CONSTRAINT [Relationship5] FOREIGN KEY
([ci_usuario]) REFERENCES [Usuario] ([ci_usuario])
go
ALTER TABLE [Detalle_factura] ADD CONSTRAINT [Relationship7] FOREIGN
KEY ([id_factura]) REFERENCES [Factura] ([id_factura])
go
ALTER TABLE [Detalle_factura] ADD CONSTRAINT [Relationship13]
FOREIGN KEY ([id_producto]) REFERENCES [Producto] ([id_producto])
go
ALTER TABLE [Factura] ADD CONSTRAINT [Relationship25] FOREIGN KEY
([ci_cliente]) REFERENCES [Cliente] ([ci_cliente])
go
ALTER TABLE [Historial] ADD CONSTRAINT [Relationship27] FOREIGN KEY
([ci_usuario]) REFERENCES [Usuario] ([ci_usuario])
go
ALTER TABLE [Detalle_factura] ADD CONSTRAINT [Relationship29]
FOREIGN KEY ([id_compra]) REFERENCES [Inventario] ([id_compra])
go
ALTER TABLE [Nota_debito] ADD CONSTRAINT [Relationship30] FOREIGN
KEY ([ci_cliente]) REFERENCES [Cliente] ([ci_cliente])
go
ALTER TABLE [Detalle_debito] ADD CONSTRAINT [Relationship33] FOREIGN
KEY ([id_debito]) REFERENCES [Nota_debito] ([id_debito])
go
ALTER TABLE [Detalle_credito] ADD CONSTRAINT [Relationship36]
FOREIGN KEY ([id_credito]) REFERENCES [Nota_credito] ([id_credito])
go
ALTER TABLE [Nota_credito] ADD CONSTRAINT [Relationship39] FOREIGN
KEY ([ci_cliente]) REFERENCES [Cliente] ([ci_cliente])
go
ALTER TABLE [Impuesto] ADD CONSTRAINT [Relationship40] FOREIGN
KEY ([id_retencion]) REFERENCES [Retencion] ([id_retencion])
go
ALTER TABLE [Retencion] ADD CONSTRAINT [Relationship42] FOREIGN
KEY ([id_facturaP]) REFERENCES [FacturaProveedores] ([id_facturaP])
go
ALTER TABLE [FacturaProveedores] ADD CONSTRAINT [Relationship43]
FOREIGN KEY ([ci_proveedor]) REFERENCES [Proveedor] ([ci_proveedor])
go
ALTER TABLE [Detalle_facturap] ADD CONSTRAINT [Relationship44]
FOREIGN KEY ([id_facturaP]) REFERENCES [FacturaProveedores]
([id_facturaP])
go
```

3.16 Procedimientos almacenados

```
create proc delcliente (  
    @ci_cliente int,  
    @estado_cliente varchar(15)  
)  
as update Cliente set estado_cliente=@estado_cliente where ci_cliente=@ci_cliente
```

```
create proc inscliente(  
    @ci_cliente varchar(15),  
    @razon_cliente varchar(50),  
    @tipo_cliente varchar(30),  
    @tipo_identificacion varchar(30),  
    @direccion_cliente varchar(50),  
    @telefono_cliente varchar(15),  
    @celular_cliente varchar(15),  
    @estado_cliente varchar(15),  
    @correo_cliente varchar(30),  
    @ci_usuario varchar(15)  
)  
AS insert into  
    Cliente values  
    (@razon_cliente, @tipo_identificacion, @ci_cliente, @tipo_cliente,  
    @direccion_cliente, @telefono_cliente, @celular_cliente, @estado_cliente,  
    @correo_cliente, @ci_usuario)
```

```
create proc updcliente(  
    @ci_cliente varchar(15),  
    @razon_cliente varchar(50),  
    @tipo_cliente varchar(30),  
    @tipo_identificacion varchar(30),  
    @direccion_cliente varchar(50),  
    @telefono_cliente varchar(15),  
    @celular_cliente varchar(15),  
    @estado_cliente varchar(15),  
    @correo_cliente varchar(30),  
    @ci_usuario varchar(15)  
)  
as update Cliente set razon_cliente=@razon_cliente,  
    tipo_cliente=@tipo_cliente, direccion_cliente=@direccion_cliente,  
    tipo_identificacion=@tipo_identificacion,  
    telefono_cliente=@telefono_cliente, celular_cliente=@celular_cliente,  
    estado_cliente=@estado_cliente, correo_cliente=@correo_cliente  
    where ci_cliente=@ci_cliente;  
create proc obtenercliente (  
    @ci_cliente varchar(15))  
as Select ci_cliente from Cliente Where ci_cliente LIKE '%' + @ci_cliente + '%';
```



```
create proc delusuario(  
    @ci_usuario int  
)  
as delete from Usuario where ci_usuario=@ci_usuario;
```

```
create proc insusuario(  
    @ci_usuario varchar(15),  
    @nombre_usuario varchar(30),  
    @apellido_usuario varchar(30),  
    @clave_usuario varchar(15),  
    @confirma_usuario varchar(15),  
    @fecha_usuario datetime,  
    @estado_usuario varchar(10),  
    @ruc_empresa varchar(15),  
    @administrador int,  
    @editar int,  
    @eliminar int,  
    @imprimir int  
)  
AS insert into  
    Usuario values  
    (@ci_usuario, @nombre_usuario, @apellido_usuario, @clave_usuario,  
    @confirma_usuario, @fecha_usuario, @estado_usuario, @ruc_empresa,  
    @administrador, @editar, @eliminar, @imprimir)
```

```
create proc updusuario(  
    @ci_usuario varchar(15),  
    @nombre_usuario varchar(30),  
    @apellido_usuario varchar(30),  
    @clave_usuario varchar(15),  
    @confirma_usuario varchar(15),  
    @fecha_usuario datetime,  
    @estado_usuario varchar(10),  
    @administrador int,  
    @editar int,  
    @eliminar int,  
    @imprimir int  
)  
as update Usuario set nombre_usuario=@nombre_usuario,  
    apellido_usuario=@apellido_usuario,  
    clave_usuario=@clave_usuario, confirma_usuario=@confirma_usuario,  
    fecha_usuario=@fecha_usuario,  
    estado_usuario=@estado_usuario, administrador=@administrador, editar=@editar,  
    eliminar=@eliminar,  
    imprimir=@imprimir where ci_usuario=@ci_usuario;
```

```
create proc delinventario(  
    @id_compra int  
)
```

as delete from Inventario where id_compra=@id_compra;

```
create proc insinventario(  
    @codigo_compra varchar(6),  
    @detalle_compra varchar (30),  
    @fecha_compra datetime,  
    @precio_compra numeric(14,2),  
    @cantidad_compra int,  
    @descuento_compra numeric(14,2),  
    @iva numeric(14,2),  
    @ice numeric(14,2),  
    @irbpnr numeric(14,2),  
    @total_compra numeric(14,2),  
    @responsable int  
)  
AS insert into  
    Inventario values  
    (@codigo_compra, @detalle_compra, @fecha_compra, @precio_compra,  
    @cantidad_compra, @descuento_compra, @iva, @ice, @irbpnr, @total_compra,  
    @responsable)
```

```
create proc delproveedor (  
    @ci_proveedor varchar(15),  
    @estado_proveedor varchar(15)  
)  
as update Proveedor set estado_proveedor=@estado_proveedor where  
ci_proveedor=@ci_proveedor;
```

```
create proc insproveedor(  
    @ci_proveedor varchar(15),  
    @razon_proveedor varchar(50),  
    @tipo_identificacion varchar(30),  
    @direccion_proveedor varchar(50),  
    @establecimiento varchar(50),  
    @telefono_proveedor varchar(15),  
    @celular_proveedor varchar(15),  
    @estado_proveedor varchar(15),  
    @correo_proveedor varchar(30)  
)  
AS insert into  
    Proveedor values  
    (@razon_proveedor, @tipo_identificacion, @ci_proveedor, @direccion_proveedor,  
    @establecimiento, @telefono_proveedor, @celular_proveedor, @estado_proveedor,  
    @correo_proveedor)
```

```
create proc updproveedor(  
    @ci_proveedor varchar(15),  
    @razon_proveedor varchar(50),  
    @direccion_proveedor varchar(50),
```

```
@establecimiento varchar(50),
@telefono_proveedor varchar(15),
@celular_proveedor varchar(15),
@estado_proveedor varchar(15),
@correo_proveedor varchar(30)
)
AS update Proveedor set
    razon_proveedor=@razon_proveedor, direccion_proveedor=@direccion_proveedor,
    establecimiento=@establecimiento,
    telefono_proveedor=@telefono_proveedor,
    celular_proveedor=@celular_proveedor, estado_proveedor=@estado_proveedor,
    correo_proveedor=@correo_proveedor where ci_proveedor=@ci_proveedor
```

```
create proc inempresa(
    @ruc_empresa varchar(15),
    @razon_social varchar(50),
    @nombre_empresa varchar(30),
    @direccion_matriz varchar(50),
    @direccion_empresa varchar(50),
    @telefono_empresa varchar(15),
    @codigo_empresa varchar(15),
    @codigo_emision varchar(15),
    @contribuyente varchar(15),
    @tipo_emision varchar(30),
    @tipo_ambiente varchar(30),
    @dir_comprobante varchar(300)
)
AS insert into
    Empresa values
    (@ruc_empresa, @razon_social, @nombre_empresa, @direccion_matriz,
    @direccion_empresa, @telefono_empresa, @codigo_empresa, @codigo_emision,
    @contribuyente, @tipo_emision, @tipo_ambiente, @dir_comprobante)
```

```
create proc inshistorial(
    @nombre_usuario char(25),
    @ultimo_acceso datetime,
    @ip_acceso char(15),
    @ci_usuario int
)
AS insert into
    Historial values
    (@nombre_usuario, @ultimo_acceso, @ip_acceso, @ci_usuario)
```

```
create proc deldetalle(
    @id_detalle int
)
as delete from Detalle_factura where id_detalle=@id_detalle;
```

```
create proc insdetalle(  
    @cod_detalle varchar (6),  
    @descripcion_detalle varchar(30),  
    @cantidad_detalle int,  
    @precio_detalle numeric(14,2),  
    @des_detalle numeric(14,2),  
    @id_factura int,  
    @id_producto int,  
    @id_compra int,  
    @iva numeric(14,2),  
    @ice numeric(14,2),  
    @irbpnr_valor numeric(14,2),  
    @total_detalle numeric(14,2)  
)  
AS insert into  
    Detalle_factura values  
    (@cod_detalle, @descripcion_detalle, @cantidad_detalle, @precio_detalle,  
    @des_detalle, @id_factura, @id_producto, @id_compra, @iva, @ice,  
    @irbpnr_valor, @total_detalle)  
  
create proc insdetallec(  
    @id_credito int,  
    @id_factura int  
)  
AS INSERT INTO Detalle_credito(cod_detalle,descripcion_detalle,  
cantidad_detalle, precio_detalle, des_detalle, id_credito, iva_detalle, ice_detalle,  
irbpnr_valor, total_detalle )  
SELECT cod_detalle,  
descripcion_detalle,cantidad_detalle,precio_detalle,des_detalle,@id_credito,  
iva_detalle, ice_detalle, irbpnr_valor, total_detalle FROM Detalle_factura where  
id_factura = @id_factura  
  
create proc insdetallep(  
    @id_facturap int,  
    @cod_detalle varchar (15),  
    @descripcion_detalle varchar(30),  
    @cantidad_detalle int,  
    @precio_detalle numeric(14,2),  
    @des_detalle numeric(14,2),  
    @iva numeric(14,2),  
    @ice numeric(14,2),  
    @irbpnr_valor numeric(14,2),  
    @total_detalle numeric(14,2)  
)  
AS insert into  
    Detalle_facturap values
```

(@id_facturap, @cod_detalle, @descripcion_detalle, @cantidad_detalle,
@precio_detalle, @des_detalle, @iva, @ice, @irbpnr_valor, @total_detalle)

```
create proc insfactura(  
  @ruc_empresa varchar(15),  
  @nombre varchar(50),  
  @razon_social varchar(50),  
  @direccion_matriz varchar(50),  
  @direccion_empresa varchar(50),  
  @tipo_comprobante varchar(30),  
  @ambiente varchar(30),  
  @emision varchar(30),  
  @numero_factura varchar(30),  
  @fecha_factura varchar(30),  
  @guia_remision varchar (50),  
  @clave varchar(20),  
  @autorizacion_factura varchar(50),  
  @ci_cliente varchar(50),  
  @razon_comprador varchar(30),  
  @correo varchar(30),  
  @telefono varchar(30),  
  @direccion varchar(30),  
  @adicional_uno varchar(50),  
  @adicional_dos varchar(50),  
  @adicional_tres varchar(50),  
  @subtotal_factura numeric(14,2),  
  @sub_12 numeric(14,2),  
  @sub_0 numeric(14,2),  
  @sub_no_iva numeric(14,2),  
  @sub_excento numeric(14,2),  
  @descuento_factura numeric(14,2),  
  @ice numeric(14,2),  
  @irbpnr numeric(14,2),  
  @iva_factura numeric(14,2),  
  @otros_factura numeric(14,2),  
  @total_factura numeric(14,2),  
  @estado_factura varchar(15),  
  @forma_pago_factura varchar(15)  
)  
AS insert into  
Factura values  
(@ruc_empresa, @nombre, @razon_social, @direccion_matriz,  
@direccion_empresa, @tipo_comprobante, @ambiente, @emision,  
@numero_factura,  
@fecha_factura, @guia_remision, @clave, @autorizacion_factura, @ci_cliente,  
@razon_comprador, @correo, @telefono, @direccion, @adicional_uno,  
@adicional_dos, @adicional_tres, @subtotal_factura, @sub_12, @sub_0,  
@sub_no_iva, @sub_excento,
```

@descuento_factura, @ice, @irbpnr, @iva_factura, @otros_factura,
@total_factura, @estado_factura, @forma_pago_factura)

```
create proc insfacturap(  
  @ci_proveedor varchar(50),  
  @razon_proveedor varchar(30),  
  @direccion_matriz varchar(50),  
  @direccion_empresa varchar(50),  
  @tipo_comprobante varchar(30),  
  @numero_factura varchar(30),  
  @fecha_factura varchar(30),  
  @ruc varchar(15),  
  @razon_social varchar(50),  
  @adicional_uno varchar(50),  
  @adicional_dos varchar(50),  
  @adicional_tres varchar(50),  
  @subtotal_factura numeric(14,2),  
  @sub_12 numeric(14,2),  
  @sub_0 numeric(14,2),  
  @sub_no_iva numeric(14,2),  
  @sub_excento numeric(14,2),  
  @descuento_factura numeric(14,2),  
  @ice numeric(14,2),  
  @irbpnr numeric(14,2),  
  @iva_factura numeric(14,2),  
  @otros_factura numeric(14,2),  
  @total_factura numeric(14,2),  
  @estado_factura varchar(15),  
  @forma_pago_factura varchar(15)  
)  
AS insert into  
FacturaProveedores values  
(@ci_proveedor, @razon_proveedor, @direccion_matriz, @direccion_empresa,  
@tipo_comprobante, @numero_factura,  
@fecha_factura, @ruc, @razon_social, @adicional_uno, @adicional_dos,  
@adicional_tres, @subtotal_factura, @sub_12, @sub_0, @sub_no_iva,  
@sub_excento,  
@descuento_factura, @ice, @irbpnr, @iva_factura, @otros_factura,  
@total_factura, @estado_factura, @forma_pago_factura)
```

```
create proc updfactura(  
  @id_factura int,  
  @fecha_factura varchar(30),  
  @clave varchar(60),  
  @subtotal_factura numeric(14,2),  
  @sub_12 numeric(14,2),  
  @sub_0 numeric(14,2),  
  @sub_no_iva numeric(14,2),  
  @sub_excento numeric(14,2),
```

```
@descuento_factura numeric(14,2),
@ice numeric(14,2),
@irbpnr numeric(14,2),
@iva_factura numeric(14,2),
@otros_factura numeric(14,2),
@total_factura numeric(14,2),
@forma_pago_factura varchar(15),
@ci_cliente varchar(15)
)
as update Factura set fecha_factura=@fecha_factura, clave=@clave,
subtotal_factura=@subtotal_factura,
sub_12=@sub_12, sub_0=@sub_0, sub_no_iva=@sub_no_iva,
sub_excento=@sub_excento,
descuento_factura=@descuento_factura, ice=@ice, iva_factura=@iva_factura,
irbpnr=@irbpnr,
otros_factura=@otros_factura, total_factura=@total_factura,
forma_pago_factura=@forma_pago_factura,
ci_cliente=@ci_cliente where id_factura=@id_factura;
```

```
create proc insnotacredito(
@ruc varchar(15),
@nombre varchar(50),
@razon_social varchar(50),
@direccion_matriz varchar(50),
@direccion_empresa varchar(50),
@tipo_comprobante varchar(30),
@numero varchar(15),
@fecha varchar(20),
@clave varchar(20),
@ci_cliente varchar(15),
@razon_comprador varchar(30),
@tipo_documento varchar(30),
@fecha_emision varchar(20),
@numero_comprobante varchar(30),
@motivo varchar(300),
@adicional_uno varchar(50),
@adicional_dos varchar(50),
@adicional_tres varchar(50),
@subtotal numeric(14,2),
@sub_12 numeric(14,2),
@sub_0 numeric(14,2),
@sub_no_iva numeric(14,2),
@sub_excento numeric(14,2),
@descuento numeric(14,2),
@ice numeric(14,2),
@irbpnr numeric(14,2),
@iva_factura numeric(14,2),
@total_factura numeric(14,2)
)
```

AS insert into

Nota_credito values

(@ruc, @nombre, @razon_social, @direccion_matriz, @direccion_empresa,
@tipo_comprobante, @numero,
@fecha, @clave, @ci_cliente, @razon_comprador, @tipo_documento,
@fecha_emision, @numero_comprobante, @motivo,
@adicional_uno, @adicional_dos, @adicional_tres, @subtotal, @sub_12, @sub_0,
@sub_no_iva, @sub_excento, @descuento, @ice, @irbpnr, @iva_factura,
@total_factura)

create proc insnotadebito(

@ruc varchar(15),
@nombre varchar(50),
@razon_social varchar(50),
@direccion_matriz varchar(50),
@direccion_empresa varchar(50),
@numero int,
@fecha varchar(20),
@clave varchar(20),
@ci_cliente varchar(15),
@razon_comprador varchar(30),
@tipo_documento varchar(30),
@fecha_emision varchar(20),
@numero_comprobante varchar(30),
@adicional_uno varchar(50),
@adicional_dos varchar(50),
@adicional_tres varchar(50),
@subtotal numeric(14,2),
@sub_12 numeric(14,2),
@sub_0 numeric(14,2),
@sub_no_iva numeric(14,2),
@sub_excento numeric(14,2),
@descuento numeric(14,2),
@ice numeric(14,2),
@irbpnr numeric(14,2),
@iva_factura numeric(14,2),
@total_factura numeric(14,2)
)

AS insert into

Nota_debito values

(@ruc, @nombre, @razon_social, @direccion_matriz, @direccion_empresa,
@numero,
@fecha, @clave, @ci_cliente, @razon_comprador, @tipo_documento,
@fecha_emision, @numero_comprobante,
@adicional_uno, @adicional_dos, @adicional_tres, @subtotal, @sub_12, @sub_0,
@sub_no_iva, @sub_excento,
@descuento, @ice, @irbpnr, @iva_factura, @total_factura)

create proc insproducto(

@codigo_producto varchar (6),


```
@descripcion_producto varchar(30),
@cantidad_producto int,
@precio_producto numeric(14,2),
@descuento_producto numeric(14,2),
@iva numeric(14,2),
@ice numeric(14,2),
@irbpnr numeric(14,2),
@total_producto numeric(14,2)
)
AS insert into
Producto values
(@codigo_producto, @descripcion_producto, @precio_producto,
@cantidad_producto, @descuento_producto, @iva, @ice, @irbpnr,
@total_producto)

create procedure salidaproducto(
@cantidad_producto int,
@codigo_producto varchar(6)
)
as update Producto set cantidad_producto=cantidad_producto-@cantidad_producto
where codigo_producto=@codigo_producto;
```

```
create proc insretencion(
@ruc varchar(15),
@nombre varchar(50),
@razon_social varchar(50),
@direccion_matriz varchar(50),
@direccion_empresa varchar(50),
@tipo_comprobante varchar(30),
@ambiente varchar(30),
@emision varchar(30),
@numero varchar(30),
@fecha varchar(30),
@clave varchar(20),
@autorizacion varchar(50),
@ruc_comprador varchar(50),
@razon_comprador varchar(30),
@tipo_documento varchar(20),
@fecha_emision varchar(30),
@numero_comprobante varchar(30),
@adicional_uno varchar(50),
@adicional_dos varchar(50),
@adicional_tres varchar(50),
@id_facturap int
)
AS insert into
Retencion values (@ruc, @nombre, @razon_social, @direccion_matriz,
@direccion_empresa, @tipo_comprobante, @ambiente, @emision,
```

@numero, @fecha, @clave, @autorizacion, @ruc_comprador, @razon_comprador,
@tipo_documento, @fecha_emision, @numero_comprobante, @adicional_uno,
@adicional_dos, @adicional_tres, @id_facturap)

```
create proc updfacturap(  
  @id_facturap int,  
  @ci_proveedor varchar(15),  
  @razon_proveedor varchar(30),  
  @direccion_matriz varchar(50),  
  @direccion_empresa varchar(50),  
  @fecha_factura varchar(30),  
  @subtotal_factura numeric(14,2),  
  @sub_12 numeric(14,2),  
  @sub_0 numeric(14,2),  
  @sub_no_iva numeric(14,2),  
  @sub_excento numeric(14,2),  
  @descuento_factura numeric(14,2),  
  @ice numeric(14,2),  
  @irbpnr numeric(14,2),  
  @iva_factura numeric(14,2),  
  @otros_factura numeric(14,2),  
  @total_factura numeric(14,2),  
  @forma_pago_factura varchar(15)  
)  
as update FacturaProveedores set fecha_factura=@fecha_factura,  
  subtotal_factura=@subtotal_factura,  
  sub_12=@sub_12, sub_0=@sub_0, sub_no_iva=@sub_no_iva,  
  sub_excento=@sub_excento,  
  descuento_factura=@descuento_factura, ice=@ice, iva_factura=@iva_factura,  
  irbpnr=@irbpnr,  
  otros_factura=@otros_factura, total_factura=@total_factura,  
  forma_pago_factura=@forma_pago_factura,  
  ci_proveedor=@ci_proveedor, razon_proveedor= @razon_proveedor,  
  direccion_matriz=@direccion_matriz, direccion_empresa = @direccion_empresa  
  where id_facturap=@id_facturap;
```

```
create proc updproducto(  
  @cantidad_producto int,  
  @codigo_producto char(6)  
)  
as update Producto set cantidad_producto=cantidad_producto+@cantidad_producto  
  where codigo_producto=@codigo_producto;
```

A.08 Manual de Instalación

INSTALACIÓN DE SQL SERVER 2008

Para empezar la instalación primeramente buscamos el archivo de instalación del producto y damos doble clic sobre el para comenzar la instalación.

Seleccionamos en la pantalla principal la opción instalación.

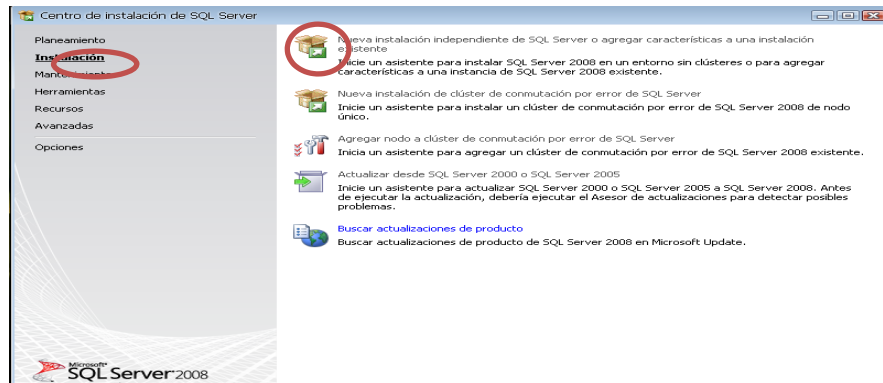


Figura 82: Instalación SQL server 2008. Paso N°1

En la siguiente ventana se cargara todas las reglas auxiliares del producto de SQL Server 2008 que son necesarias para su instalación.

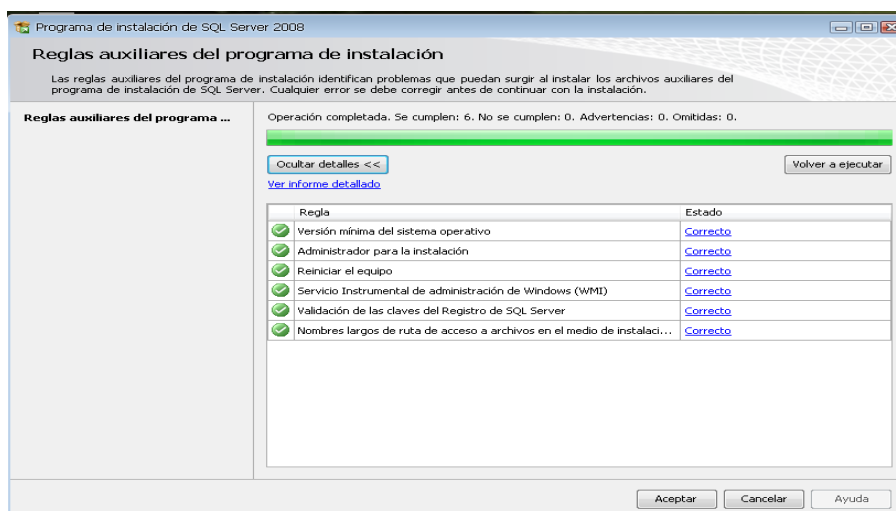


Figura 83: Instalación SQL server 2008. Paso N°2

En la siguiente ventana nos aparecerá la opción de instalación solo hacemos clic en el botón instalación para instalar algunos archivos auxiliares del programa de Instalación.

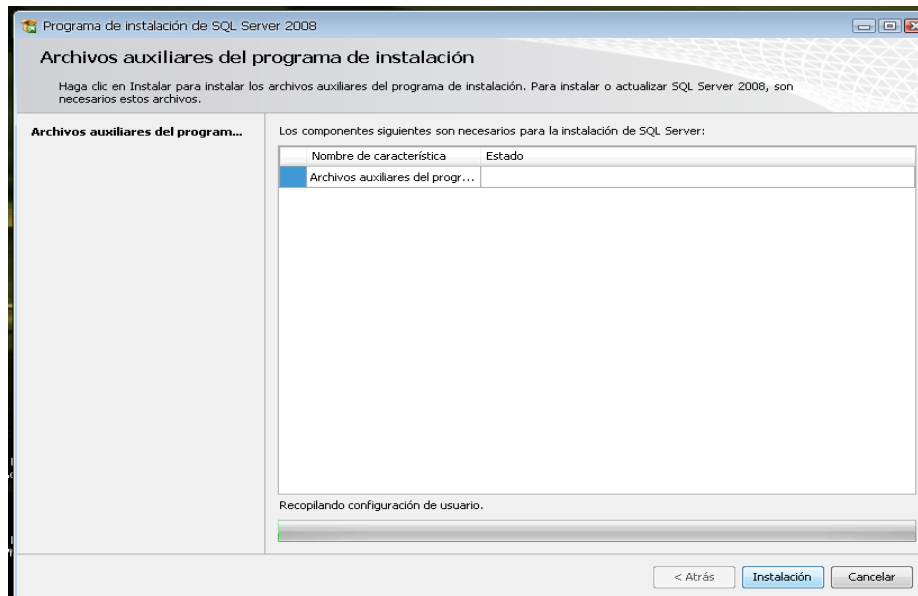


Figura 84: Instalación SQL server 2008. Paso N°3

La siguiente ventana visualizamos la clave del producto y nos dan dos opciones, tenemos que seleccionar la segunda opción y dar clic en siguiente.

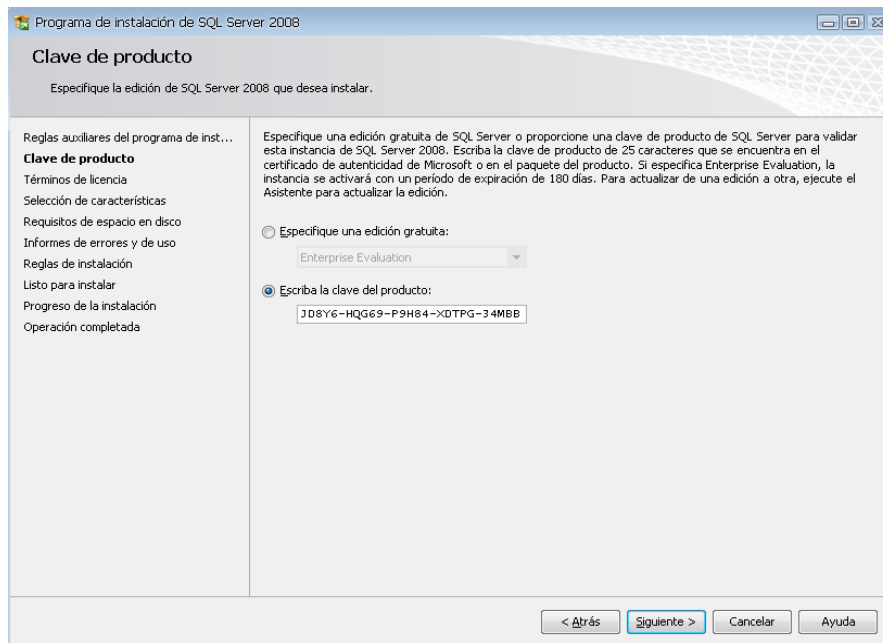


Figura 85: Instalación SQL server 2008. Paso N°4

En la siguiente ventana podemos visualizar que nos pide que aceptemos los términos de licencia del producto

Damos clic en la parte inferior izquierda en donde dice “Acepto los términos de licencia” y damos clic en siguiente.



Figura 86: Instalación SQL server 2008. Paso N°5

La siguiente ventana que nos muestra son las características que queremos agregar a la instalación, lo recomendable es seleccionar todas las características y luego clic en Siguiente.

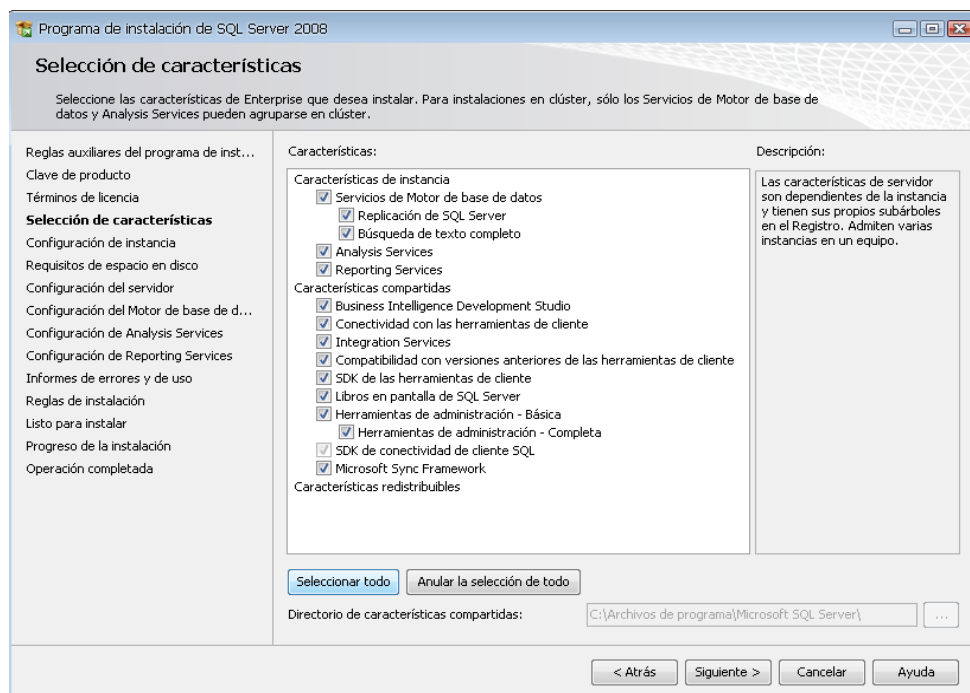


Figura 87: Instalación SQL server 2008. Paso N°6

La siguiente ventana es sobre la configuración de la instancia de SQL Server 2008, seleccionamos la primera opción y luego clic en siguiente.

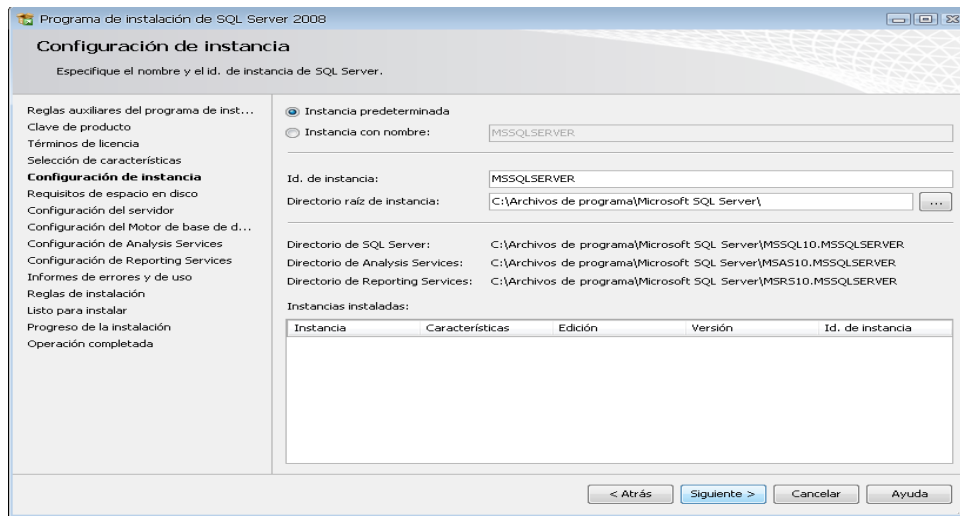


Figura 88: Instalación SQL server 2008. Paso N°7

En la siguiente ventana configuramos el servidor y para eso nos pide una cuenta de usuario, lo cual podemos crear un usuario específico para la instalación SQL, o de lo contrario podemos usar el usuario actual. En este caso usaremos "NT AUTHORITY\Servicio de red", luego damos clic en la pestaña "INTERCALACION".

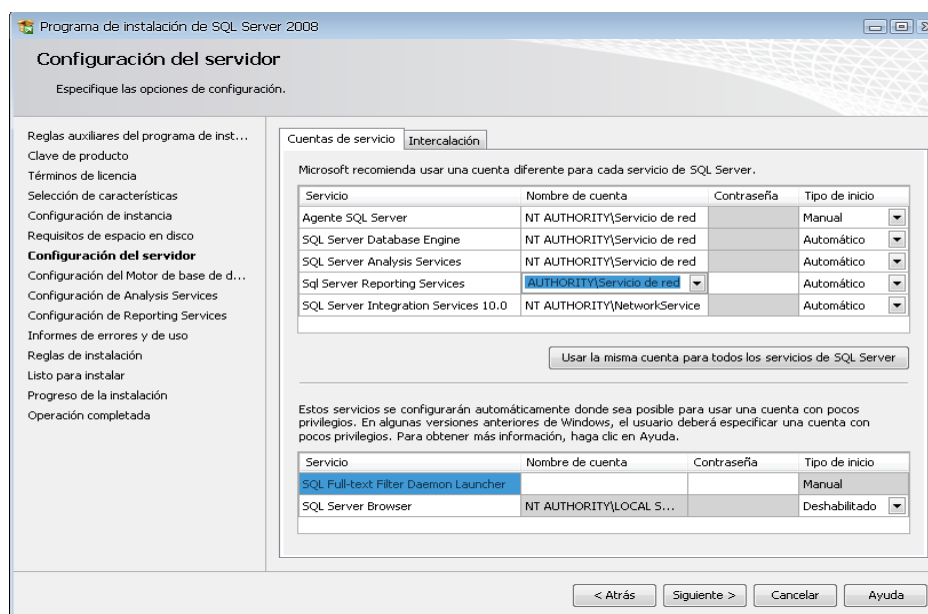


Figura 89: Instalación SQL server 2008. Paso N°8

En esta pestaña de Intercalación configuramos el “motor de Base de Datos” y el “Servicio de Análisis”, para lo cual hacemos clic en el botón Personalizar en cada uno. Luego lo personalizamos como nos muestra las figuras “Motor de Base de Datos” y “AnalysisServices” y luego hacemos clic en siguiente.

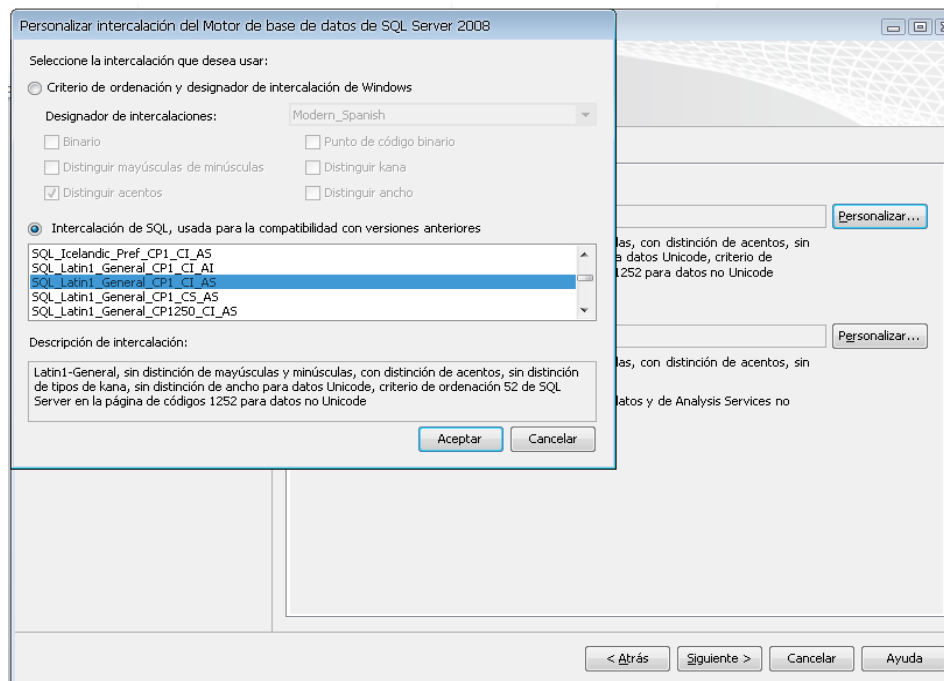


Figura 90: Instalación SQL server 2008. Paso N°9

Luego la siguiente ventana que nos muestra es la configuración del motor de base de datos, nos dan 2 opciones modo autenticación de windows y modo mixto (autenticación de sql server y de windows). Lo recomendable es seleccionar el modo mixto, al seleccionar esta opción se activan las cajas de contraseña para el inicio de sesión de autenticación de SQL Server (inicio de sesión de sa). Luego nos pide que agreguemos los administradores de SQL Server para lo cual agregamos el usuario actual; luego clic en siguiente.

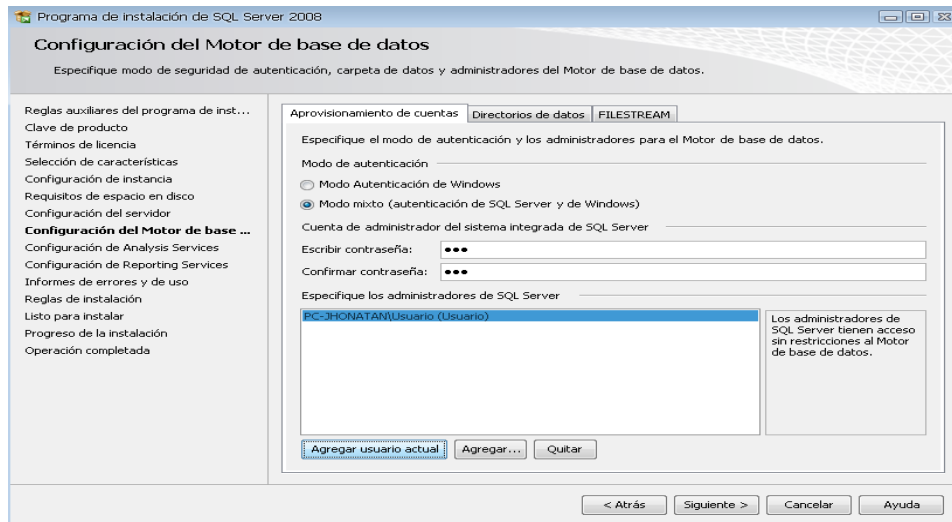


Figura 91: Instalación SQL server 2008. Paso N°10

Luego la siguiente ventana a mostrar son reglas que nos piden para poder INSTALAR SQL SERVER. Si todas las reglas están en estado **Correcto** hacer clic en siguiente y luego hacemos clic en instalación y esperamos que termine la instalación, luego clic en siguiente y finalmente en cerrar.

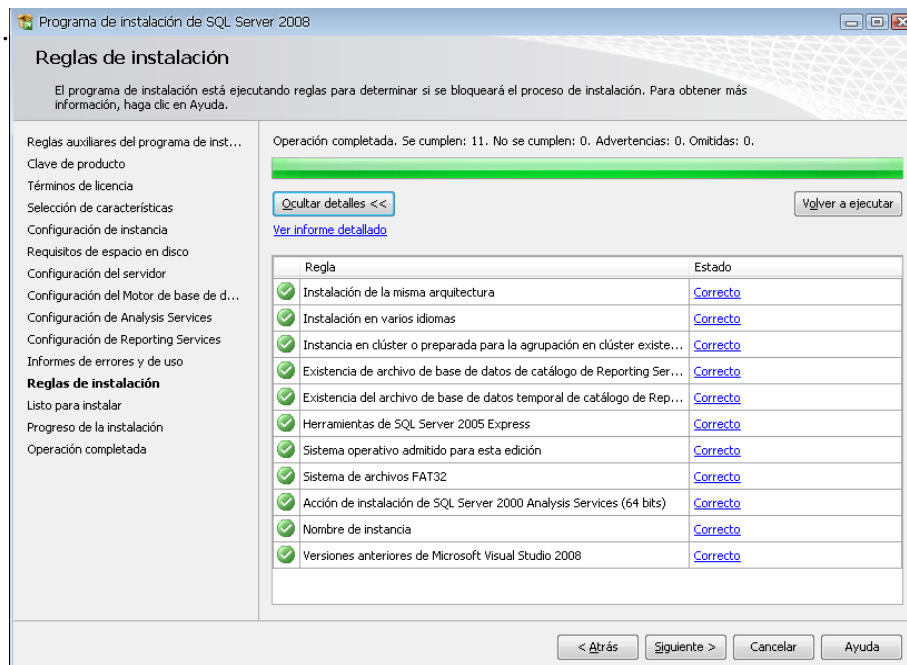


Figura 92: Instalación SQL server 2008. Paso N°11

En la opción Autenticación seleccionamos “Autenticación de SQL Server”, en inicio de sesión “sa” y escribimos la contraseña (contraseña que escribimos cuando configuramos el Motor de Base de Datos).

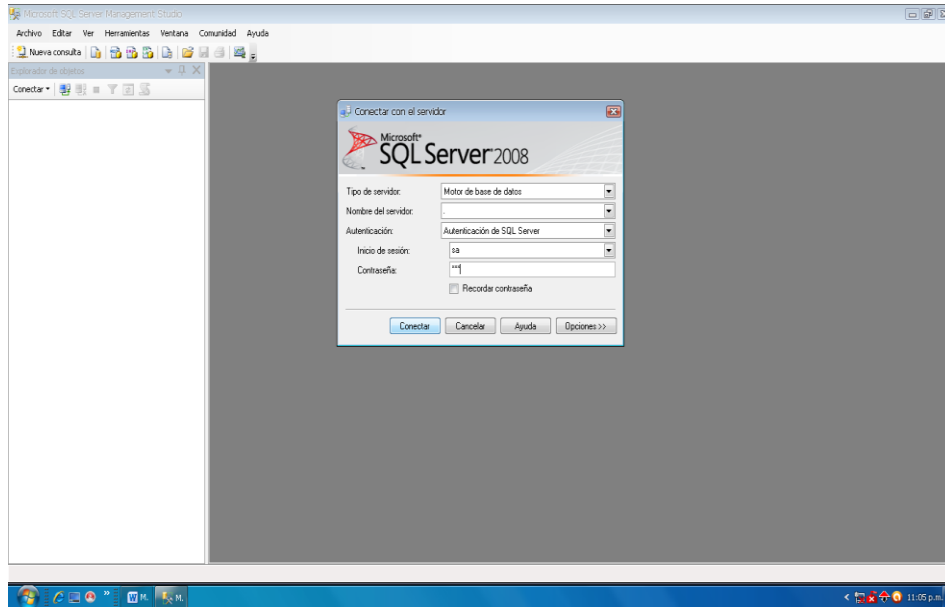


Figura 93: Instalación SQL server 2008. Paso N°12

INSTALACIÓN DE VISUAL STUDIO 2013

Lo primero es navegar a la página de descarga de VS 2013, pulsamos “Sign int to download” para acceder a la página de descarga.

En la página de descarga, la primera novedad que nos encontramos es que podemos indicar información de nuestro perfil y que desde la misma podremos acceder a un listado de descargas de versiones de VS 2013 disponibles: Ultimate, Premium, Professional, Test y por supuesto Team Foundation Server 2013 Preview. Hacemos clic en la versión Ultimate.

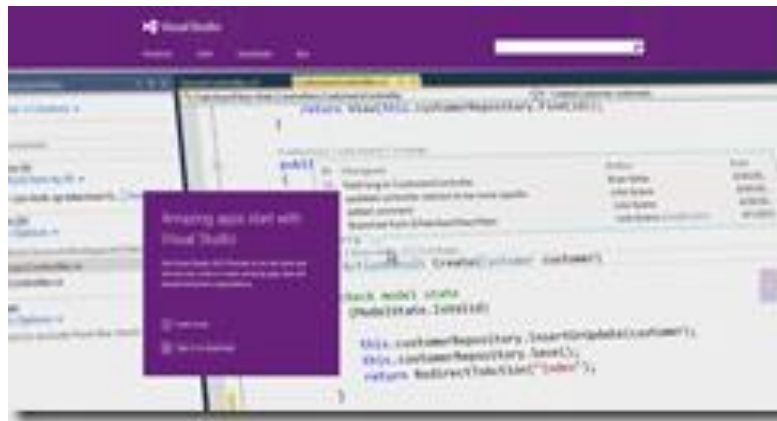


Figura 94: Página Visual Studio .Página web de bienvenida de Visual Studio alojada en la red para descargas.

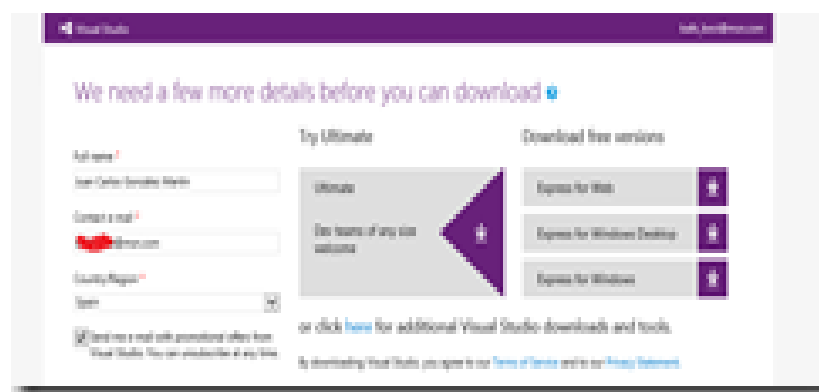


Figura 95: Login Visual Studio .Página web descarga Visual Studio nos muestra que tipo de visual studio deseamos descargar.



Figura 96: Descarga Visual Studio .Pagina descarga Visual Studio una vez escogida la versión de visual studio nos muestra el link de descarga.

Aceptamos el correspondiente acuerdo de licencia y pulsamos “Next”.

A continuación, se muestran los elementos que tenemos disponibles para instalar como parte de VS 2013 Preview. Por defecto, están seleccionados todos los elementos. Pulsamos el botón “Install”.

A partir de aquí VS 2013 Preview se toma su tiempo para instalar.

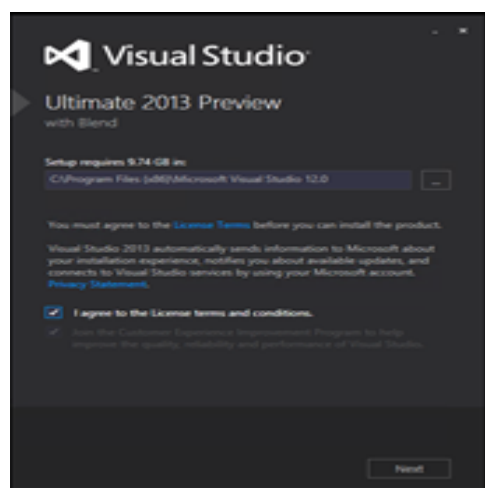


Figura 97: Contrato y Disco. Contrato y lugar en el disco a instalar Visual Studio donde el usuario puede escoger cualquier lugar en su disco.



Figura 98: Herramientas. Herramientas a instalar en Visual Studio nos muestra que herramientas deseamos instalar el usuario puede marcar o desmarcar según sus necesidades.

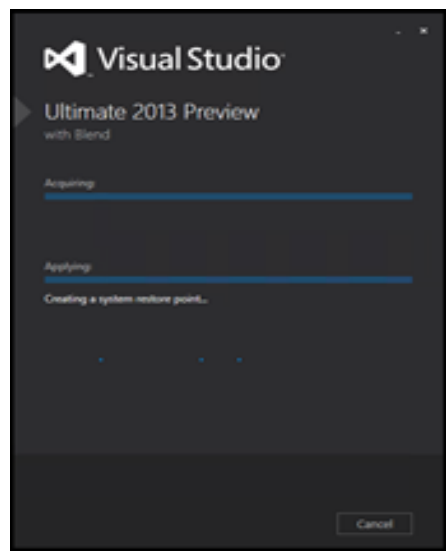


Figura 99: Instalación de Visual Studio muestra el progreso de la instalación esto puede tardar varios minutos.

Cuando el proceso de instalación concluye, es necesario re-iniciar el equipo.

Una vez que el equipo está de nuevo operativo, arrancamos VS 2013 Preview.

A continuación, se muestra la pantalla de SignIn de VS 2013 Preview, novedad en esta nueva versión del IDE.

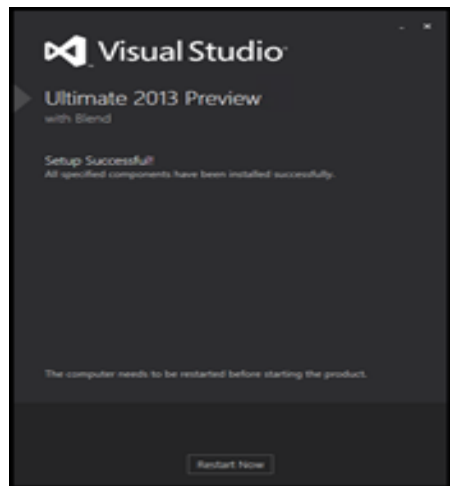


Figura 100: Reiniciar Visual Studio nos muestra la pantalla que nos dice que debemos reiniciar la aplicación antes de iniciar por primera vez.

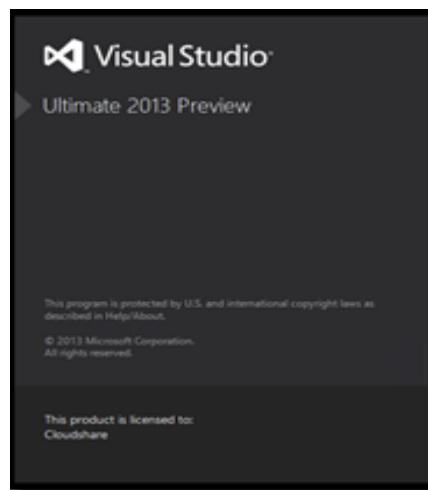


Figura 101: Inicio Visual Studio es la pantalla de bienvenida.

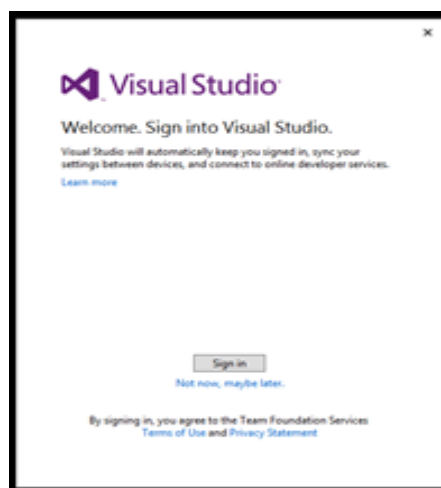


Figura 102: Pagina de Bienvenida de Visual Studio esta pantalla es para cargar si tenemos una cuenta Microsoft para ingresar logeados de no poseer ingresar directamente.

Especificamos las credenciales de nuestro Live ID para ingresar”.

Antes de “arrancar” VS 2013 Preview, lo personalizamos a nuestro gusto escogiendo la configuración de desarrollo por ejemplo y el tema a aplicar al IDE. Pulsamos “Start Visual Studio” y esperar unos minutos a tener el IDE listo.

Finalmente, la versión del IDE aparecerá con la configuración escogida y lista para que empecemos a utilizar.

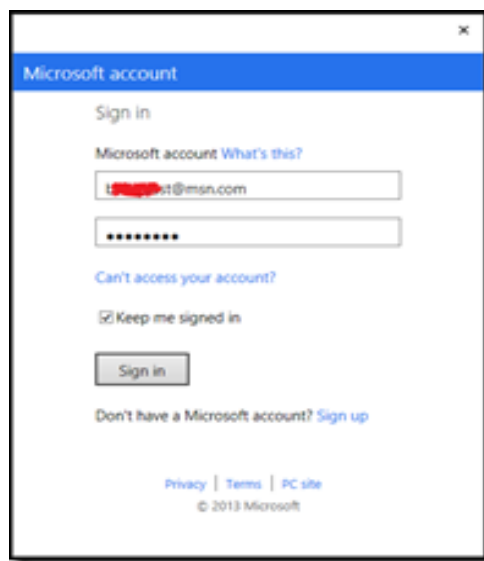


Figura 103: Login de Visual Studio donde nos muestra el correo y contraseña a ingresar.

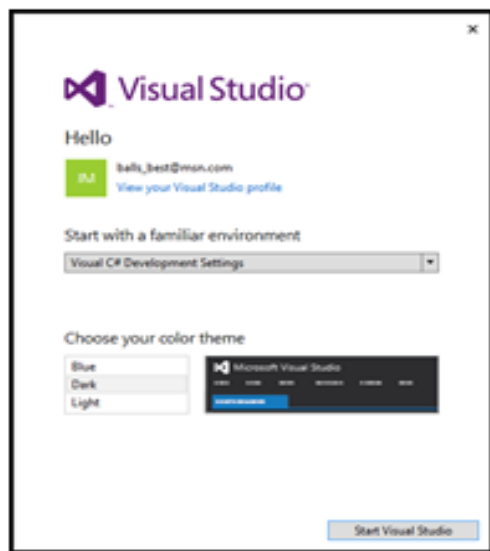


Figura 104: Plataforma a usar de Visual Studio escoger la instancia a ejecutar de visual studio.

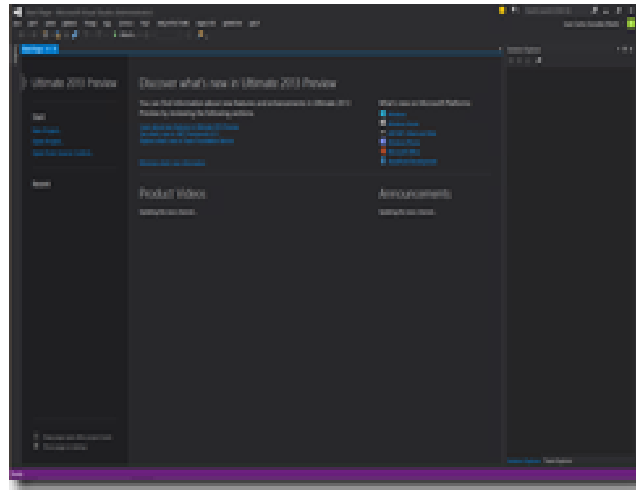


Figura 105: Página de inicio de Visual Studio esta página es la principal del Visual studio ya ejecutándose.

BIBLIOGRAFÍA

- santiago, C. d. (04 de Abril de 2008-2009). *Factura electrónica en chile*. Obtenido de http://www.sii.cl/factura_electronica/camara_comercio.pdf
- León, C. (08 de Abril de 2007). *Análisis de alternativas*. Obtenido de http://www.eumed.net/libros-gratis/2007a/232/analisis_alternativas.html
- Arregui, M. (06 de Octubre de 2004). *Tutorial de Uml*. Obtenido de http://www.conganat.org/SEIS/inforsalud04/2004_Inforsalud_TutorialUML-UP.doc
- Bonaparte, U. J. (05 de Enero de 2012). *Proyectos UML Diagramas de clases y aplicaciones*. Obtenido de http://www.edutecne.utn.edu.ar/tutoriales/uml_JAVA.pdf
- Silberschatz, A. (Bell L., Korth, H. F. (Bell L., & Sudarshan, S. (Instituto Indio de Tecnología, B. (2002). *Fundamentos de bases de datos*. Victoria.
- Tichelaar, S., & Demeyer, S. (1999). SNiFF+ Talks to Rational Rose --- Interoperability using a Common Exchange Model. In *SNiFF Users Conference*.
- AREVALO, M. E. (08 de Septiembre de 2014). *Estandares de Programación*. Obtenido de <http://ri.ues.edu.sv/6292/1/tesis%20completa%202.pdf>
- B., I. A. (09 de Marzo de 2010). *Pruebas Unitarias I*. Obtenido de http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php
- PEDILLA, E. (29 de Julio de 2013). *ARQUITECTURA*. Obtenido de <http://eyssel.blogspot.com/2013/07/arquitectura-de-software-toda.html>
- JIMENEZ, J. (06 de Marzo de 2013). *ESTRUCTURA D EDATOS*. Obtenido de http://jcjg868809.blogspot.com/2013_02_01_archive.html
- Morrow, J. (12 de Noviembre de 2014). *Scrum-QA Pruebas de Sistemas*. Obtenido de <http://scrum-qa.blogspot.com/2013/03/pruebas-no-funcionales-k2-entender.html>