



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS Y SISTEMAS

ORGANIZACIÓN DEL PROCESO DE FACTURACION MEDIANTE
UNA APLICACIÓN INFORMÁTICA PARA EL INSTITUTO DE
CAPACITACION CENDOCAP DE LA CIUDAD DE QUITO.

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

Autor: Amado Fernando Nogales Bastidas

Tutor: Prof. Luis Ríos

Quito, Marzo 2014

DECLARACIÓN DE APROBACIÓN TUTOR

En mi calidad de tutor del trabajo sobre el tema: “ORGANIZACIÓN DEL PROCESO DE FACTURACION MEDIANTE UNA APLICACIÓN INFORMÁTICA PARA EL INSTITUTO DE CAPACITACION CENDOCAP DE LA CIUDAD DE QUITO.”, presentado por el ciudadano: Amado Fernando Nogales Bastidas, 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 Distinguido Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Marzo de 2014

Prof. Luis Ríos

TUTOR

CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante AMADO FERNANDO NOGALES BASTIDAS, 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 “ORGANIZACIÓN DEL PROCESO DE FACTURACIÓN MEDIANTE UNA APLICACIÓN INFORMÁTICA PARA EL INSTITUTO DE CAPACITACIÓN CENDOCAP DE 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 reconvención, 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 21 días del mes de Abril del dos mil catorce.

f) _____

C.C. 172188037-3

CEDENTE

f) _____

Instituto Superior Tecnológico Cordillera

CESIONARIO

DECLARATORIA

Declaro que el estudio de esta investigación es completamente original, único, personal, que aquí he citado las fuentes principales que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor. Las opiniones, métodos, resultados y conclusiones a los que he llegado son de mi incondicional compromiso.

.

Amado Fernando Nogales Bastidas

C.C. 172188037-3

CESIÓN DE DERECHOS

Yo, Amado Fernando Nogales Bastidas Alumno de la Escuela de Análisis y Sistemas, libre y voluntariamente cedo los derechos de Autor de mi investigación en favor del Instituto Tecnológico Superior "Cordillera".

Amado Fernando Nogales Bastidas

C.C. 172188037-3

AGRADECIMIENTO

Primeramente agradezco a Dios, por darme salud, inteligencia, perseverancia y Amor. A mis Padres, por brindarme todo su apoyo económico, moral, entregarme todo su cariño y su confianza. A mi Esposa, por ser la persona que ha llenado mi vida de amor, certeza, seguridad y comprensión. A mis Hermanos por estar ahí cuando los necesitaba apoyándome dándome fuerzas coraje y valentía, gracias a todos he podido llegar a cumplir mi proyecto de vida que años antes me avía planteado llegar a graduarme llegar a tener un título más.

Agradezco a todas las personas con las que he podido compartir esta etapa tecnológica de los cuales he extraído muchos conocimientos logrando aprender cada día más.

Agradezco a mi tutor el Profesor Luis Ríos por el apoyo brindado en todo momento. Agradezco a mi Lector el Ingeniero Jaime Padilla por el apoyo incondicionalmente en la culminación de mi tesis de grado.

A los amigos y compañeros que compartieron conmigo la vida institucional y fueron partícipes de cada experiencia, gracias por permitirme aprender de ustedes.

Amado Fernando Nogales Bastidas

DEDICATORIA

A mi Esposa Joselin Calderón;
que es la principal fundamento de este logro;
a mis Padres Raúl Nogales y María De Lourdes Bastidas;
a todos mis familiares y amigos
que se han convertido en el umbral
para poder lograr este sueño tan grande
y cumplir mi objetivo.

ÍNDICE GENERAL

Contenido	Pág.
Declaración de Aprobación del Tutor	ii
Contrato de cesión sobre derechos propiedad intelectual	iii
Declaratoria	vi
Cesion de Derechos	vii
Agradecimiento	viii
Dedicatoria	ix
Resumen Ejecutivo	xvi
Abstract.....	xvii
Capítulo I: Antecedentes	1
1.01 Contexto	1
1.02 Justificación e Importancia.....	2
1.03 Definición de problema central	4
Capítulo II: Análisis de Involucrados	5
2.01 Mapeo de Involucrados	5
2. 02 Matriz de Análisis de los involucrados	6
Capítulo III: Problemas y Objetivos.....	7
3.01 Árbol de Problemas	7
3.02 Árbol de Objetivos	8
Capítulo IV: Análisis de Alternativas.	9
4.01 MATRIZ DE ANÁLISIS DE ALTERNATIVAS	9
Tabla 4.01.01 Matriz de Análisis de Alternativas	9
4.02 Matriz de Análisis de Impacto de los Objetivos	10
4.03 Diagrama de Estrategias	11
4.04 Matriz de Marco Lógico.....	12
Capítulo V: Justificación Propuesta.....	13
5.01 Justificación del Software	13
5.02 Análisis y Diseño	14
5.02.01 Casos de Uso	14
5.02.02 Diagrama de Caso de Uso General	14
5.02.02 Diagramas de Realización.....	15
5.02.03 Diagramas de Secuencias.....	20
5.02.04 Diagrama de Colaboración	24

	Pág.
5.02.05 Diagrama de Componentes	27
5.02.06 Diagrama de Clases.....	31
5.02.07 Diagrama Lógico	32
5.03 Desarrollo.....	33
5.03.01 Arquitectura Del Sistema.....	33
5.03.01.01 Capa de Presentación	33
5.03.01.02 Capa de Negocio	33
5.03.01.03 Capa de Datos	34
5.03.01.04 Capa de Servidor Web	34
5.03.01.06 Módulos	35
5.03.01.06.01 Módulo de Seguridad.....	35
5.03.01.06.02 Módulo Mantenimiento	36
5.03.01.06.03 Módulo Lógica Negocios.....	36
5.03.01.06.04 Módulo de Aprobación y generación de Factura.....	36
5.03.02 Estándares.....	36
5.03.02.01 Estándares de Diseño UML	36
5.03.03.01 Estándares de Programación	39
Tabla 5.03.03.01 Estándar de Base de Datos	42
Tabla 5.03.03.02 Atributos.....	42
5.03.03.03 Tipos de Datos	42
5.03.04 Diseño de Interfaces	45
5.04 Pruebas	47
5.04.02. Pruebas de módulo.	47
5.04.02.01. Módulo de Seguridad.....	47
5.4.2.2. Módulo Mantenimiento	48
5.4.3. Pruebas de Interfaz de Usuario.	49
5.4.4 Pruebas de Desempeño.	50
5.4.5. Pruebas de Carga.....	51
5.4.6 Pruebas de Validación.....	51
Capítulo VI: Aspectos Administrativos.....	52
6.01 Recursos.	52
6.02 Presupuesto.....	53
6.03 Cronograma.....	54
Capítulo VII: Conclusiones y Recomendaciones	55

	Pág.
7.01 Conclusión.....	55
7.02 Recomendación	55
ANEXOS.....	56
A.01 Instalación de programas utilizados	57
A.01 Instalación del SQL Server Management.....	57
A.02 Instalación de Visual Studio 2010.....	64
A.02 Script de la Base de Datos	69
A.03 Diccionario de Datos	81
A.04 Manual Técnico.....	87
A.05 Manual de Usuario	96
GLOSARIO.....	106
WEBGRAFÍA	108

ÍNDICE TABLAS

	Pág.
Tabla 1.03.01 Matriz de Análisis de Involucrados	4
Tabla 2.02.01: Matriz de Análisis de los involucrados	6
Tabla 4.01.01: Matriz de análisis de alternativas	9
Tabla 4.02.01: Matriz de Análisis de Impacto de los Objetivos	10
Tabla 5.02.02.01: Despacho de cliente.	15
Tabla 5.02.02.02 Informe de Matriculación.....	16
Tabla 5.02.02.03 Entrega de Recibo	17
Tabla 5.02.02.04 Matriculas Pérdidas.....	18
Tabla 5.02.02.05 Matricula Exitosa	19
Tabla 5.02.02.06 Registro de Recibos	20
Tabla 5.03.02.02 Elementos de Diseño.....	37
Tabla 5.03.02.03 Diagramas de Diseño	38
Tabla 5.03.02.03.02 Atributos	45
Tabla 5.03.02.03.03 Tipo de Datos	45
Tabla 5.03.02.03.05 Ingreso de un Nuevo Socio	45
Tabla 5.04.07: Recursos	52
Tabla 5.04.08: Recurso Material	52
Tabla 6.02.01 Presupuesto para la realización del proyecto.	53

INDICE DE FIGURAS

	Pág.
Figura 1 Mapeo de Involucrados.....	5
Figura 2 Árbol de Problemas	7
Figura 3 Árbol de Objetivos.....	8
Figura 4 Diagrama de Estrategias	11
Figura 5 Matriz de Marco Lógico	14
Figura 6 Diagrama de Caso de Uso General.....	15
Figura 7 Diagramas de Realización 1	15
Figura 8 Diagramas de Realización 2	17
Figura 9 Diagramas de Realización 3	18
Figura 10 Diagrama de Realización 4.....	18
Figura 11 Matricula exitosa	23
Figura 12 Registro de Recibos	23
Figura 13 Diagrama de Colaboración 1	21
Figura 14 Diagrama de Colaboración 2	25
Figura 15 Diagrama de Colaboración 3	26
Figura 16 Diagrama de Colaboración 4	26
Figura 17 Diagrama de Colaboración 5	24
Figura 18 Diagrama de Colaboración 6	27
Figura 19 Diagrama de Componentes	28
Figura 20 Diagrama de Componentes 2.....	29
Figura 21 Diagrama de Componentes 3.....	30
Figura 22 Diagrama de Clases	31
Figura 23 Modelo Lógico	32
Figura 24 Arquitectura Lógica	35
Figura 25 Diagrama de Componentes 3.....	36
Figura 26 Diagrama de Componentes 4.....	36
Figura 27 Modelo Lógico.....	38
Figura 28 Arquitectura Lógica.....	38
Figura 29 Ingreso al sistema	45
Figura 30 Ingreso de un Nuevo Alumno.....	46
Figura 31 Modificación de Alumnos	47
Figura 32 Cronograma de actividades.....	54
Figura 33 Ingreso General de acceso del sistema	95
Figura 34 Menú de Login de Usuario	35
Figura 35 Menú de Opciones del Usuario.....	96
Figura 36 Registros de Empleados.....	97
Figura 37 Registro de Nuevo Empleado	97
Figura 38 Modificar empleado.....	98
Figura 39 Registro de Nuevo Empleado	98
Figura 40 Modificar Cliente.....	99
Figura 41 Matricula.....	99
Figura 42 Registro de Nueva Matricula	100
Figura 43 Modificar de Matricula	100
Figura 44 Menú de Opciones del Administrador.....	101

	Pág.
Figura 45 Maestro de Institutos.....	101
Figura 46 Nuevo instituto	102
Figura 47 Periodos de clases	102
Figura 48 Creación de periodo de clases.....	103
Figura 49 Maestro de Ciudades	103
Figura 50 Registro de Ciudades	104

RESUMEN EJECUTIVO

El presente proyecto de tesis propone la organización del proceso de Facturación mediante una aplicación informática para el Instituto de Alta Capacitación CENDOCAP de la Ciudad de Quito. Este planteamiento es debido al inadecuado control de los ingresos y manejo de su documentación que lleva la institución.

Con la falta de una aplicación informática estandarizada ha aumentado el índice de pérdida de documentos, ingresos económicos, duplicidad de funciones estratégicas, desorganización de los colaboradores, demora en los procesos y procedimientos que el centro periódicamente desempeña.

El centro de capacitación ha visto la necesidad de automatizar los procesos de facturación ya que estos no se realizan con seguridad y agilidad. En este trabajo se presenta el desarrollo de módulos de software con fórmulas exactas, diciendo claramente que tendremos una interfaz en donde podremos visualizar los pagos de sus pensiones del cliente en una grilla; un módulo administrativo el cual nos permitirá manejar el proceso de entrega de recibos y facturas.

Con los avances de Software y Tecnología que hoy en día se están realizando las instituciones pueden poseer una aplicación informática de calidad que agilite todos los procesos es por eso que el centro de capacitación CENDOCAP ha optado por la implementación de este proyecto.

La principal ventaja de esta aplicación es que el Director General de la Institución puede tener una excelente toma de decisiones.

ABSTRACT

EXECUTIVE SUMMARY

This thesis project proposes the organization of the Billing process a computer application for the Institute for Advanced Training CENDOCAP City Quito. This approach is due to inadequate control of managing your income and documentation maintained by the institution.

With the lack of a standardized software application has increased the rate of loss of documents , income, duplication of strategic functions , disruption of reviewers , delay in the processes and procedures that the center plays regularly .

The training center has seen the need to automate billing processes der as these are not made with ease and agility. In this paper the development of software modules with exact formulas presented , clearly saying that we will have an interface where you can view their pension payments customer in a grid ; an administrative module which will allow us to manage the delivery process receipts and invoices.

With advances in software and technology today are being made institutions can have a quality software application that processes all agility is why the training center CENDOCAP has chosen to implement this project.

The main advantage of this application is that the Director General of the Institution may have an excellent decision.

CAPÍTULO I: ANTECEDENTES

1.01 Contexto

Centro de alta capacitación CENDOCAP Agente autorizado de New York Language Center Estados Unidos. Creada el 08 de octubre del año 1999 se creó firmemente con el fin de facilitar mejorar los conocimientos de todas las personas que deseen capacitarse. Contamos con 12 años de experiencia en el sector, con más de quinientos estudiantes nivelados y capacitados, logrando que el 98% de nuestros estudiantes consigan su objetivo y su meta propuesta.

CENDOCAP cuenta con dos extensiones, está ubicado en la ciudad Quito Parroquia Carapungo y Calderón tiene acceso a una amplia red de operaciones en los mercados dedicados a dar servicios de aprendizaje.

El centro actualmente maneja toda su información registro de clientes y toda la facturación manualmente motivo que existe procesos retardados y una demora en la entrega de la factura a sus clientes debido a este problema diagnosticado a través de las investigaciones realizadas.

El Instituto de Alta Capacitación CENDOCAP no queda indiferente al avance de la tecnología, es por eso que teniendo una visión a futuro, ha optado por la implementación de un software que le permitan automatizar sus principales procesos y el control de operaciones de facturación. La implementación de este software permitirá al centro de alta capacitación CENDOCAP tener un mejor control y seguimiento de los procesos mencionados, dando como resultado una información rápida y confiable, ahorrando tiempo y dinero. Los registros almacenados en la base de datos, son elementos específicos, es decir que la información ingresada es a base de los

requerimientos establecidos por la institución. Este sistema de información debe ser capaz de ayudar al desarrollo del centro de capacitación, así como aumentar su calidad de eficiencia en la enseñanza y competitividad en el mercado.

Actualmente el instituto se dedica a brindar servicios de capacitación y aprendizaje a todas las personas que buscan mejorar sus conocimientos a través de los cursos profesionales que se ofrecen en la institución.

Si no se realiza el proyecto de investigación afectara directamente al funcionamiento de la institución, se seguirá llevando la información manualmente como lo ha venido haciendo solo que no se permitirá mejorar, automatizar los procesos de facturación entregar un servicio más óptimo a sus clientes.

1.02 Justificación e Importancia

Según las investigaciones que se han logrado realizar en el Instituto de Alta Capacitación CENDOCAP, y analizando los resultados de los estudios hechos, nos demuestra que los clientes como el centro de capacitación posee un muy alto índice de pérdida de recursos como tiempo y dinero, motivo por el cual se ha optado por el desarrollo de una aplicación que dará solución a la insatisfacción de todos sus clientes, permitiendo al centro de alta capacitación brindar un servicio más rápido, oportuno y eficaz a los clientes aumentando muchas posibilidades de fidelidad por parte de los clientes y reducir en lo más mínimo el tiempo y el costo perdido.

El motivo de la realización de este proyecto es contribuir a la necesidad que tiene el cliente, desarrollando esta aplicación optimizaremos varios procesos que se ejecutan en el centro de alta capacitación manualmente, los beneficiados directos por este proyecto serán los usuarios por ellos pondrán en funcionamiento al sistema, y como

beneficiario directo será también el propietario centro de alta capacitación CENDOCAP, porque al desarrollar este software podemos agilizar muchos procesos, proporcionando comodidad y brindar así una mejor atención al cliente, será también beneficiario directo el Ministerio de Educación por que será quien haga cumplir las obligaciones de tipo legal y como beneficiarios indirectos tendremos toda nuestra distinguida clientela.

Realmente se hace muy indispensable la ingeniería de requerimientos, la cual desempeña un papel primordial en el proceso de producción de software, pues se encarga de definir claramente y con precisión lo que vamos a realizar, a través de esquemas, casos de uso entre otros, que permitan al administrador visualizar el funcionamiento del sistema que se elaborará, y demostrar si está de acuerdo con el diseño o si desea realizar modificaciones en el mismo, para luego, pasar a la etapa de elaboración del sistema.

Dando a conocer los tipos de impactos que tenemos en el proyecto, impactos sociales, impactos ambientales.

El impacto social diríamos que es un proyecto de desarrollo productivo que ayudara crecer y mejorar de una manera única al instituto CENDOCAP y en general a toda la sociedad. Así como permite un manejo eficiente de la información produciendo un mejor ambiente de trabajo y control para los empleados. Socialmente también por la mejora en la atención a los clientes del instituto reduciendo tiempos de espera y errores en la documentación emitida para estos.

1.03 Definición de problema central

Tabla 1.03.01

Matriz de análisis de fuerzas T

ANÁLISIS DE FUERZAS					
SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
Mal, manejo del personal encargado del proceso de facturación de estudiantes.	Inadecuado control del proceso de las actividades de facturación en el Centro de Alta Capacitación CENDOCAP.				Mejorar el control del proceso de facturación en todos los cursos abiertos.
FUERZAS IMPULSADORAS	I	PC	I	PC	FUERZAS BLOQUEADORAS
Cumplimiento con el manejo del nuevo sistema de facturación	2	4	2	4	Incumplimiento con el manejo del sistema de facturación.
Implementar tecnología actual para realizar el proceso de facturación.	2	4	3	4	Mal ingreso de la de información, registro de datos manualmente
Llevar estándares del proceso facturación apegada a todas las normas y leyes legales vigentes.	1	4	3	5	Desorganización de los procesos al instante de facturar.
Rapidez en el ingreso de información para un mejor servicio de facturación al cliente en el centro CENDOCAP.	1	4	4	5	Falta de tiempo, y rapidez.

Nota: En la tabla se explica las fuerzas impulsoras y bloqueadoras que existen con respecto a la problemática central del proyecto.

77CAPÍTULO II: ANÁLISIS DE INVOLUCRADOS

2.01 Mapeo de Involucrados

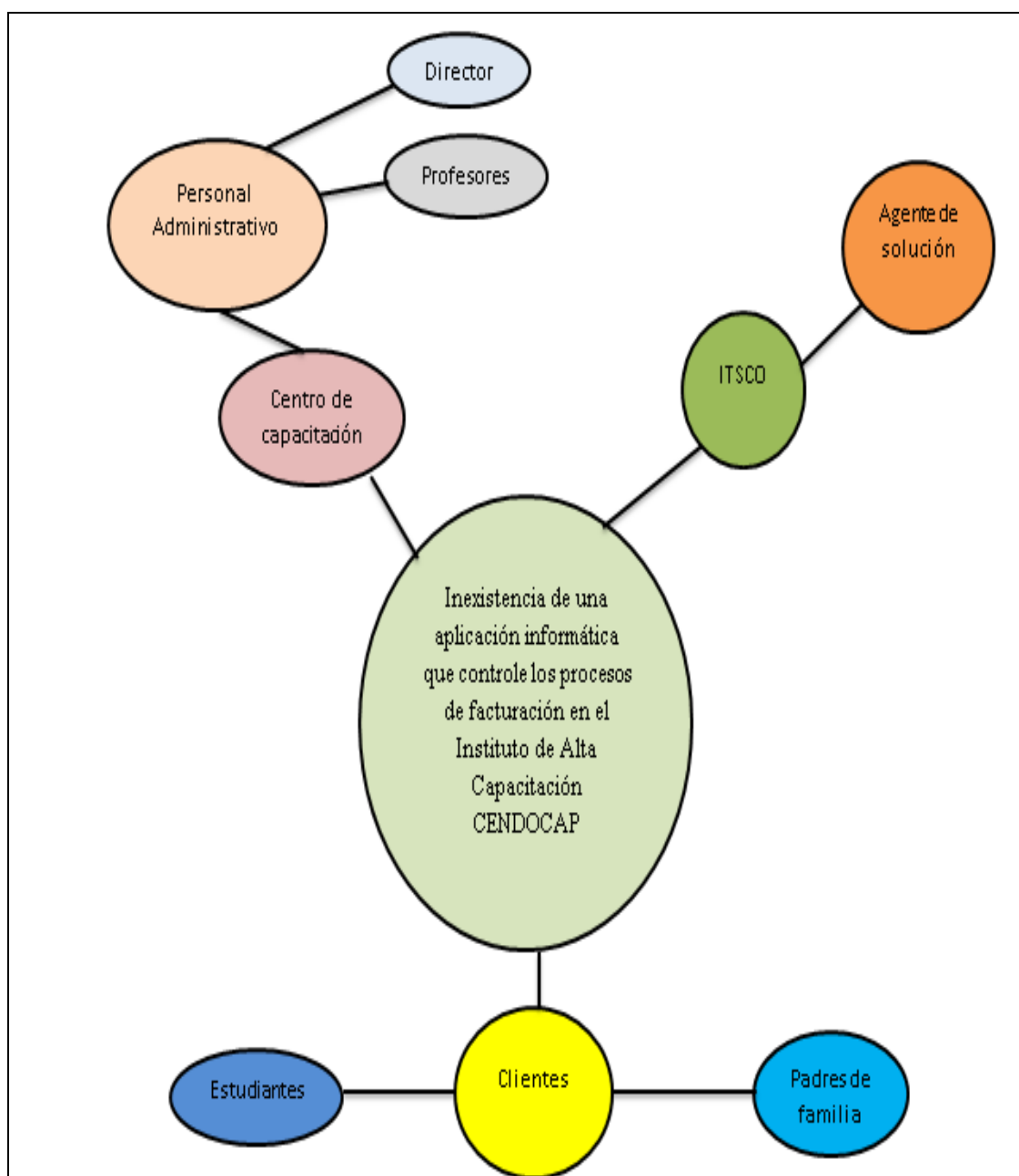


Figura 1 Mapeo de Involucrados

Nota: en la Grafica se explica los involucrados que pertenecen al proyecto. La aportación del sistema nos ayudara a un mejor mantenimiento preventivo y correctivo.

2. 02 Matriz de Análisis de los involucrados

ACTORES INVOLUCRADOS	INTERESES SOBRE EL PROBLEMA CENTRAL	PROBLEMAS PERCIBIDOS	RECURSOS, MANDATOS Y CAPACIDADES	INTERESES SOBRE EL PROYECTO	CONFLICTOS POTENCIALES
CENTRO DE ALTA CAPACITACION CENDOCAP	Brindar una atención de calidad a los clientes.	Demora en la entrega de factura al cliente.	Diseño e Infraestructura del Centro de alta capacitación CENDOCAP.	Agilidad en los procesos del centro de capacitación frente a sus clientes	Desorganización en la administración del centro
DIRECTOR	Mantener un control de los ingresos económicos al momento de la facturación	Pérdida de recursos	Mejorar todo sobre infraestructura a personal	Agilizar los procesos de facturación en la institución	Molestias en las actividades de facturación
PROFESORES	Mejorar conocimientos que un docente entrega a un alumno	Problemas de capacidad en las aulas	Cultura técnica, vocación del profesor al momento de educar	Optimizar el tiempo estimado	Examinar estrategias de control de estudiantes
ESTUDIANTES	Rapidez en el proceso de entregar la factura	Problemas existentes al entregar la factura	Compromiso a ser un alumno de calidad	Estudiantes complacidos por el proceso automatizado y rápido	Desorganización en diferentes procesos

Tabla 1.02.01 Matriz de Análisis de Involucrados

Análisis de la Matriz.

La Matriz Muestra los involucrados del proyecto. El inadecuado control de los procesos y actividades provoca en los involucrados un conflicto.

CAPÍTULO III: PROBLEMAS Y OBJETIVOS.

3.01 Árbol de Problemas

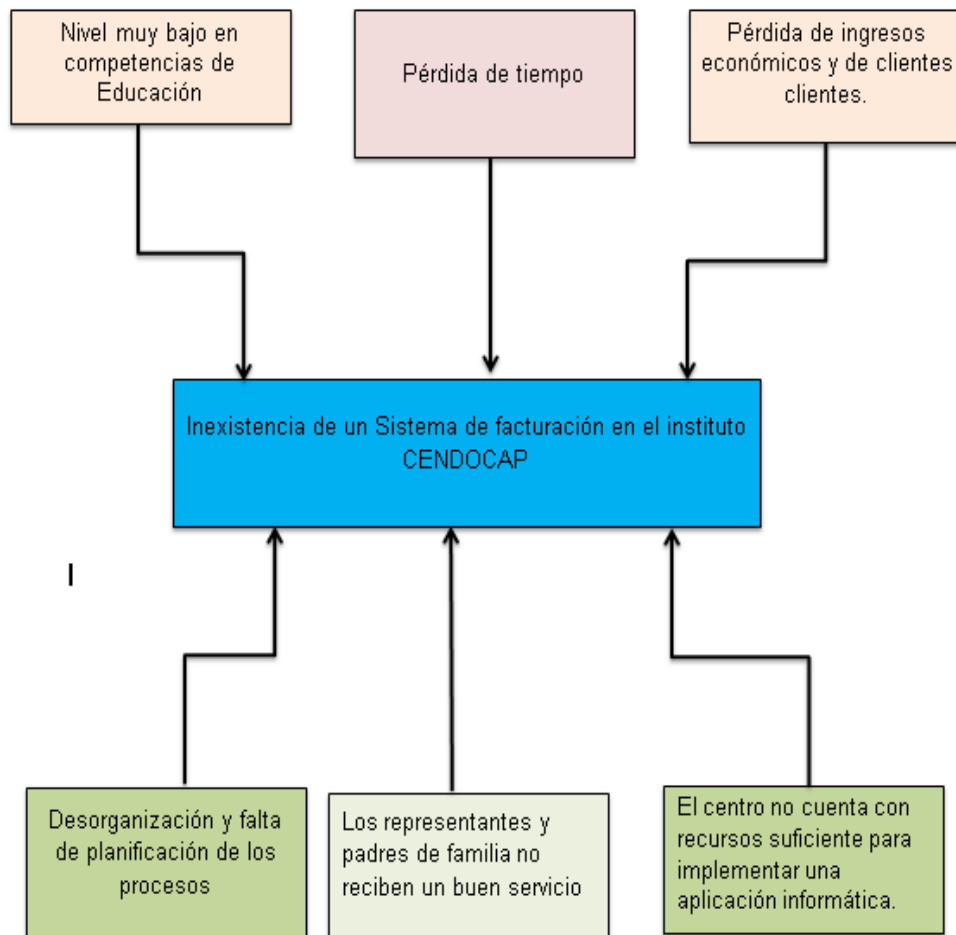


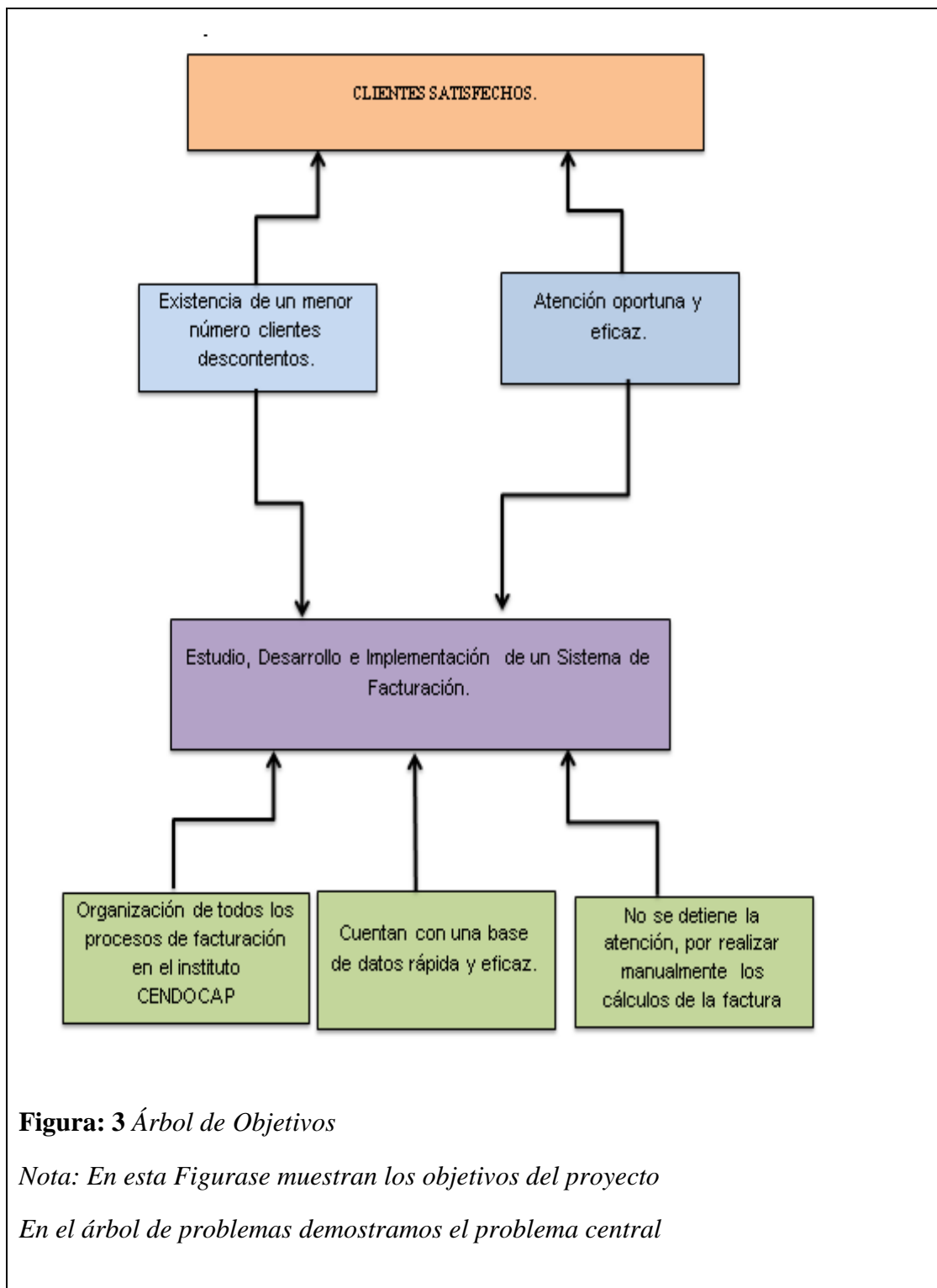
Figura 2 Árbol de Problemas

Nota: Esta Figura explica los problemas que existen con respecto al problema central con sus causas y efectos.

Descripción del Árbol de Problemas

En el árbol de problemas demostramos el problema central por el cual el Centro está atravesando cuales han sido las causas para que se origine y los efectos que ha producido. La demora en la entrega de recibos y facturas provoca pérdida de los ingresos. Estos inconvenientes amenazan pérdidas de clientes debido al mal manejo documental.

3.02 Árbol de Objetivos



CAPÍTULO IV: ANÁLISIS DE ALTERNATIVAS.

4.01 Matriz de análisis de alternativas

OBJETIVOS	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Se ha minimizado el tiempo de espera al realizar la factura	5	3	3	4	1	16	MEDIA ALTA
El proceso de facturación cumple políticas institucionales y leyes vigentes	4	4	3	4	1	16	MEDIA ALTA
Existencia de mejoras de relación entre representantes, padres y el personal administrativo	4	3	4	4	1	16	MEDIA ALTA
TOTAL	13	10	10	12	3	48	

Tabla 4.01.01 Matriz de Análisis de Alternativas

Análisis de la Matriz de Alternativas

El cuadro de alternativas tiene como propósito determinar las técnicas que permitan desarrollar las actividades, tomando en cuenta el recurso financiero, minimizando el costo ante el proyecto

4.02 Matriz de Análisis de Impacto de los Objetivos

4.2 MATRIZ DE ANALISIS DE IMPACTO DE LOS OBJETIVOS

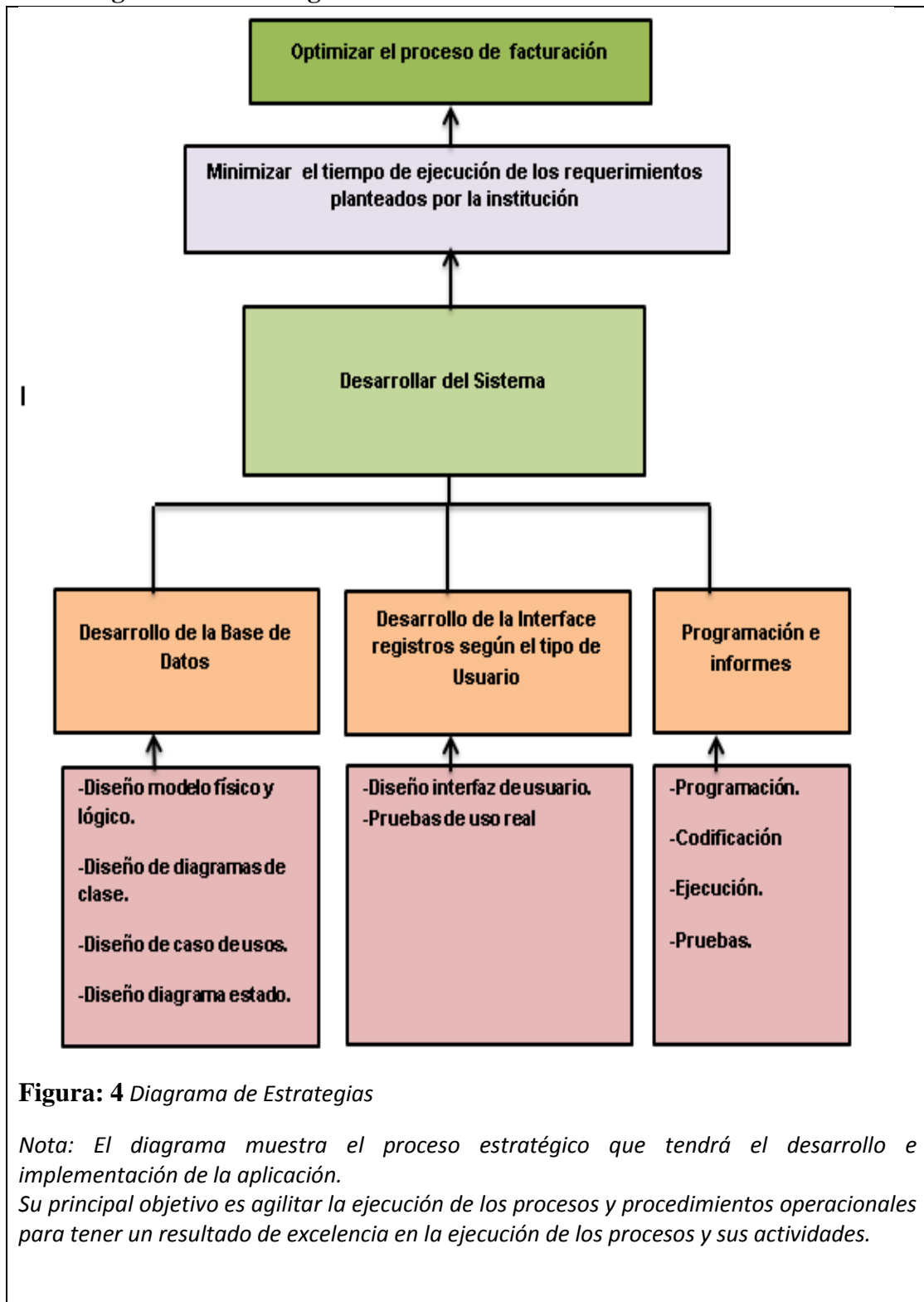
OBJETIVOS	Factibilidad de Lograrse (alta-media-baja)	Impacto de Género (alta-media-baja)	Impacto Ambiental (alta-media-baja)	Relevancia (alta-media-baja)	Sostenibilidad (alta-media-baja)	Total
	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	
	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	4 - 2 - 1	
Los beneficios son elevados ya que de esa forma baja los costos	No excluye a nadie por el género.	Reduce material físico.	-Los beneficios son los esperados.	-Fortalece el área de inscripción y facturación.	48 Puntos	
-Está financiado	-Realizable para todo género.	Disminuye tiempos de ejecución	-Beneficia a todos los clientes de la institución.	-Se mantiene fortalecida la institución.	7-17 BAJA	
-Existe tecnología para la realización	-Fácil manejo para toda persona.	-Favorece en educación de herramientas tecnológicas.			18-28 MEDIA BAJA	
					29-39 MEDIA ALTA	
	12 Puntos	12 Puntos	8 Puntos	8 Puntos	8 Puntos	40-50 ALTA

Nota: El cuadro muestra el Impacto que tendrá la realización del proyecto en base a los objetivos planteados.

Tabla 4.02.01 Matriz de Análisis de Impacto de los Objetivos

Nota: El cuadro muestra el análisis y el impacto que causa el desarrollo del proyecto

4.03 Diagrama de Estrategias



4.04 Matriz de Marco Lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
FIN DEL PROYECTO			
Incrementar la calidad de presentación en el área de facturación del instituto CENDOCAP	Se reduce toda la documentación que no es de mucha importancia para la institución	Resultados del levantamiento de información realizados en el instituto CENDOCAP Cronograma de desarrollo del sistema	El personal administrativo sigue apoyando a continuar con el proyecto y buscar más mejoras
PROPOSITO DEL PROYECTO	Se reduce los costos económicos en tecnología		
Automatización de las actividades y procesos de facturación	Se reducen los errores en los procesos de cálculos de facturas	Registros del rectorado seguimiento en los procesos de facturación	Se incrementa el interés y compromiso en el proyecto durante la implementación
	Se incrementa normas de seguridad		
COMPONENTES DEL PROYECTO			
Disminuir la inversión económica del desarrollo de la aplicación	Cambios en la imagen del instituto según estudios después de la implementación de sistema	Documentación de los procesos validados Manuales de Usuario y Técnicos	Organización de los procesos al elaborar la factura al cliente
Documentación realizada del sistema	Conjunto de documentación de los manuales de capacitación después del arranque del proyecto		Se optimiza tiempo y costos a la institución

Figura: 5 Matriz de Marco Lógico

Nota: El Cuadro muestra la secuencia lógica que lleva el proyecto desde su análisis hasta los costos de su implementación.

CAPÍTULO V: JUSTIFICACIÓN PROPUESTA

5.01 Justificación del Software

El actual proyecto está elaborado con el objetivo de analizar, definir y ejecutar los procesos y procedimientos, que estructura al sistema de agilización de los procesos de facturación con el fin de optimizar los recursos, tanto económicos como administrativos de las actividades que realiza el Centro de Alta Capacitación CENDOCAP.

Las aplicaciones a utilizarse en la elaboración del software de agilización de los procesos de Facturación son:

Servidor: SQL Server, modelamiento UML: Racional Rose, desarrollo: Visual Studio.

Aplicar la metodología orientada a objetos nos permite conjugar el grupo de actividades que encierra el desarrollo de una aplicación web, ha establecido una investigación muy extensa, pero a decir de la estructuración misma lo desarrolla RUP (Proceso Unificado Racional), esta metodología en sus etapas correspondientes establecen visiblemente lo que ha significado la construcción de la presente aplicación; su diseño está enfocado a un ambiente gráfico, para la realización hemos utilizado la herramienta UML(Lenguaje de Modelamiento Unificado), lo que nos ha facilitado visualizar claramente y comprender los procesos, procedimientos y movimientos que se cumplen en las áreas de la institución de capacitación CENDOCAP.

5.02.02 Diagramas de Realización

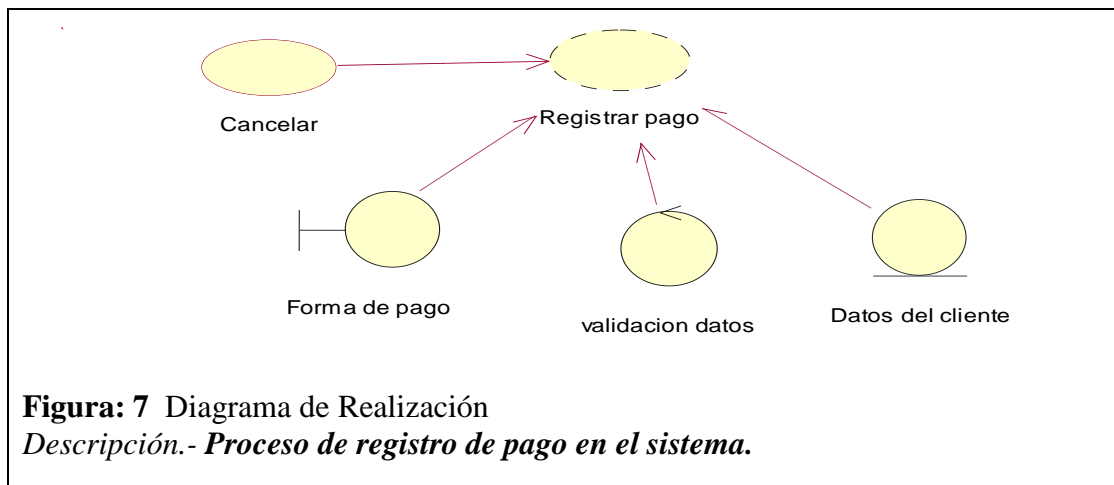


Tabla 5.02.02.01 Despacho de cliente

ID	UC001
Nombre	Registro de pago
Actores:	Secretaria/o, Cliente
Precondiciones:	<ol style="list-style-type: none"> 1. Verificación de cliente existente en la Base 2. Verificación de datos del cliente. 3. Constatación de pago. 4. Entrega de recibo al cliente.
Flujo de Eventos:	<ol style="list-style-type: none"> 1. Registro de cliente. 2. Registro de pago. 3. Registro de recibo entregado. 4. Informe de clientes despachados.
Flujo Alternativo.	<ol style="list-style-type: none"> 1. Registro de datos físicamente para luego registrarlos en el sistema.
Poscondiciones.	<ol style="list-style-type: none"> 1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

5.02.02. Diagramas de Realización

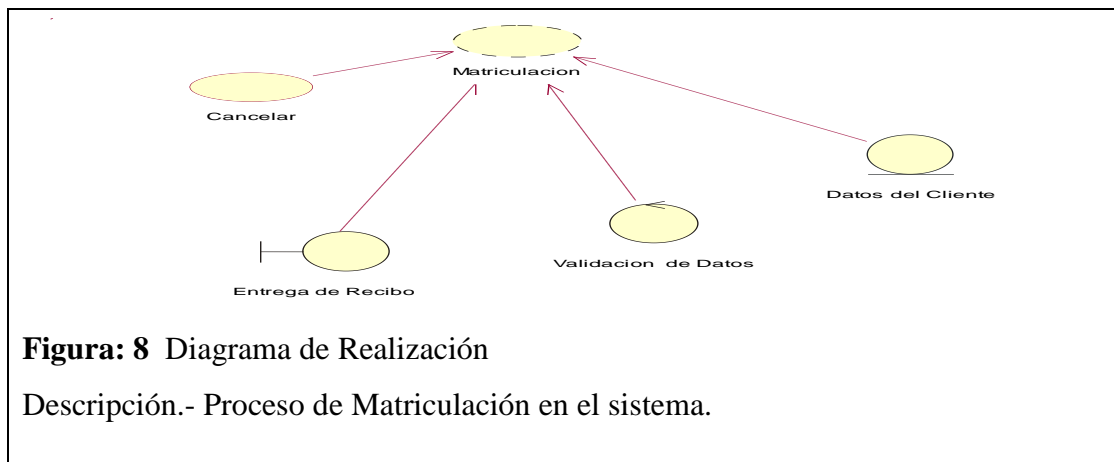


Figura: 8 Diagrama de Realización

Descripción.- Proceso de Matriculación en el sistema.

Tabla 5.02.02.02 Informe de Matriculación

ID	UC002
Nombre	Informe de la Matriculación
Actores: Secretario/a, Cliente	
Precondiciones: <ol style="list-style-type: none"> 1 Lleva un control adecuado de la información. 2 Mantener Actualizada la información. 3 Conocer la verificación de los datos en los proceso. 	
Flujo de Eventos: <ol style="list-style-type: none"> 1 Registra y guarda los datos del cliente. 2 Informar el estado de los registros de las inscripciones de la institución. 3 Muestreo de la integridad de datos ingresados. 4 Digitalizar toda la información del Centro. 	
Flujo Alternativo. <ol style="list-style-type: none"> 1. Entrega de informe al Director General de la Institución 	
Poscondiciones. <ol style="list-style-type: none"> 2. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso. 	

5.02.02.03. Diagramas de Realización

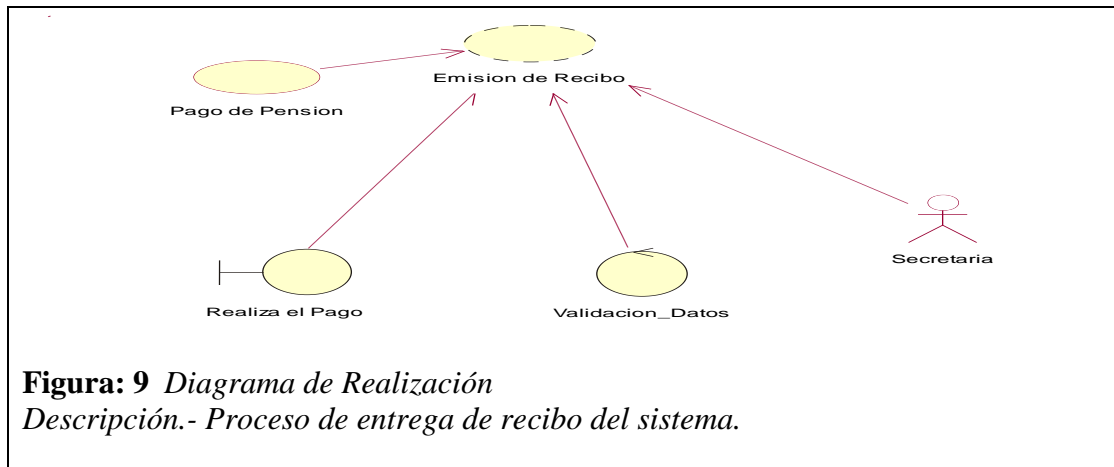


Tabla: 5.02.02.03 Entrega de Recibo

ID	UC004
Nombre	Pago de Servicio
Actores:	Secretaria/o, Cliente
Precondiciones:	<ol style="list-style-type: none"> 1. Pago de pensiones mensuales 2. Pago en efectivo 3. Entrega de recibo al cliente 4. Verificación de datos de la cancelación
Flujo de Eventos:	<ol style="list-style-type: none"> 5. Entrega recibos diarios. 6. Entrega de recibos realizados a la secretaria. 7. Registro de recibos
Flujo Alternativo.	<ol style="list-style-type: none"> 2. Información telefónica o por Email en caso de que no se realice el pago correspondiente.
Poscondiciones.	<ol style="list-style-type: none"> 3. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

5.02.02. 04 Diagramas de Realización

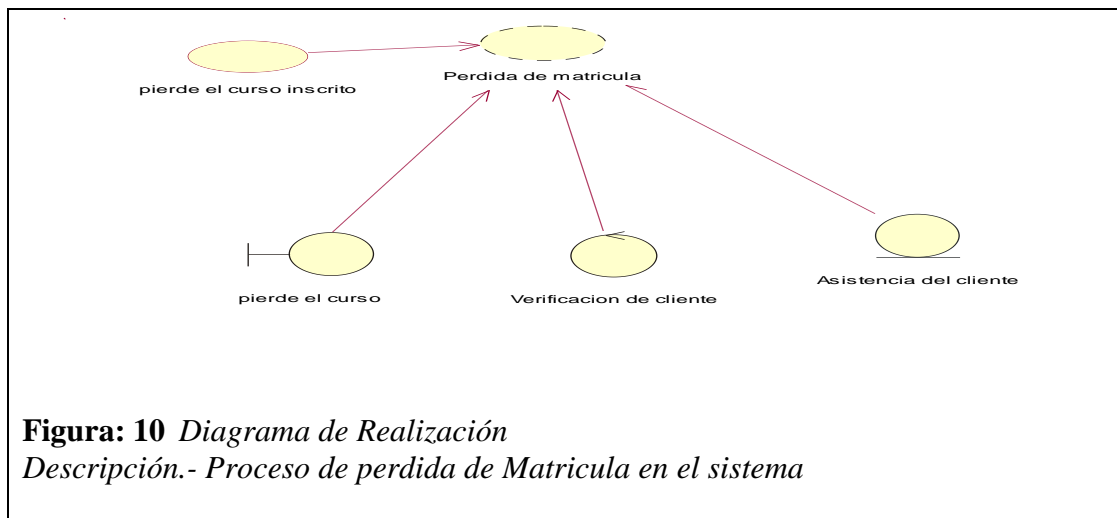


Tabla: 5.02.02. 04 Matriculas Pérdidas.

ID	UC005
Nombre	Matriculas Perdidas
Actores:	Institución, cliente
Precondiciones:	5. Profesor informa que el estudiante no asistió a clases. 6. Se informa que el estudiante ha perdido el curso
Flujo de Eventos:	8. Registro del estudiante que perdió el curso por no cumplir con la institución. 9. Registro de matrículas pérdidas o canceladas. 10. Informe de estudiantes que se dieron de baja.
Flujo Alternativo.	3. Estudiantes que no cumplan las reglas de la institución perderá el curso tomado.
Poscondiciones.	4. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

5.02.02. 05 Diagramas de Realización

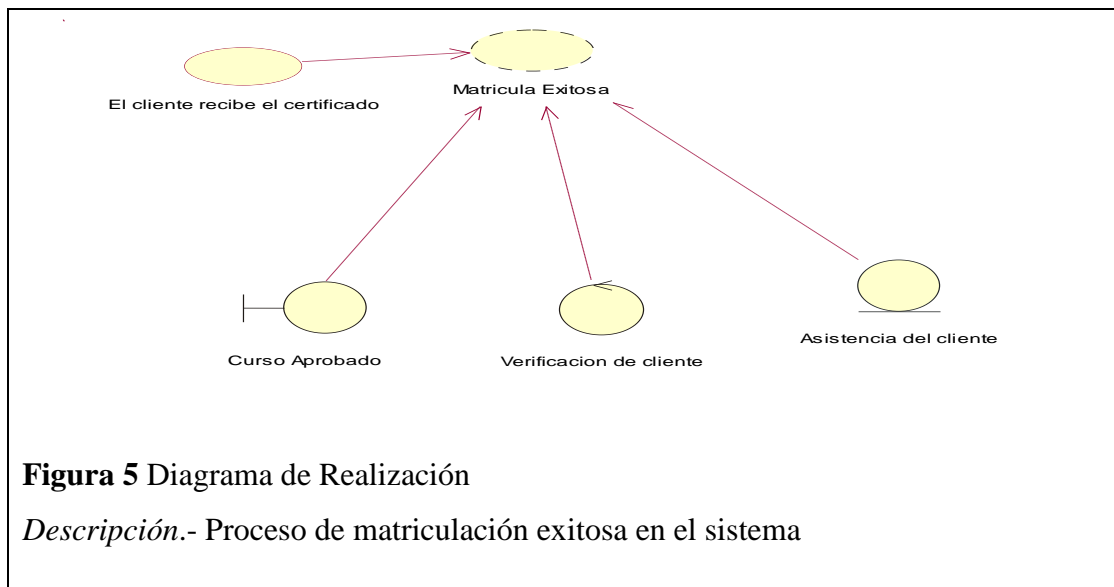


Tabla: 5.02.02. 05 Matricula Exitosa.

ID	UC006
Nombre	Matricula Exitosa
Actores:	Institución, cliente
Precondiciones:	7. Recibe el curso asignado en la matricula 8. Estudiante aprueba el curso con éxito. 9. Informa al Director que el curso ha terminado.
Flujo de Eventos:	11. Registro del curso ya terminado. 12. Registro de cliente. 13. Informe estadístico de estudiantes que terminan el curso cada mes.
Flujo Alternativo.	4. Estudiante que no tenga el 90% de asistencia no recibirá el certificado.
Poscondiciones.	5. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

5.02.02. 06 Diagramas de Realización

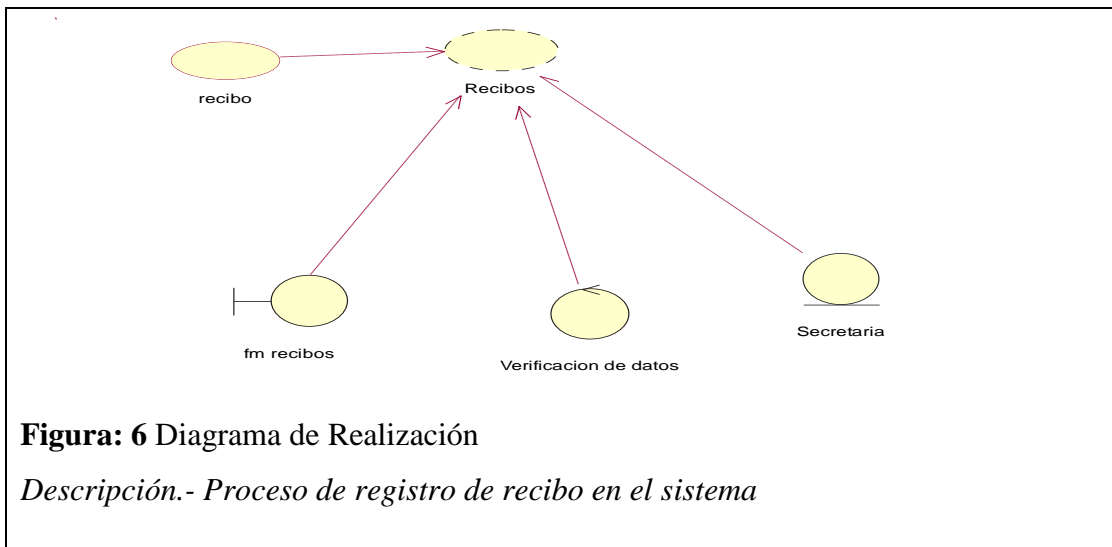
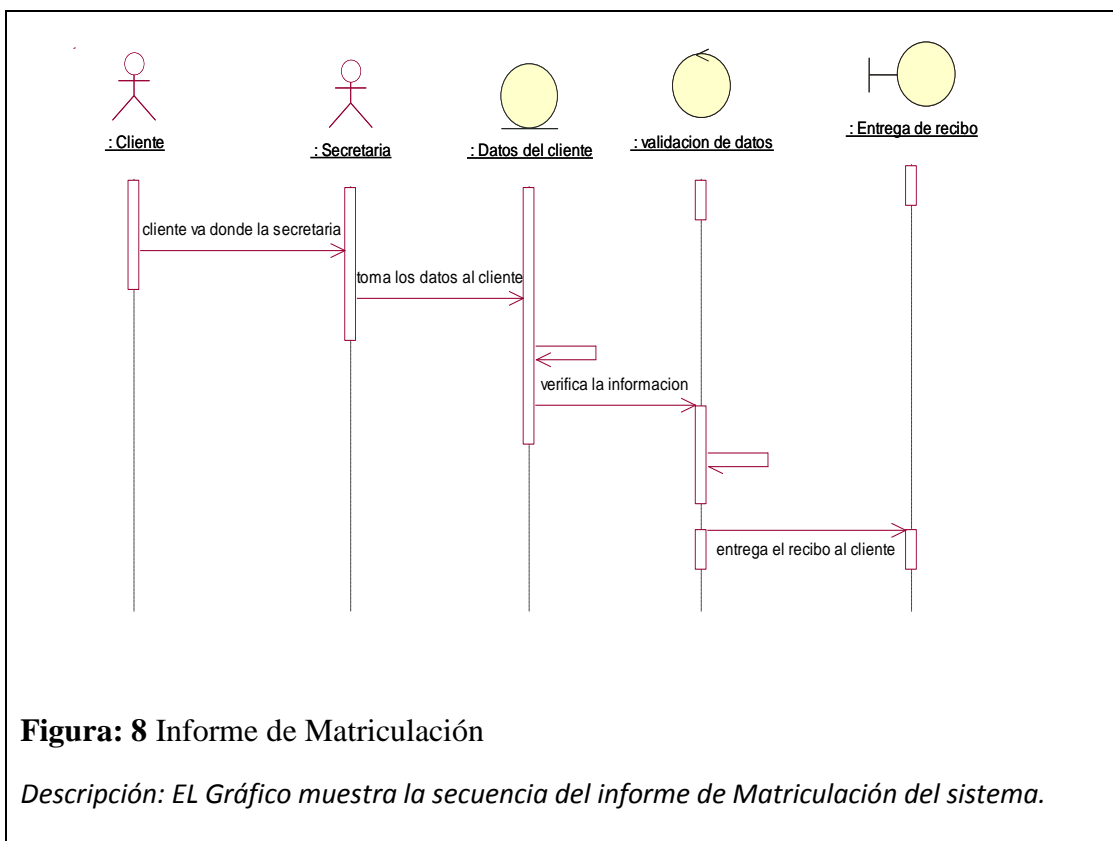
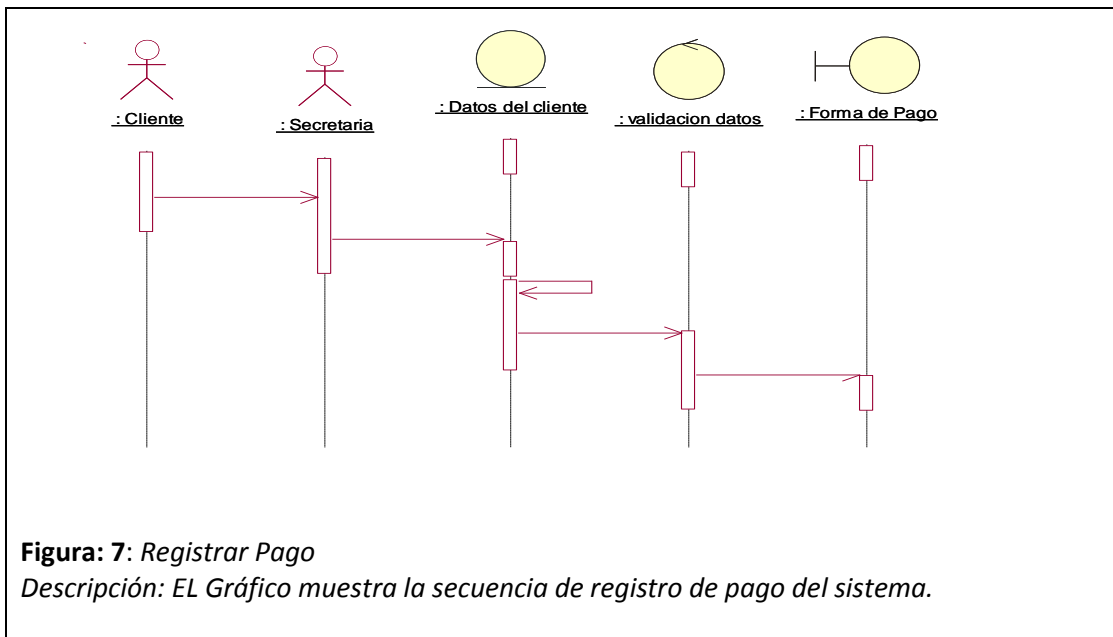


Tabla: 5.02.02. 06 Registro de Recibos

ID	UC007
Nombre	Registro de Recibos
Actores:	Chofer, Operador/a, Cliente
Precondiciones:	10. Control de recibos existentes. 11. Cuantificación de recibos utilizados. 12. Informe del total de recibos entregados.
Flujo de Eventos:	14. Registro de recibos en el sistema. 15. Verificación de datos en los recibos. 16. Búsqueda de recibos por número o fecha. 17. Cuadros estadísticos de recibos emitidos.
Flujo Alternativo.	5. En el caso de no registrar los recibos serán multados por el Director General de la institución.
Poscondiciones.	6. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

5.02.03 Diagramas de Secuencia

Un diagrama de secuencia demuestra la interacción de un grupo de objetos que intervienen en una aplicación a través del tiempo y se modela para cada caso de uso de realización.



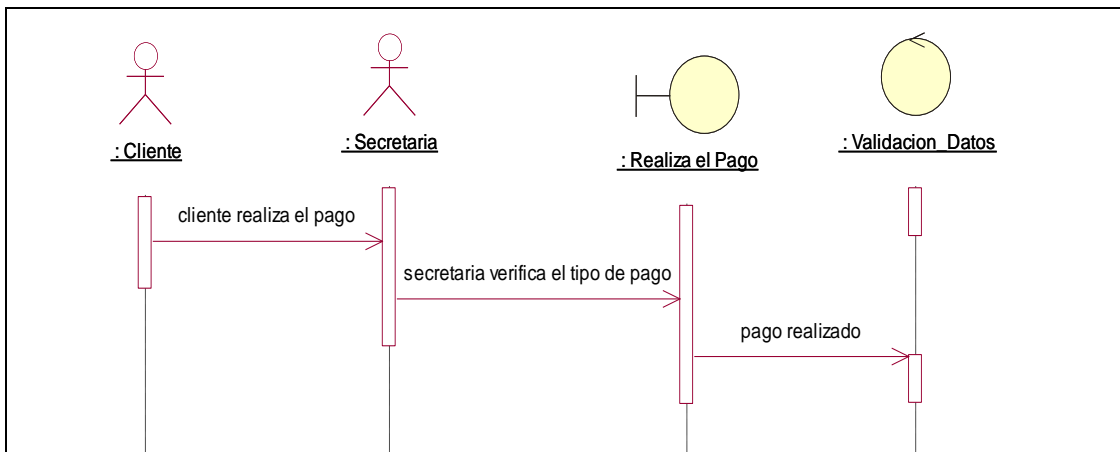


Figura: 9 Entrega de Recibo

Descripción: EL Gráfico muestra la secuencia de la entrega de recibo del sistema.

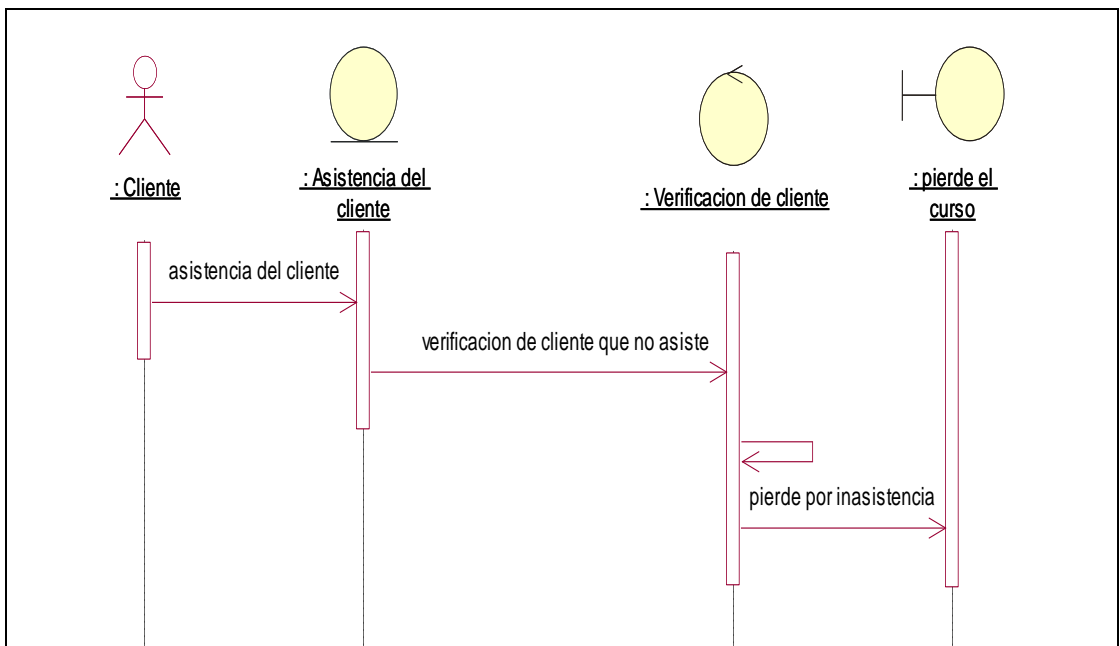
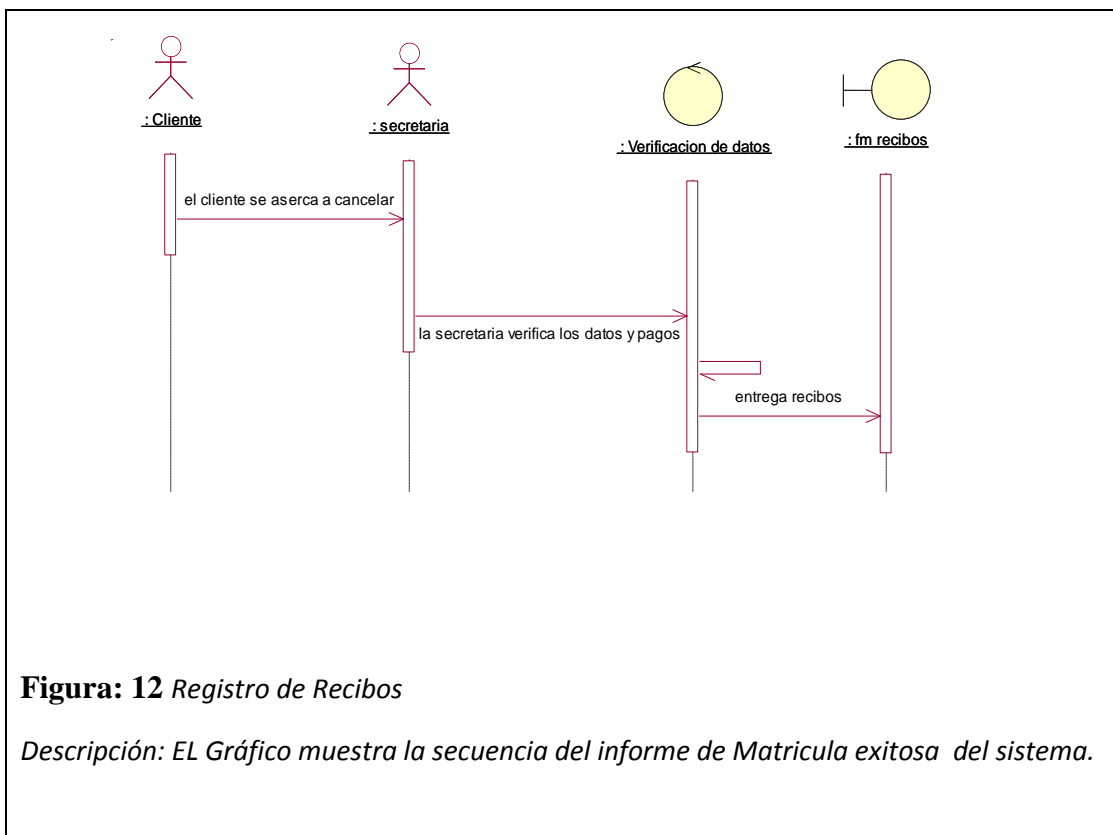
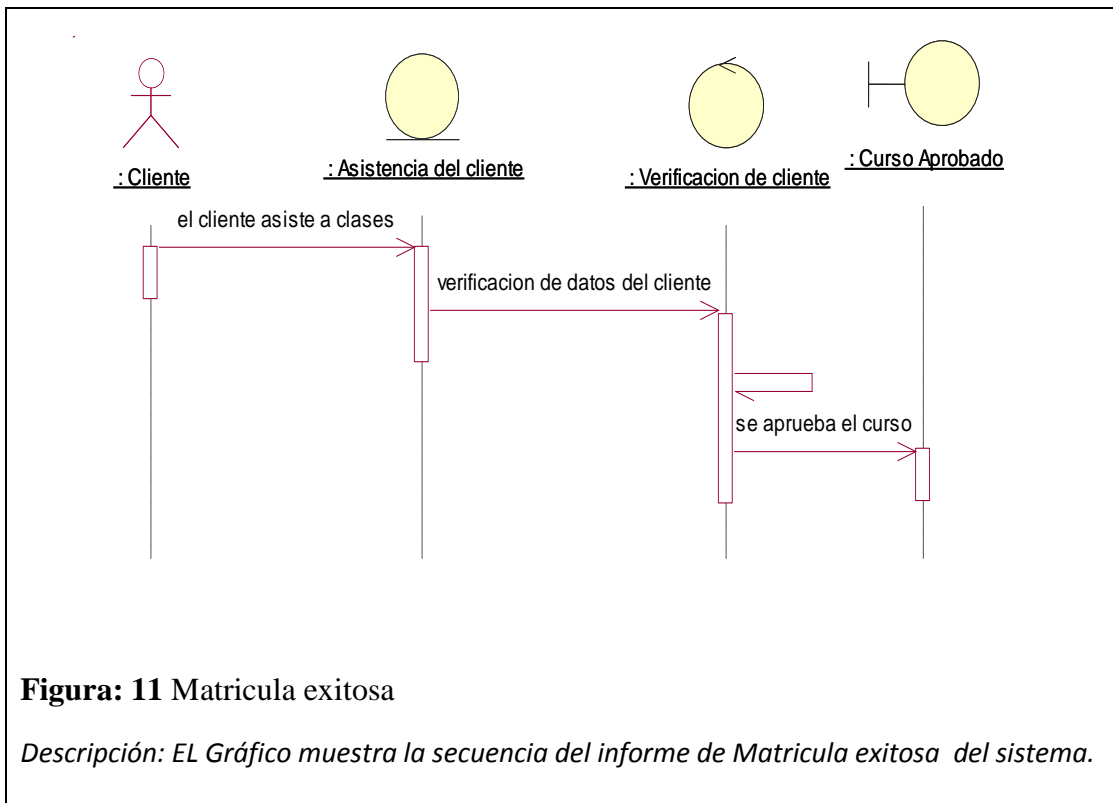


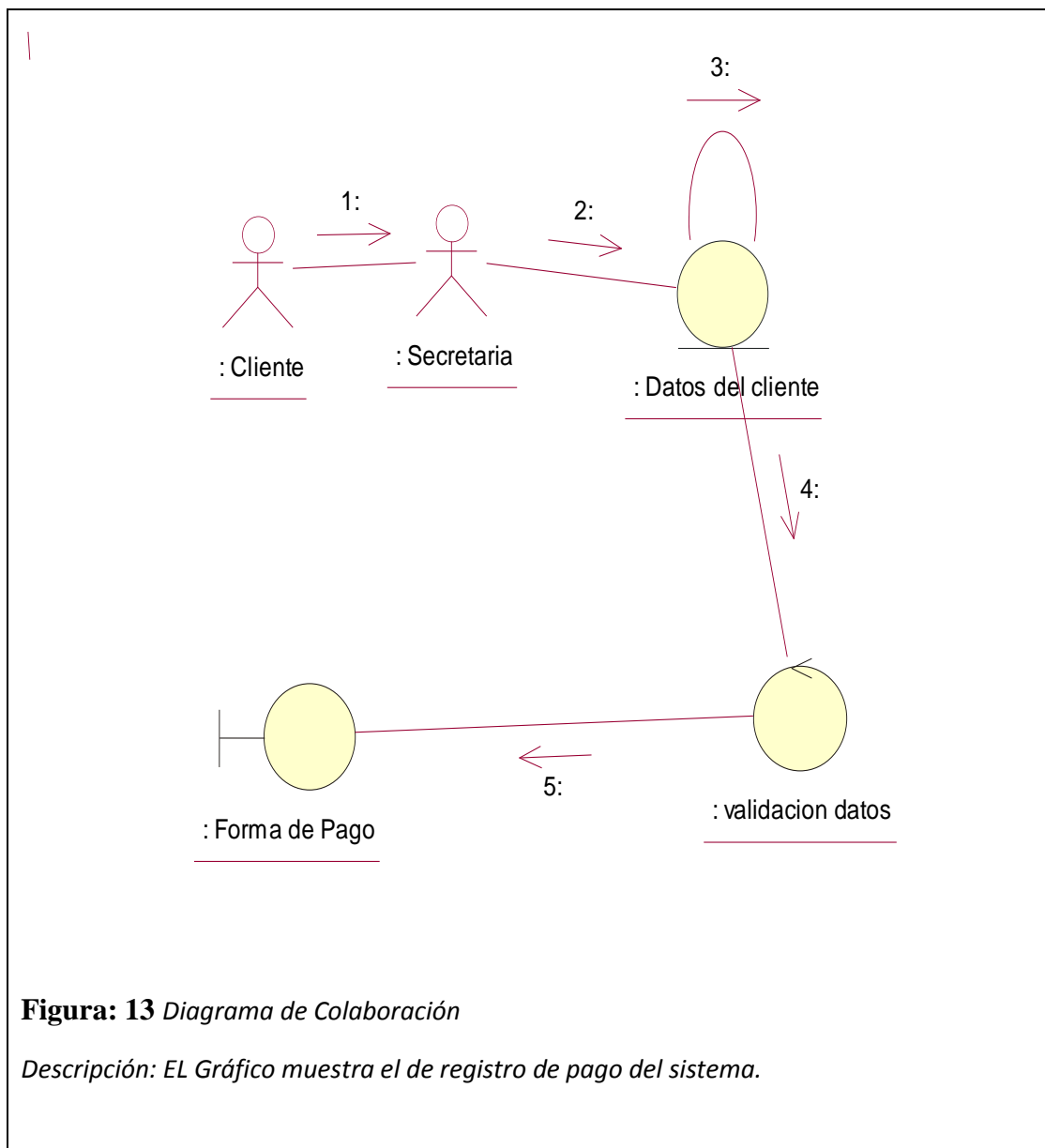
Figura: 10 Matriculas Perdidas

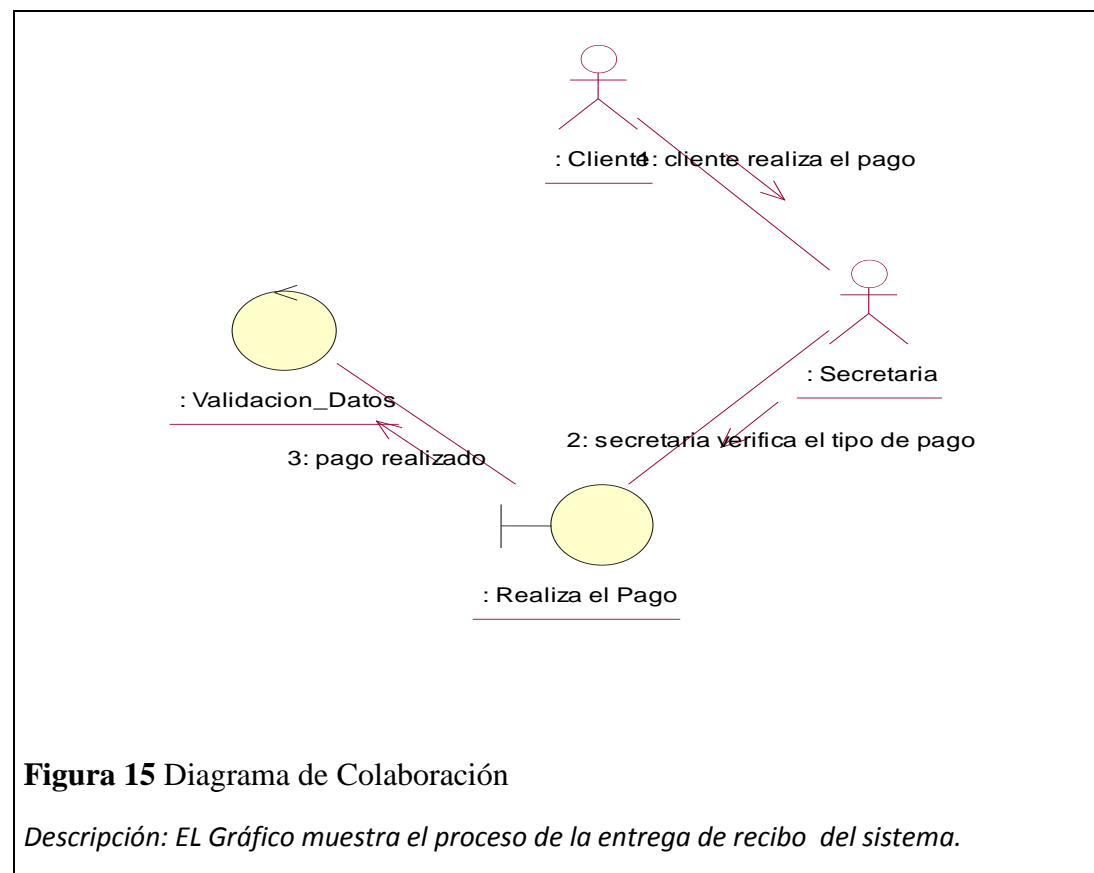
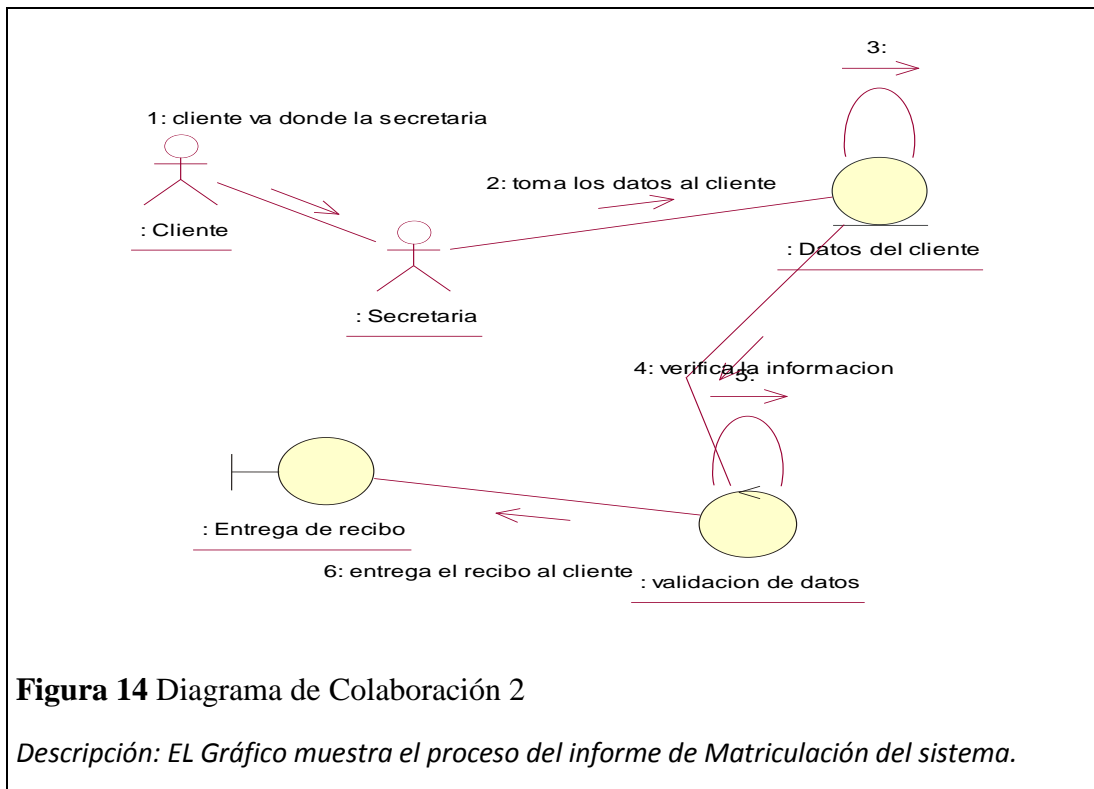
Descripción: EL Gráfico muestra la secuencia del informe de Matricula perdida del sistema.

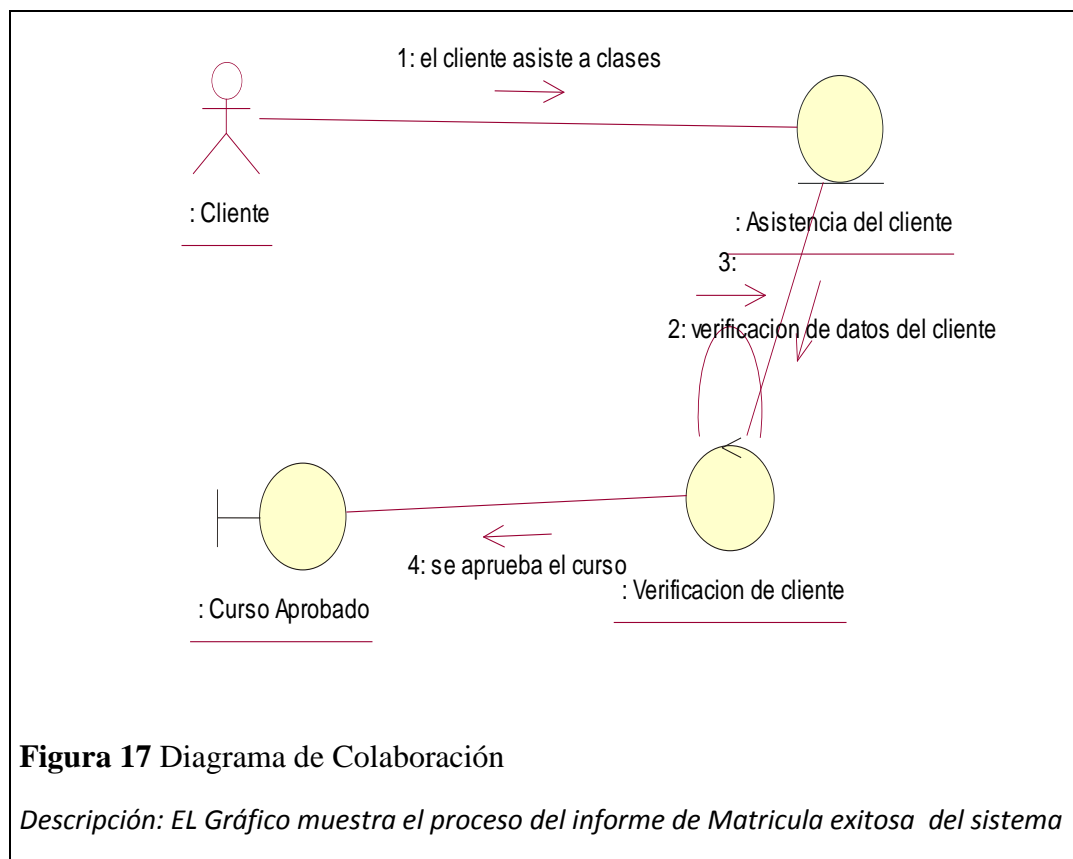
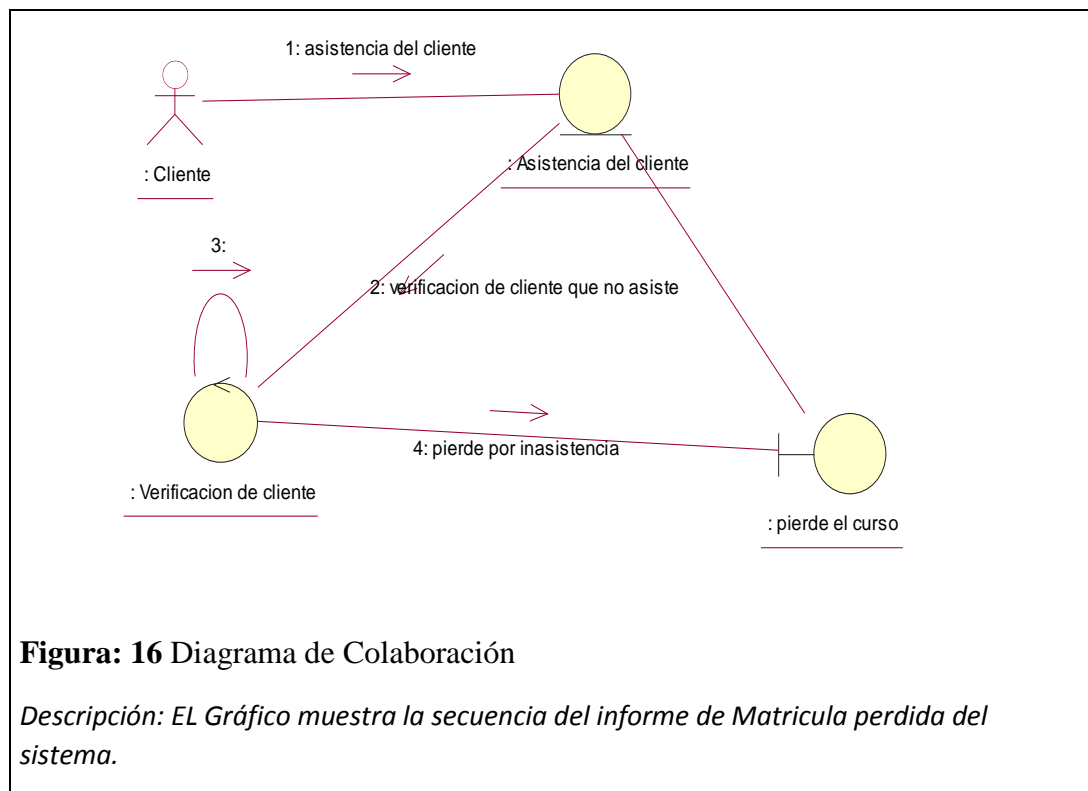


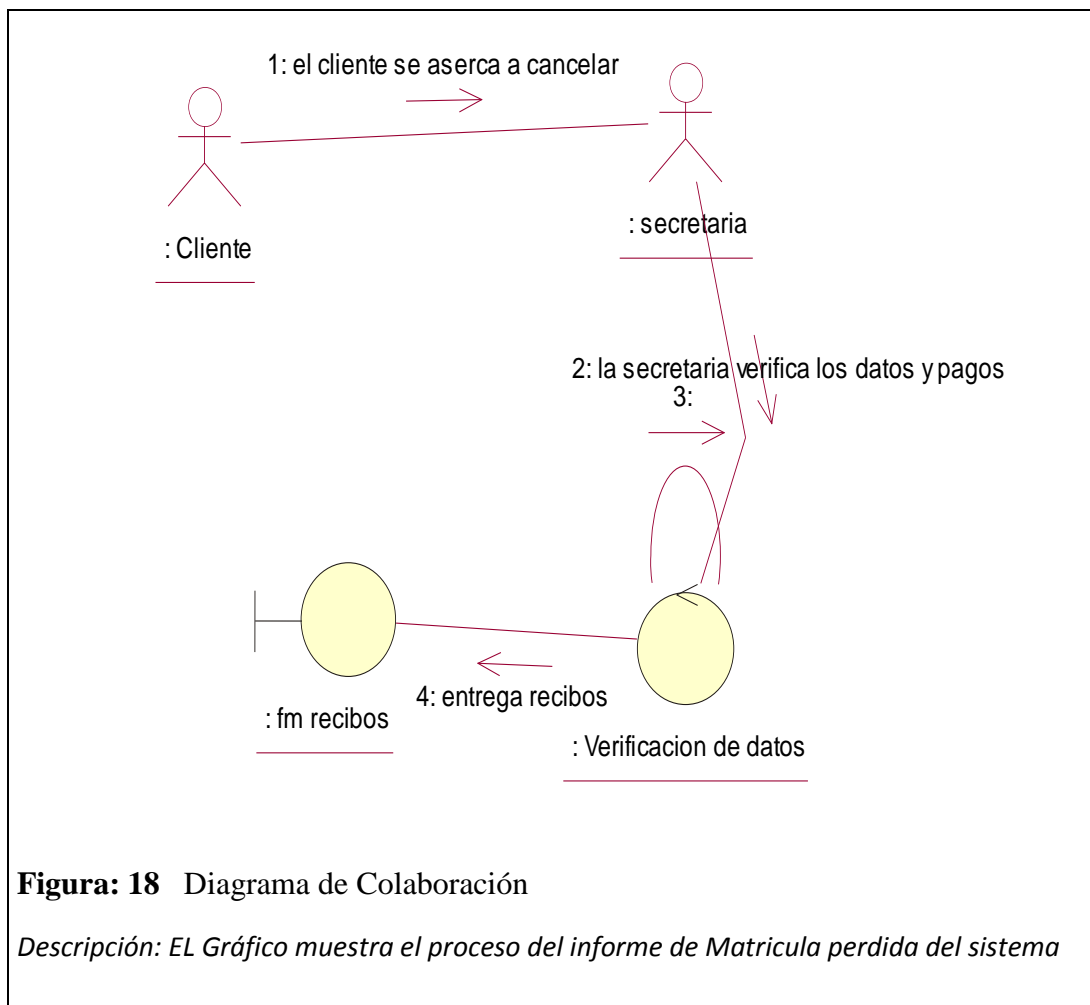
5.02.04 DIAGRAMA DE COLABORACIÓN

Los Diagramas de Colaboración son diagramas que nos permiten visualizar claramente todas las interacciones establecidas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, nos permiten visualizar claramente las relaciones de los roles. Se dice también que los diagramas de colaboración nos permiten identificar el tiempo del proceso de ejecución, siendo muy necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos presentes.



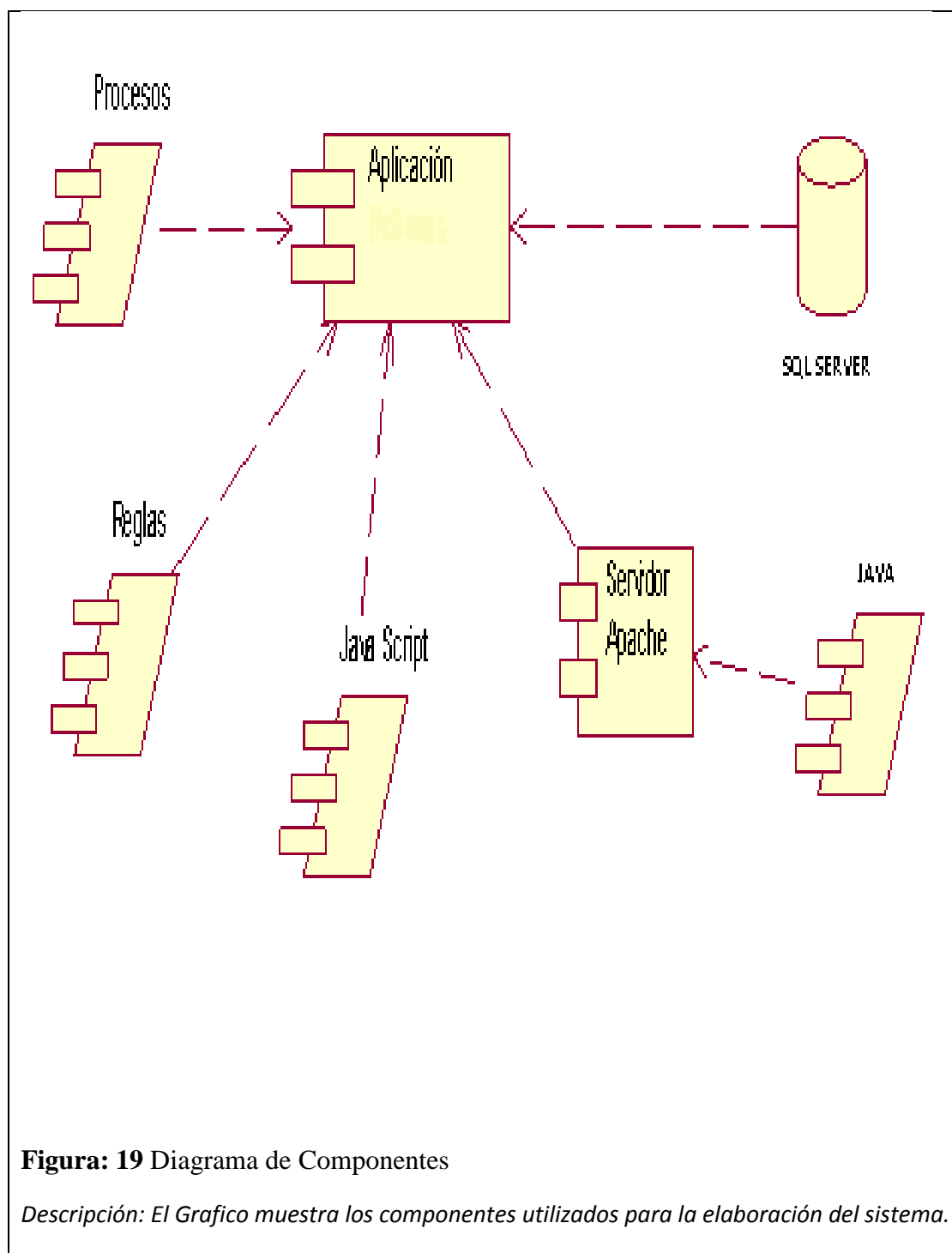






5.02.05 Diagrama de Componentes

El diagrama de componentes representa. Los componentes físicos, archivos, cabeceras, bibliotecas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser utilizados para modelar y documentar cualquier arquitectura del sistema.



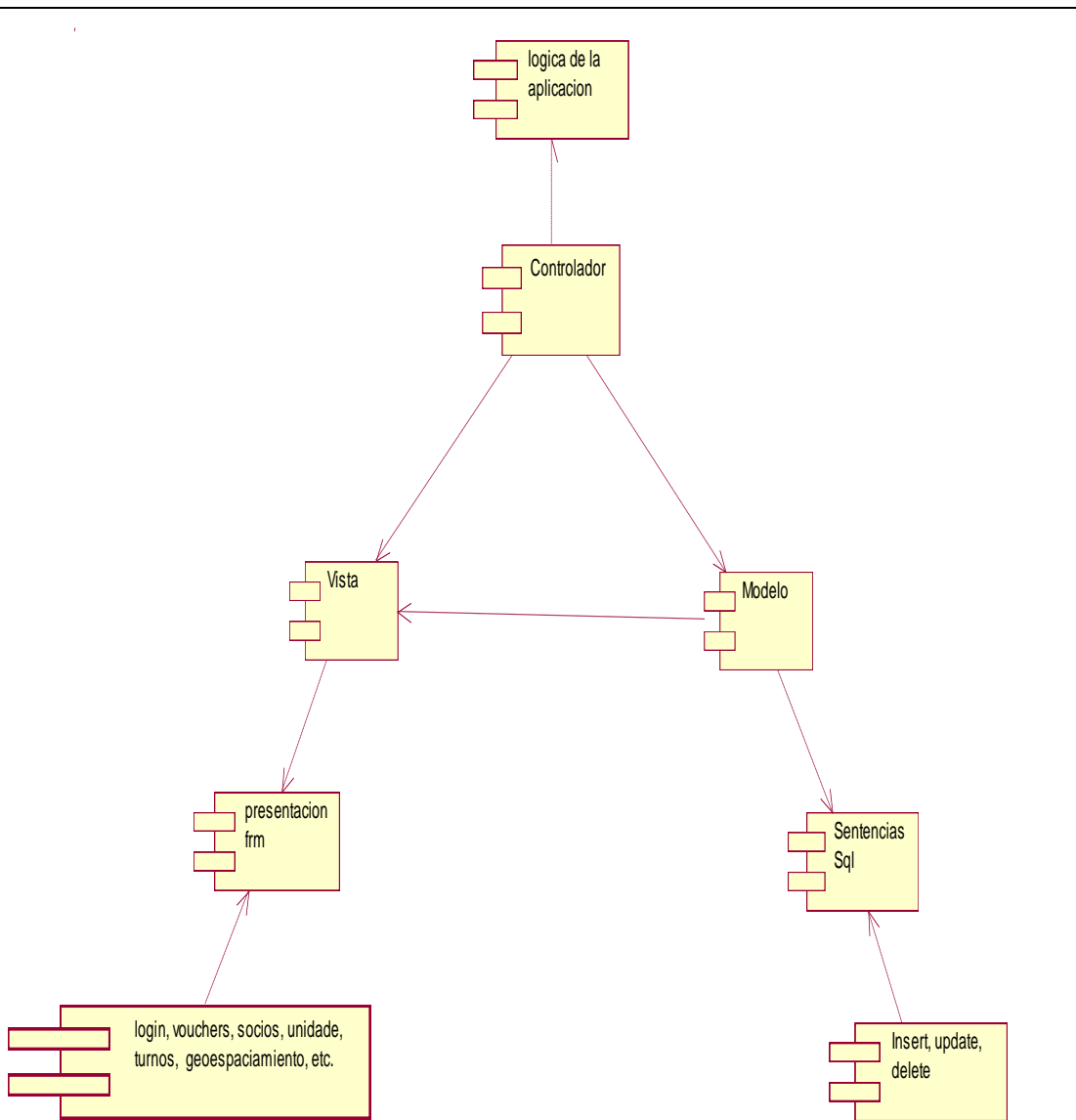
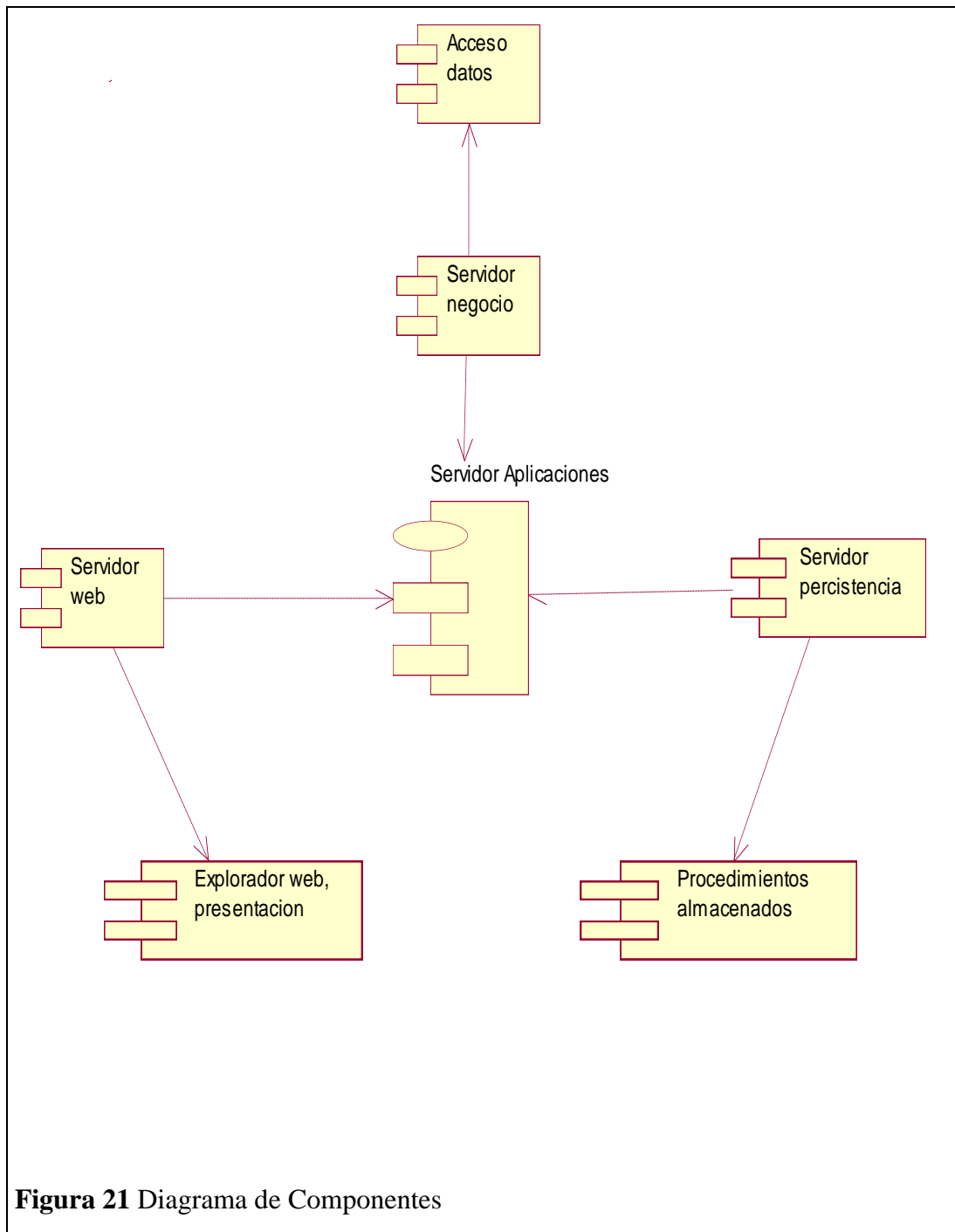


Figura 20 Diagrama de Componentes

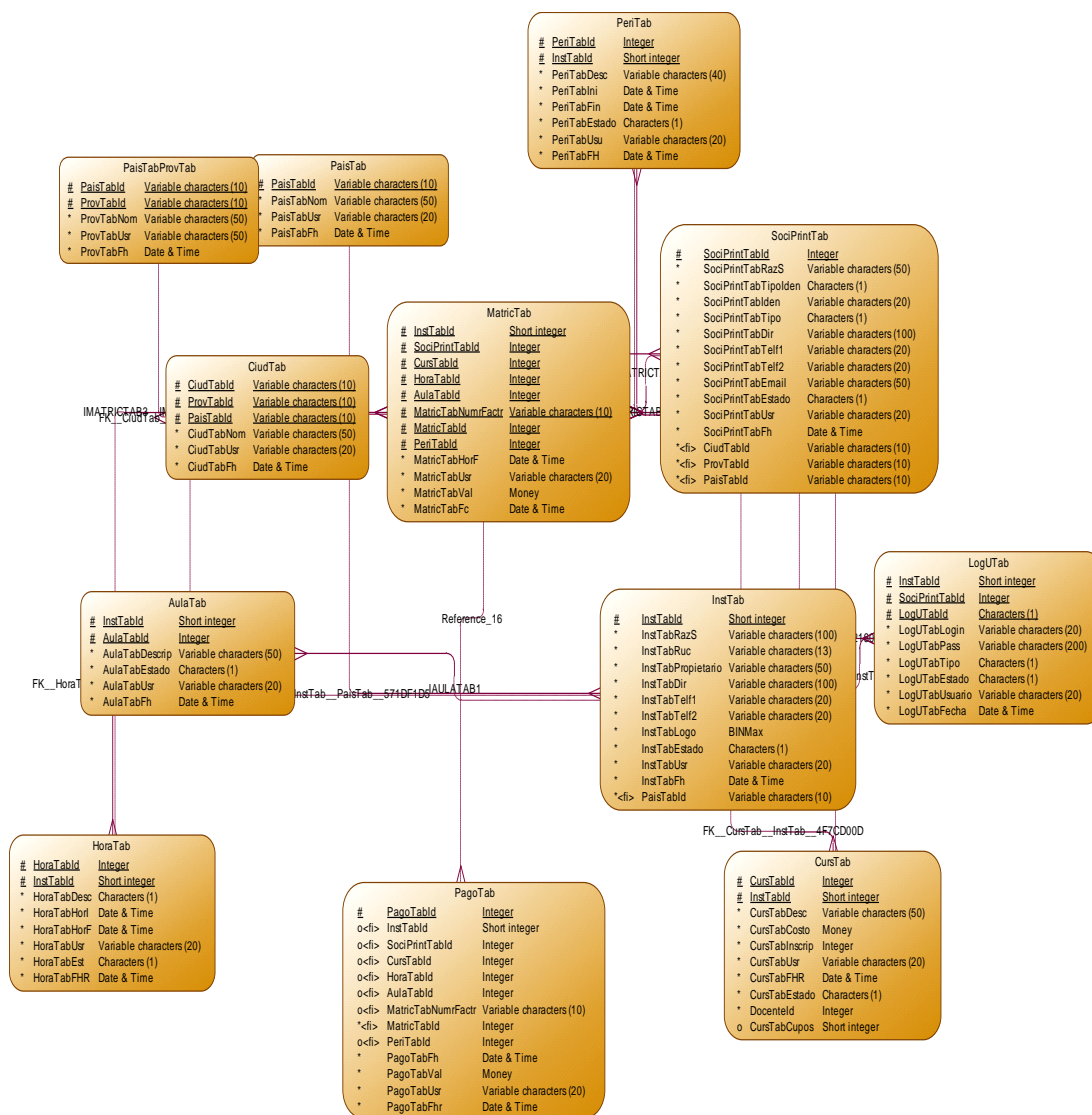
Descripción: El Grafico muestra el proceso que sigue el diagrama de Componentes



Conclusión del Diagrama de Componentes: Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo el código binario, fuente y ejecutable se utilizan para modelar la vista estática de un sistema y la organización y dependencias entre un conjunto de componentes.



Figura 23 Modelo Lógico



5.03 Desarrollo

5.03.01 Arquitectura Del Sistema

5.03.01.01 Capa de Presentación

La capa de presentación llamada también "capa de usuario" es la capa que se presenta el sistema al usuario como primer enfoque, le muestra toda información y la retiene la información del usuario en un proceso muy mínimo.

Conocida también como interfaz gráfica y siempre deberá ser caracterizada por ser muy fácil de comprenderla, fácil manejarla, llamativa y principalmente muy "amigable" para el usuario.

5.03.01.02 Capa de Negocio

Llamada Capa de Negocios o Lógica del Negocio porque es aquí donde se determinan todas las reglas y condiciones que deben cumplirse en la Aplicación en sí. Esta capa es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso.

La Capa de Negocios se comunica principalmente con la capa de presentación, para recibir las peticiones del Usuario y presentar los resultados, y también se conecta con la Capa de Datos, para pedir al Gestor de Base de Datos almacenar o recuperar datos de ingresados o guardados.

5.03.01.03 Capa de Datos

La Capa de Datos realiza todo el almacenamiento de datos de la institución, recibiendo peticiones de almacenamiento o recuperación de información desde la capa de negocio.

La capa de base de datos está compuesta de un modelo de clases el cual contiene el modelo conceptual y modelo físico del sistema de suministros con su Script de generación de base de datos, diccionario de datos, relaciones, claves primarias y foráneas y los índices respectivos.

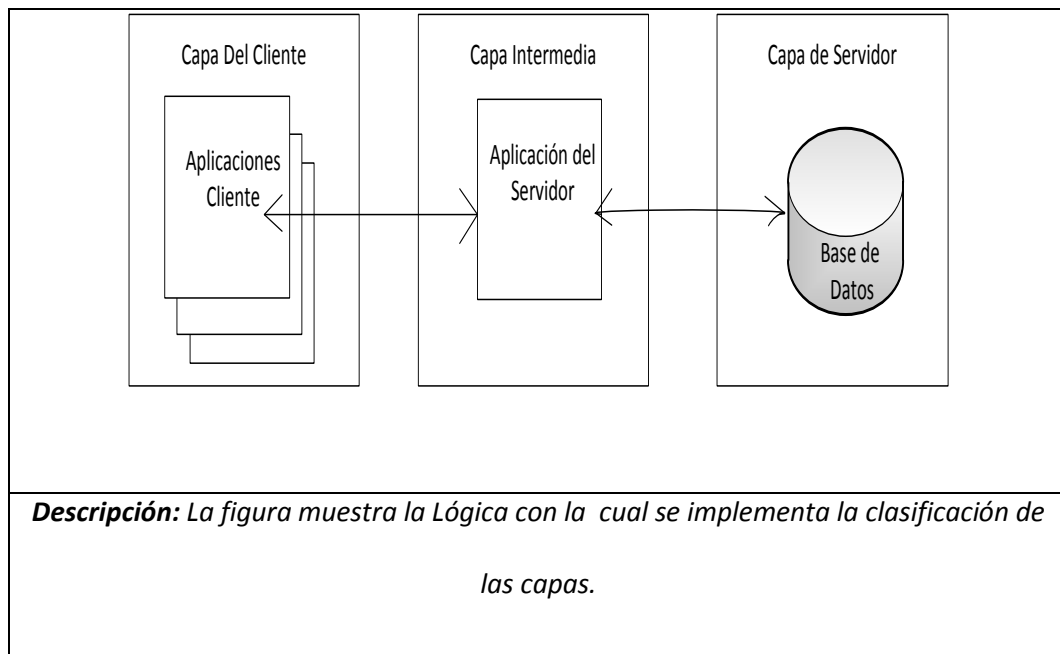
El diccionario de Datos nos facilitara con la descripción de cada uno de los campos en cada tabla de la base de datos.

Los Datos que se encuentren en la Base, serán manipulados con consultas, procedimientos almacenados y consultas anidadas para el manejo de información de las plataformas virtuales.

5.03.01.04 Capa de Servidor Web

Responde a las solicitudes de los usuarios con interacción con la capa del cliente, nos ayuda a además controlar los procesos que se manejan dentro de la lógica del sistema, esta capa es la encargada de realizar consultas SQL a la base de datos para así poder obtener la información y mostrarlas al usuario de la forma más precisa.

Figura 24 Arquitectura Lógica



5.03.01.06 Módulos

5.03.01.06.01 Módulo de Seguridad

El módulo de seguridad permite identificar el usuario que va a manipular el sistema, así como asignar los debidos roles a cada usuario, permitiendo así generar niveles de seguridad de acuerdo a cada perfil de usuario. Para resolver los problemas de autenticación y autorización, tanto para aplicaciones Web como para aplicaciones SQL Server, Visual Studio, GeneXus Acces Manager.

El módulo de seguridad permite identificar al usuario y asignarle los debidos roles a cada uno de ellos, permitiendo así generar niveles de seguridad de acuerdo a cada perfil de usuario.

5.03.01.06.02 Módulo Mantenimiento

El Módulo de Mantenimiento es el encargado de agregar, modificar, eliminar y buscar los datos necesarios en las respectivas tablas.

El Módulo de Mantenimiento es el encargado de agregar, modificar, eliminar y buscar los datos necesarios en los respectivos módulos.

Los mantenimientos son a:

Usuarios

Clientes

Maestros

Cobros

5.03.01.06.03 Módulo Lógica Negocios

Permite crear las validaciones de los procesos que realiza el sistema con la finalidad de que exista integridad en la información.

5.03.01.06.04 Módulo Reportes Estadísticos

Este módulo nos ayuda a generar la factura generada de acuerdo al pedido realizado por el cliente y validado por el Usuario.

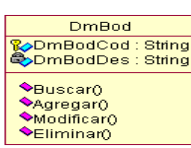

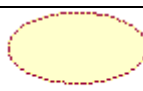

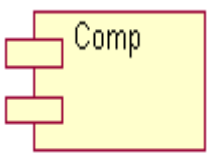
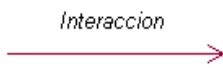
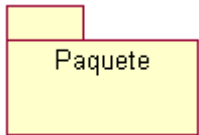
5.03.02 Estándares

5.03.02.01 Estándares de Diseño UML

Los estándares de diseño de UML aplicados para el desarrollo de este proyecto son:

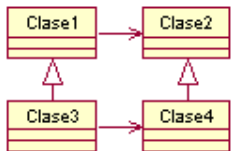
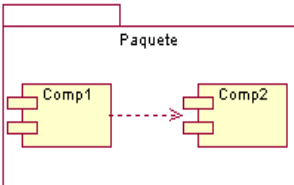
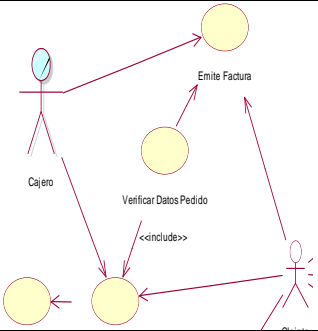
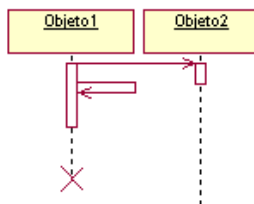
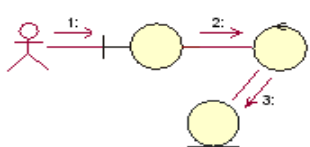
Elementos

Tabla: 5.03.02.02 Elementos de Diseño

E L E M E N T O S E S T R U C T U R A L E S	Clase	 <pre> classDiagram class DmBod { DmBodCod : String DmBodDes : String Buscar() Agregar() Modificar() Eliminar() } </pre>	Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos
	Interfaz	 <p>Interfaz</p>	El UML permite emplear un círculo para representar las interfaces.
	Colaboración	 <p>Colaboracion</p>	Define una interacción entre elementos que cooperan.
	Caso de uso	 <p>Caso de Uso</p>	Describe un conjunto de secuencias de acciones que un sistema ejecuta, para producir un resultado.
	Componente	 <p>Comp</p>	Parte física de un modelo, que agrupa un conjunto de interfaces.
Elementos de comportamiento	Interacción	 <p>Interaccion</p>	La flecha indica un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
Elementos de agrupación	Paquete	 <p>Paquete</p>	Se emplea para organizar otros elementos en grupos.

Diagramas

Tabla: 5.03.02.03 Diagramas de Diseño

M O D E L A N E S T R U C T U R A	Clases		Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubriendo la vista de diseño estática del sistema.
	Componentes		Muestra la organización de un conjunto de componentes. Cubren la vista de implementación estática de un sistema.
M O D E L A N C O M P O R T A M I E N T O	Casos de Uso		Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.
	Secuencia		El diagrama de secuencia resalta la ordenación temporal de los mensajes.
	Colaboración		El Diagrama de colaboración resalta la organización estructural de los objetos.

5.03.02.02.04 Corchetes e Identificación.

La identificación es algo que ayuda a darle claridad a un programa y es **INDISPENSABLE** que se haga bien. Debe hacerse con "tabs" y no con espacios en blanco.

Los corchetes de un bloque if, o switch, o for, deben ir en la misma línea de la cláusula. A continuación mostramos la forma apropiada de hacerlo.

Corchetes de identificación

Ejemplo

```
function verificarCondicion() {  
    if (condicion1) {  
        if (condicion2) {  
            while (condicion3) {  
                instruccion1;  
            };  
        };  
        instruccion2;  
    }else{  
        instruccion3;  
    };  
};
```

5.03.02.02.05 Clases

Los nombres de las clases deben de poseer un prefijo en conjunto con el nombre propio de la clase.

<Nombre de la capa><Nombre>

El nombre de la clase debe ser especificado por el consultor a cargo de su creación, debe de poseer un nombre significativo. El nombre debe de estar en singular.

5.03.02.02.06 Convención de nombres de variables

Estas deberían de nombrarse con un prefijo de tres letras el cual define el tipo de la misma. Los nombres deben ser descriptivos y concisos. Todos los nombres deben estar en minúsculas. En caso de usar más de una palabra, esta será separada por el signo “_”.

Convención de nombres de variables

Ejemplo

```
$txt_descripcion: Campo tipo text  
$chr_nombres: Campo tipo char o varchar  
$int_cantidad: Campo tipo integer  
$flt_total: Campo tipo float  
$dbl_precio : Campo tipo double  
$dtm_fecha: Campo tipo fecha  
$bol_activo: Campo tipo boolean
```

5.03.02.02.07 Convención de nombres de funciones

En el caso de funciones, es importante que el nombre denote su función inmediatamente.

Convención de nombres de funciones

Ejemplo

```
imprimir_datos() esta bien pero estaría mejor:  
imprimir_datos_usuario()
```

5.03.02.02.08 Precedencia entre operadores

Lo mejor es usar siempre paréntesis entre estos operadores.

Precedencia entre operadores

Ejemplo

```
// ¿Qué da esto como resultado?  
$bool = ($i < 7 && $j > 8 || $k == 4);  
  
//En cambio si ponemos así es obvio y sencillo  
$bool = (($i < 7) && (($j > 8) || ($k == 4)));  
  
//Pero este es incluso mejor, porque está más  
optimizado y su lectura es superior  
$bool = ($i < 7 && ($j > 8 || $k == 4));
```

5.03.02.02.09 Funciones de JQuery

Es un Framework de Javascript, o ambiente de desarrollo, utilidades programadas, probadas y podemos utilizarlas de una manera muy simplificada.

```
$( document ).ready(function() {});
```

```
$( "a" ).click(function() {});
```

```
$.ajax({  
    url: "someurl.asp",  
    success: function() {}  
});
```

5.03.02.02.010 Ajax en JQuery

```
$.ajax({  
    url: "test.html",  
    context: document.body
```

```
}).done(function() {  
  
    $(this ).add Class( "done" );  
  
});
```

5.03.02.03.01 Estándares de Base de Datos

Los objetos de base de datos deberán estar debidamente documentados con las siguientes propiedades.

Descripción: debe contener la descripción del uso o la funcionalidad el objeto y que no sea una simple repetición del nombre.

Base de Datos

La base de datos y cada uno de sus objetos poseen también una nomenclatura y una Forma de documentación. La descripción de los tipos de datos a utilizar dentro del diseño de la base de datos debe ser clara y precisa.

Atributos

Para el caso específico de atributos que son llaves foráneas y corresponden al código de la tabla donde está la llave la primaria, al nombre del campo se le debe ir después del nombre de la tabla seguido del atributo que corresponda.

Tabla 5.03.02.03.02 Atributos

NOMBRE	NOMENCLATURA
SociPrintTabId	SociPrintTabId
CiudTabId	CiudTabId
PaisTabId	PaisTabId

Nota: La tabla muestra la nomenclatura de los códigos PK que se asignaron a las tablas.

Tabla: 5.03.02.03.03 Tipo de Datos

Tipo de Datos	Característica
Boolean	Un número entero que puede ser 0 ó 1
SmallInt	Número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535.
Integer, Int	Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295
Float	Número pequeño en coma flotante de precisión simple. Los valores válidos van desde -3.402823466E+38 a -1.175494351E-38, 0 y desde 1.175494351E-38 a 3.402823466E+38.
Date	Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día
DateTime	Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas: minutos: segundos
Char	Almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 255 caracteres.
VarChar	Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.

Nota: La tabla muestra el Tipo de dato usado en las diferentes Tablas del Sistema

Tablas

Para los nombres de las tablas de rompimiento se considerará el nombre de las tablas involucradas en el rompimiento tomando las tres primeras letras de cada tabla o si llegara a coincidir las tres primeras letras se toman cuatro de cada tabla de rompimiento.

El nombre de la tabla se escribirá en minúsculas ya que el gestor de base de datos SQL server al momento de recuperar el script los escribe así, y se escribirá en singular con algunas excepciones que pueden presentarse para una mejor descripción de la tabla.

Nombres de Tablas

Nombre de tablas

Nombre	Nomenclatura
Tabla de usuario	tbl_usuario
Tabla cliente	tbl_cliente

5.03.02.03.04 Campos

El nombre de un campo debe ser de las tres o cuatro primeras letras dependiendo del nombre de la tabla, el nombre del campo del atributo es auto descriptivo. Únicamente las claves primarias y las claves foráneas deben ir antepuesta la palabra id.

Sistema de Control de Facturación

Tabla: 5.03.02.03.05 Ingreso al sistema

Ítem	Representación	Descripción
A	Caja de Texto	Usuario
B	Caja de Texto	Clave

Figura: 29 Ingreso al sistema

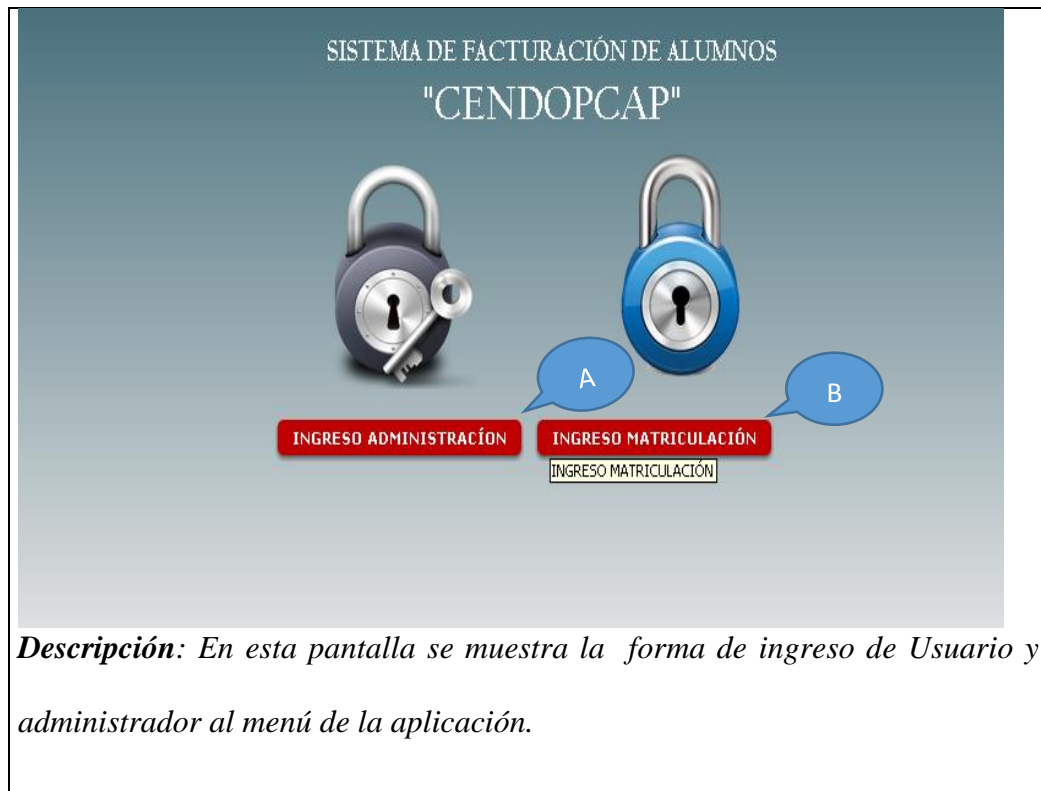


Tabla: 5.03.02.03.05 Ingreso de un Nuevo Socio

Ítem	Representación	Descripción
A	Caja de Texto	Código Empleado
B	Caja de Texto	Código Curso
C	Caja de Texto	Hora
D	Caja de Texto	Fecha
E	Caja de Texto	Valor Matricula
F	Botones	Confirmar, Cancelar

Figura: 30 Ingreso de un Nuevo Alumno



Creación de Alumnos

Matriculación de alumnos

Código de Matricula 2

Código de Empleado

Código de curso

Código de Hora

Código de aula

Código de Periodo

Factura N.

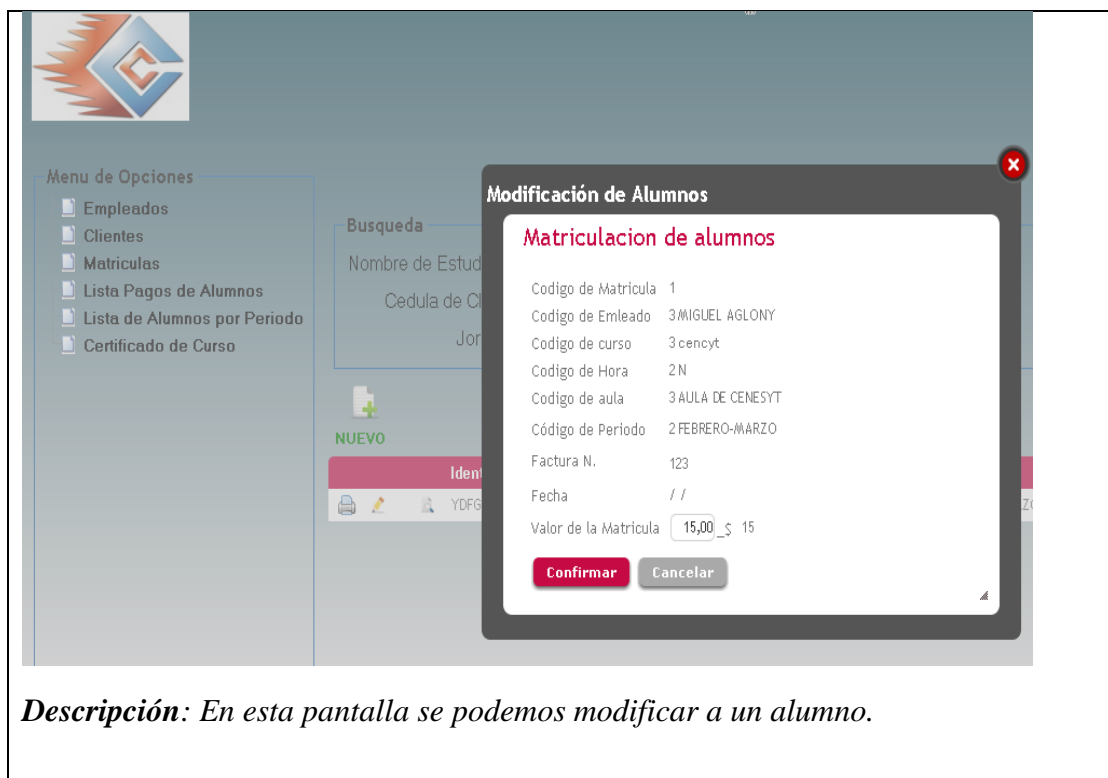
Fecha

Valor de la Matricula 0.00 \$ 0

Confirmar **Cancelar**

Descripción: En esta pantalla se muestra la forma de ingreso de nuevo alumno en la Matriculación la aplicación.

Figura: 31 Modificación de Alumnos



Descripción: En esta pantalla se podemos modificar a un alumno.

5.04 Pruebas

5.04.01. Objetivo

Identificar que la información existente en el proyecto de control de los procesos de facturación y los componentes cumplan con los estándares de seguridad y calidad. Verificar que los objetos y las clases tengan relación directa con los componentes de software y que sean probados y verificados.

5.04.02. Pruebas de módulo.

5.04.02.01. Módulo de Seguridad

Este tipo de prueba se realizara mediante la validación de cada una de las cajas de texto que manejen la obtención o captura de datos, estas pruebas serán correlacionadas con los estándares de manejo de seguridad en el campo informático

ISO 27000 lo que implica que el usuario deba mantener 12 caracteres como mínimo combinados entre mayúsculas y minúsculas números y caracteres especiales. En lo referente a las cajas de clave deben tener 8 caracteres combinaciones mayúsculas minúsculas y caracteres especiales.

El cliente podrá realizar su debido ingreso y manipulación de los datos que el administrador le hubiese asignado tomando en cuenta que solo podrá realizar cambios en su cuenta.

El ingreso erróneo en cualquiera de estas dos cajas de captura de datos (usuario y clave) abortará el acceso al sistema y enviará a la página de acceso.

El usuario que hubiese ingresado y que estuviese inactivo por más de 5 minutos automáticamente se cerrará su cuenta y tendrá que volver a ingresar al sistema.

5.04.02.02. Módulo Mantenimiento

El módulo de mantenimiento tiene que ser sometido a pruebas en las cuales se verifique el acceso idóneo de información hacia la base de datos; por esta razón inicialmente se determinará que los campos que manejen código de acceso en una tabla que tenga una función que me permita generar códigos secuenciales sin que estos se repitan y puedan generar errores de duplicidad de claves primarias., las cajas que requieran la captura de información deberán estar validadas de acuerdo al dato de ingreso ya sea esté números enteros o decimales, fecha y hora deberá ser los estandarizados por las normas ISO año 4 mes 2 día 2 separados mediante un -. En las cajas de texto que requiera identificar la cédula deberá estar validado que el número ingresado sea válido para poder continuar con las operaciones de ingreso.

5.04.03. Pruebas de Interfaz de Usuario.

Esta prueba se realiza inicialmente verificando facilidad con la que el usuario se desenvuelve en realizar las distintas operaciones en el sistema de control geoespacial, así como la carga tiene que ser transparente para el usuario. Con la capacidad de medir el tiempo de respuesta en la navegación del ambiente browser (se procederá a la manipulación del elemento que más implica carga de información hacia la interfaz de usuario para poder conocer el tiempo de respuesta del sistema de control geoespacial), se realizó el mantenimiento de la aplicación en el ambiente web y se obtuvo que el tiempo de respuesta es óptimo en el desenvolvimiento de entrega de resultados específicamente en la aprobación de un pedido ya esa interfaz es la que maneja mayor obtención de información de la base de datos.

En este formulario verificamos que el salto de las cajas de texto se las realice con el enter, igualmente se cumple con la verificación de los estándares GUI que implica color de fondo de las cajas de texto, color de texto de la recuperación de datos simetría en la distribución de cajas recuperadoras de información, diseño de ubicación de objetos de interfaz de usuario (text box, check box etc).

De igual manera se verifica el estándar de comunicación que existe entre los datos estableciendo tiempos de respuesta en la recuperación de información desde la base de datos hacia la aplicación y desde el aplicativo hacia la base de datos.

Por otro lado se determinó una prueba de ejecución del manejo de los iconos inicialmente que estén acorde con la información solicitada; en este punto se verifico ventanas y mensajes de alerta, ventanas y mensajes de información de la

acción que se ha ejecutado, ventana y mensajes de captura de errores con el manejo de excepciones.

5.04.04 Pruebas de Desempeño.

Con la finalidad de poder ejecutar este tipo de pruebas y que las mismas tienen incidencia con la arquitectura montada, fue necesario estructurar el sistema de facturación, en un ambiente distribuido que claramente se pueda evidenciar el servidor de datos, servidor de aplicaciones, y los clientes. La primera prueba realizada fue la verificación del tiempo de respuesta del cliente hacia el servidor de aplicaciones y posteriormente del servidor de aplicaciones al servidor de datos, el tiempo de respuesta final se lo determina con la suma total de tiempo realizado desde la petición de información hasta la recepción de la misma y esto dividido por dos. Se considera que el servidor debe estar separado para poder realizar las pruebas, de esta manera podremos verificar el tiempo de respuesta de acceso remoto a la aplicación.

Otra de las pruebas fue la verificación del tiempo de respuesta en las transacciones en la que tiempo se demora.

Continuando con las pruebas de desempeño tenemos que verificar tiempos válidos de respuesta para recibir y enviar información mediante el XML igualmente el tiempo de respuesta que se demora en generar este tipo de información.

5.04.05. Pruebas de Carga.

Para realizar esta prueba es necesario ejecutarla sobre una sola tabla la misma que es la que tiene más transaccionalidad dentro del en el proyecto la tabla "Tbl_Clientes" con un soporte mínimo de 50 clientes. Se medirá la persistencia de las tablas y de la programación para así obtener la persistencia de la transaccionalidad del sistema de control de Facturación.

5.04.06 Pruebas de Validación.

El objetivo principal de las prueba de validación es obtener información útil para la validación de la implementación. Una vez que el sistema de control de facturación ha cumplido con la verificación de errores por lo tanto está libre de errores en tiempos de ejecución, lo que significa que está libre de errores lógicos.

El módulo de seguridad es donde se realiza la prueba de validación pues controla el acceso al sistema validando y verificando que los datos ingresados sean los correctos.

Se verifica que existan mensajes acorde a las acciones erróneas que se realizan en el ingreso de datos del sistema. La validación del software se la realiza con pruebas de verificación que demuestren que es 100% funcional

CAPITULO VI

6.1 Recursos

Tabla 5.04.07: Recursos

RECURSOS HUMANOS			
Humano	Nombre	Actividad	Responsabilidad
Promotor del proyecto	Fernando Nogales	Desarrollo del proyecto.	Llevar a cabo la investigación y desarrollo del proyecto Ventas y cobros .
Tutor	Prof. Luis Ríos	Guiar el desarrollo del proyecto.	Responsable de la guiar e instruir al Promotor.
Lector	Ing. Jaime Padilla	Revisar el desarrollo del proyecto.	Constatar que el desarrollo del proyecto elaborado cumple con las normas establecidas.

Nota: El cuadro muestra un detalle sobre los recursos humanos utilizados en el Proyecto

Tabla 5.04.08: Recurso Material

RECURSOS MATERIALES			
Material	Cantidad	Valor Unitario	Valor Total
Computador	2	700,00	900,00
Hojas de papel bon	1000	0,03	27,00
Servicios Básicos	1000 horas	0,25	175,00
Servicio de internet	1000horas	0,02	16,00
Alimentación	90 días	1,50	135,00
Transporte	90 días	0,50	45,00
Total			2.099,20

Nota: El cuadro muestra un detalle de los recursos materiales utilizado en el desarrollo del proyecto.

Tabla 5.04.08: Recurso Material

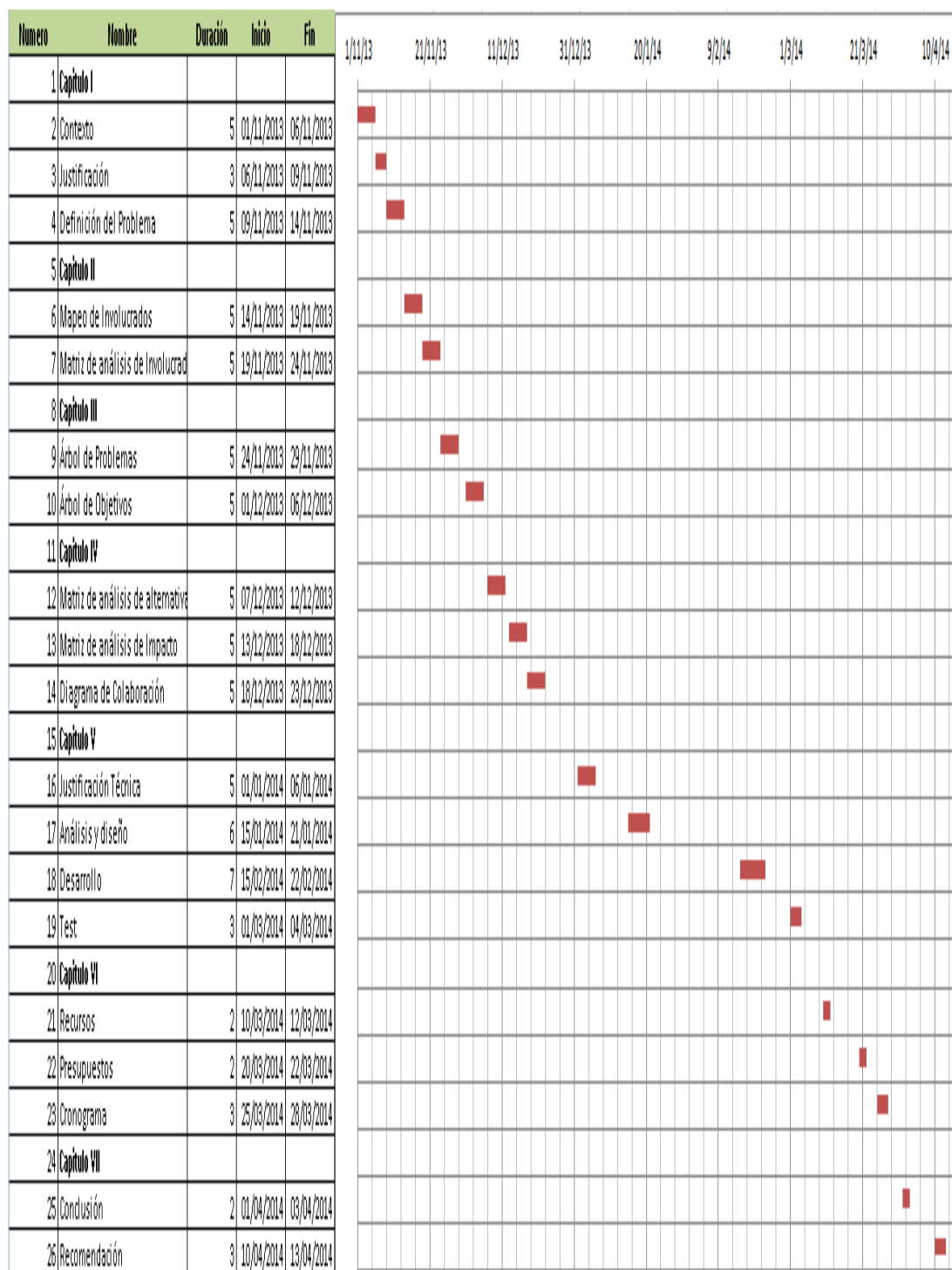
Tabla 6.02.01 Presupuesto para la realización del proyecto.

Nombre	Cantidad	Valor Unitario	Valor Total
Empastado	1	10,00	10,00
Anillado	3	5,00	15,00
Impresiones	400	0,10	40
Servicios Básicos	900 horas	0,25	575,00
Servicio de internet	800horas	0,02	16,00
Alimentación	90 días	1,50	535,00
Transporte	90 días	0,50	45,00
Seminario de profesionalización	1	520,00	520,00
Tutoría	1	200,00	200,00
Computador	1	700,00	900,00
Total			2.856,00

Nota: El cuadro muestra el presupuesto con el que se cuenta para la elaboración del proyecto.

6.03 Cronograma

Figura: 32 Cronograma de actividades



Nota: El Grafico muestra el cronograma de actividades en la realización del proyecto

CAPITULO VII: CONCLUSIONES Y RECOMENDACIONES

7.01 Conclusión

Concluimos que la presente investigación nos ayudó en lo siguiente:

Con la implementación del sistema de agilización de los procesos de facturación obtendremos la optimización de los procesos operacionales a través de los módulos con el que cuenta el sistema.

El sistema ayudara a tener la información sistematizada debido al exceso de pérdida de documentos agilizando los procesos.

El sistema tendrá una interfaz gráfica que nos permitirá visualizar toda la información del cliente.

Se elaboró un manual de usuario del sistema de agilización de los procesos de facturación con una fácil comprensión y gráficas ilustrativas que permita el adecuado uso de la aplicación.

7.02 Recomendación

Con el fin de poder cumplir los requerimientos que la institución necesita se recomienda.

Obtener un control de trabajo asignado a cada área a través de una adecuada organización.

Capacitar a los usuarios del programa con el objetivo de que conozcan y utilicen adecuadamente el software.

Seguir de una manera sistemática los procedimientos que se exponen en el manual de usuario para conseguir un correcto funcionamiento y evitar una saturación del sistema.



ANEXOS

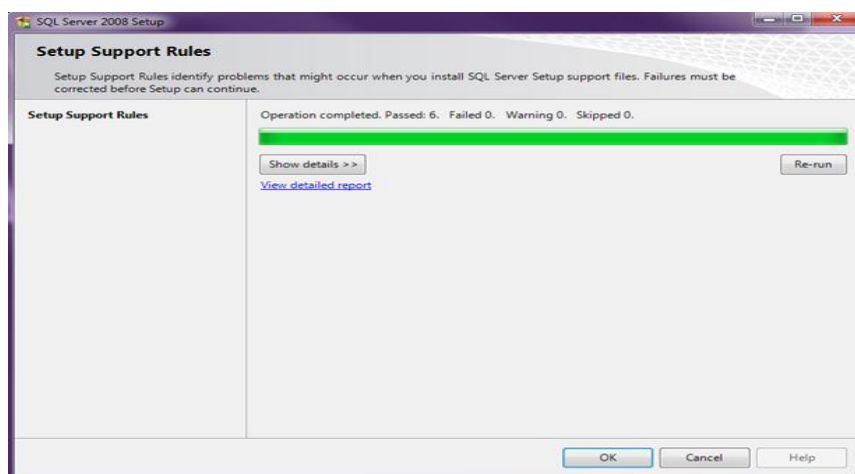
8. Instalación de programas utilizados

8.01 Instalación del SQL Server Management

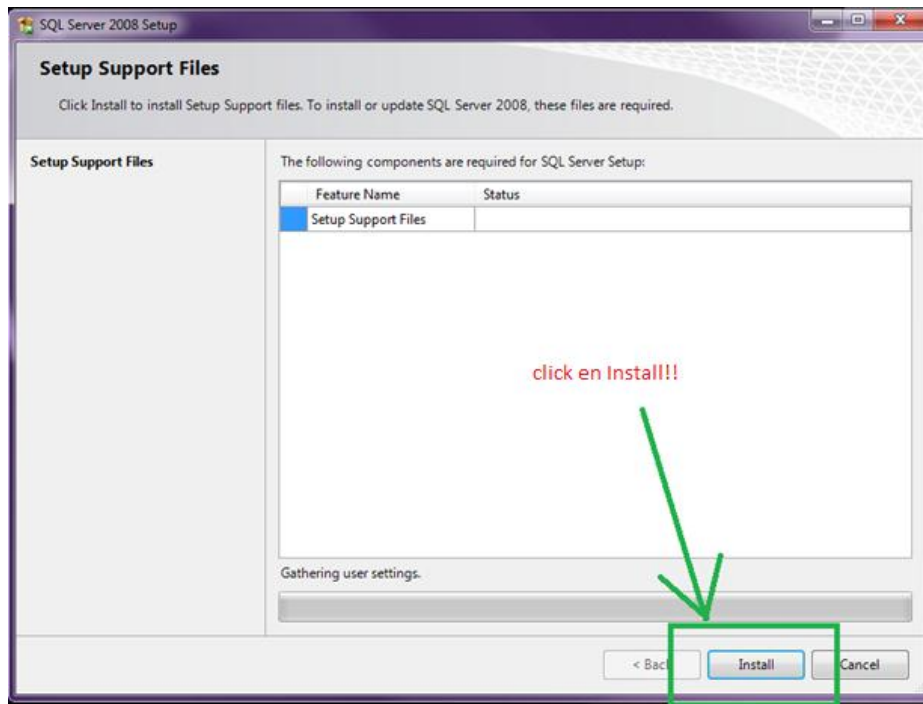
En el "SQL Server Installation Center", ingresar en Installation.



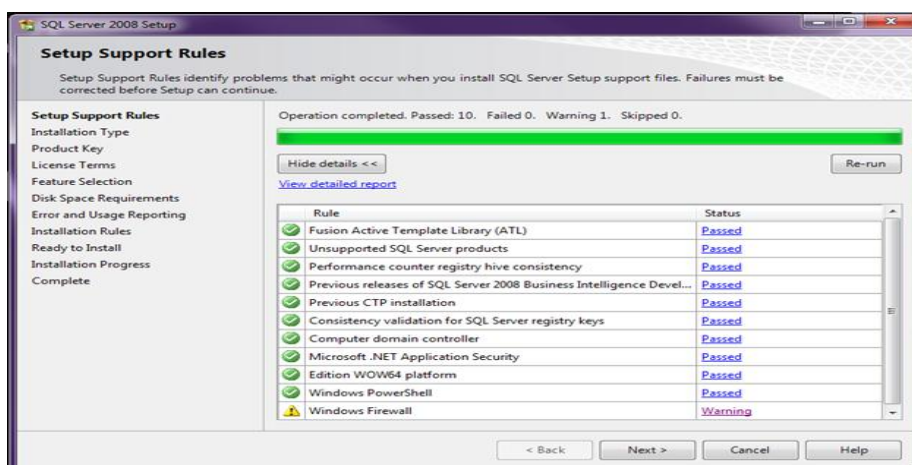
Seleccionar la primera opción "New SQL Server stand-alone installation or add features to an existing installation", va a realizar un chequeo de reglas y si no hay errores presionar ok.



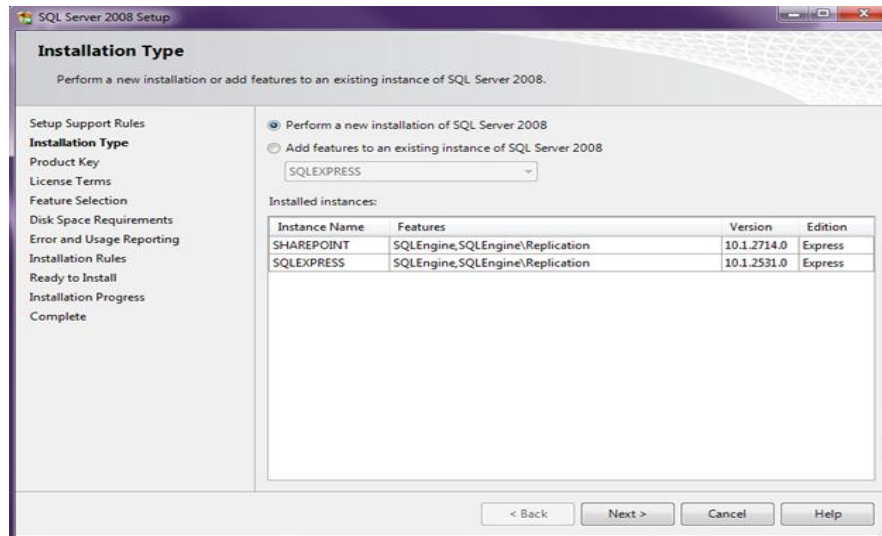
Seguir adelante haciendo click en *Install*.



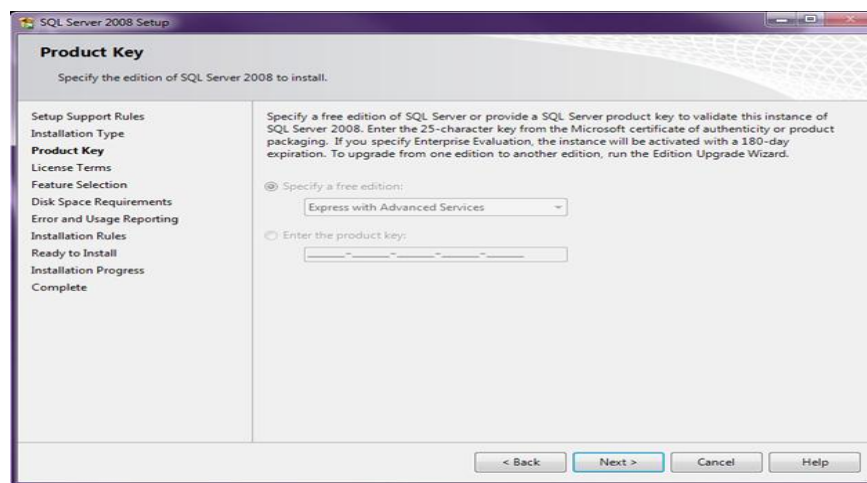
Se ejecutan otras comprobaciones, probablemente aparezca un warning del firewall como se muestra en la siguiente imagen. Si se necesita acceso remoto al SQL es necesario configurar el firewall luego en este caso seleccionar "Next".



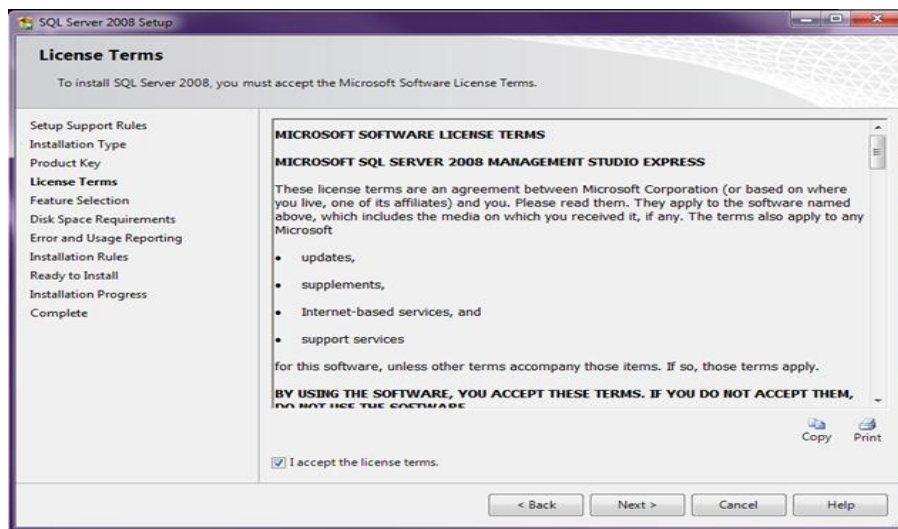
En la ventana "Installation Type", aparecen dos opciones: "Perform a new installation of SQL Server 2008" ó "Add features to an existing instance of SQL Server 2008", luego click en Next.



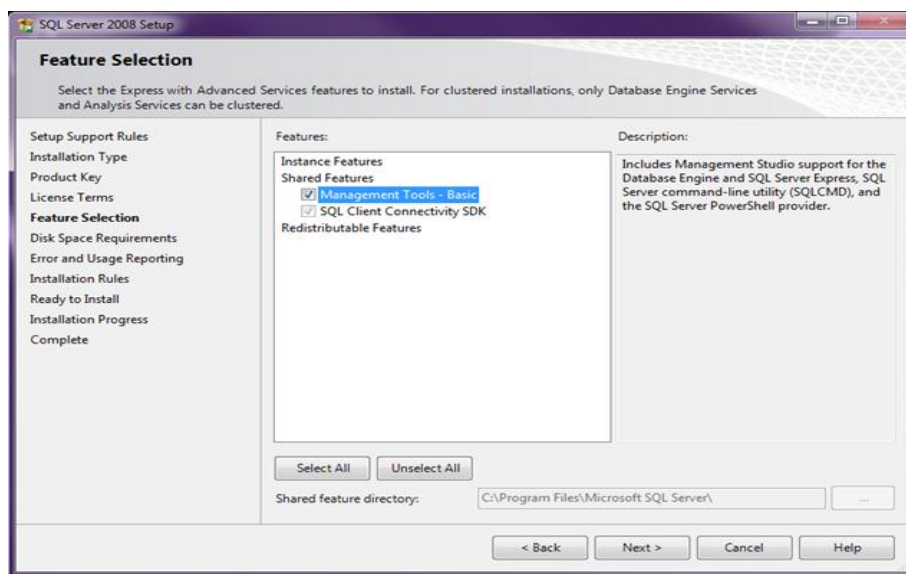
Hacemos click en Next.



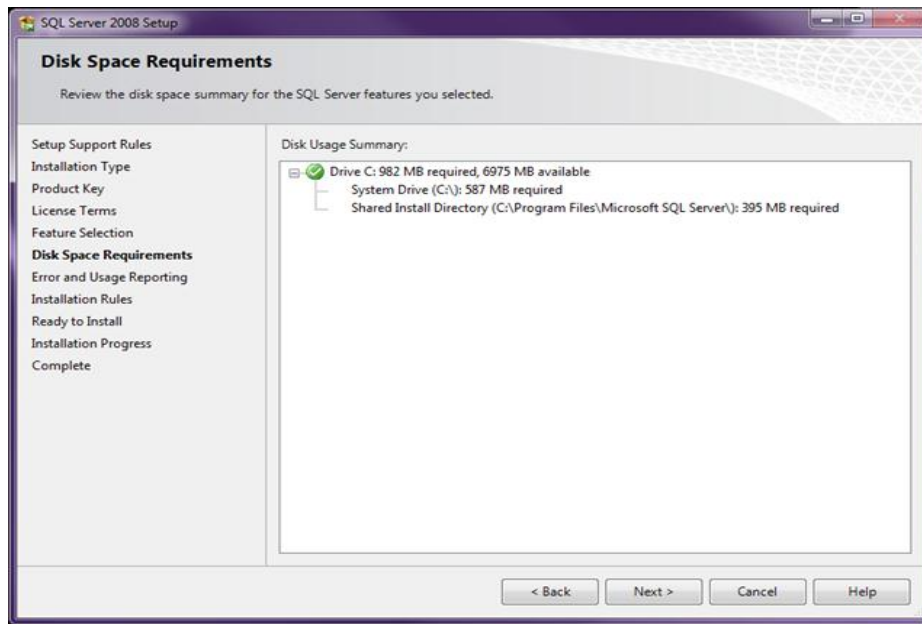
Aceptar los términos de la licencia y click en *Next*.



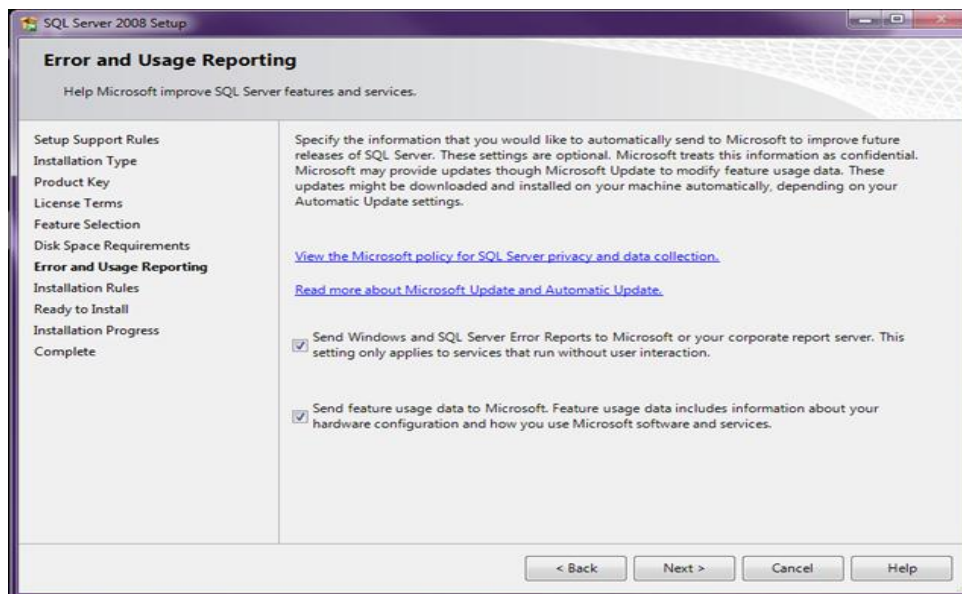
En la siguiente ventana aparece la opción de seleccionar nuevas características. Por supuesto elegir "Management tools - Basic" y click en *Next*.



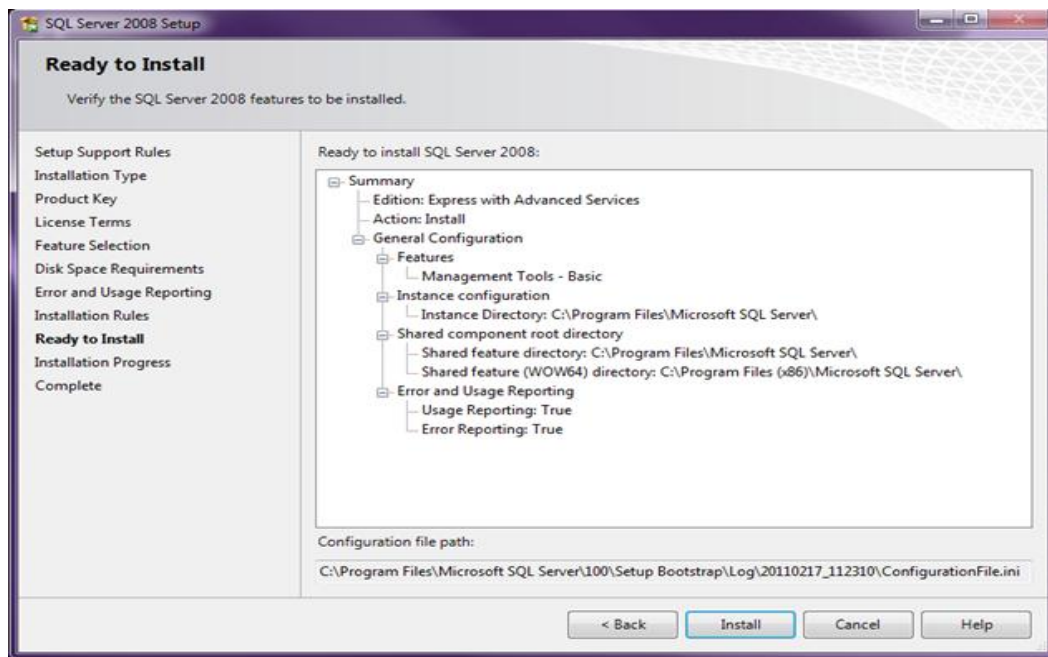
Verificación de espacio en disco. Click *Next*.



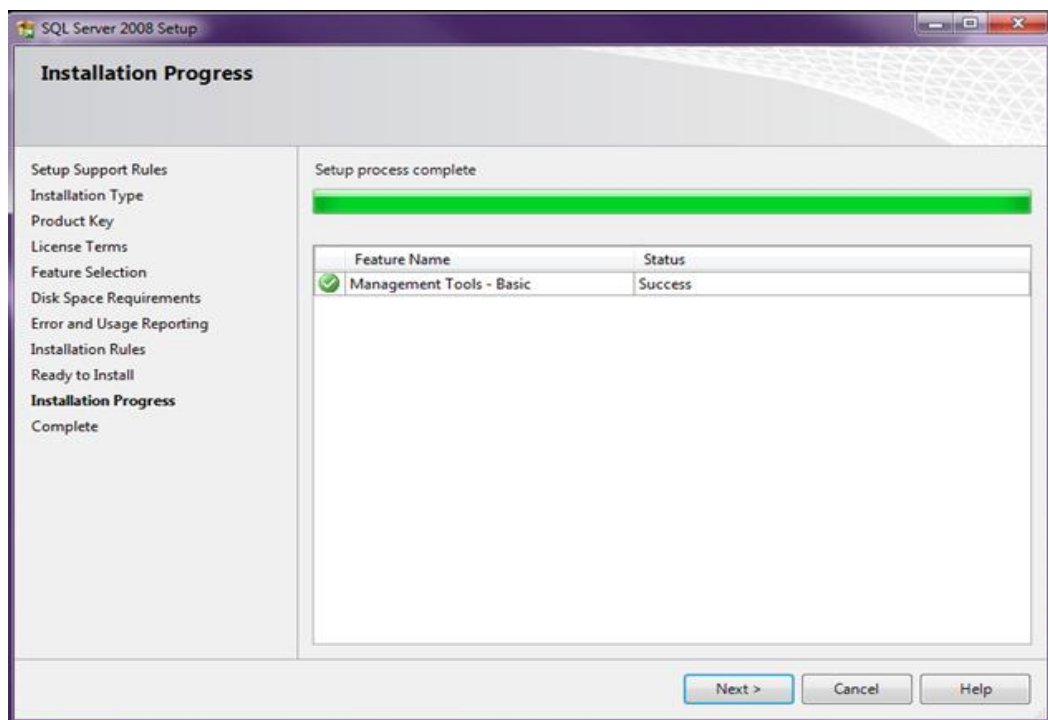
Seleccionar si se desea enviar reportes de error a Microsoft. Click *Next*.



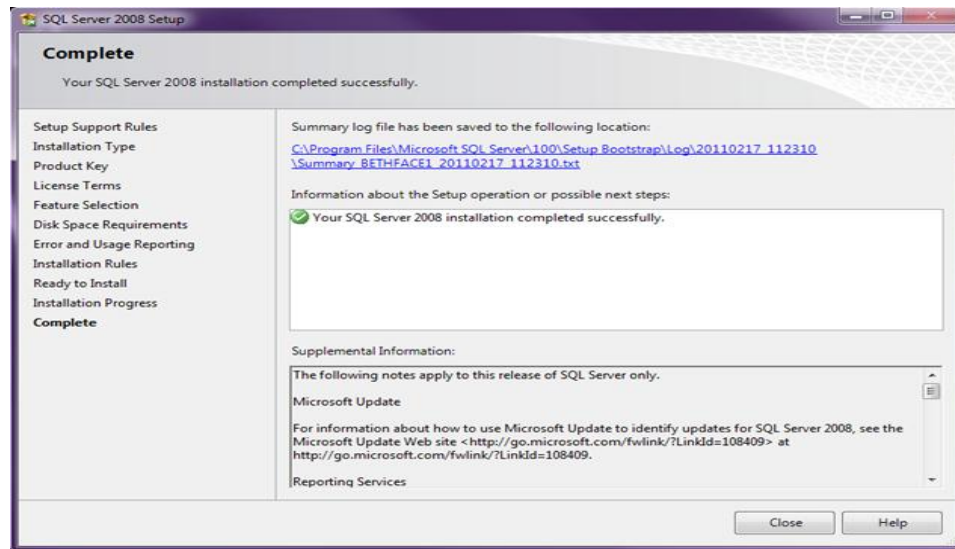
Hacemos click en *INSTALL*.



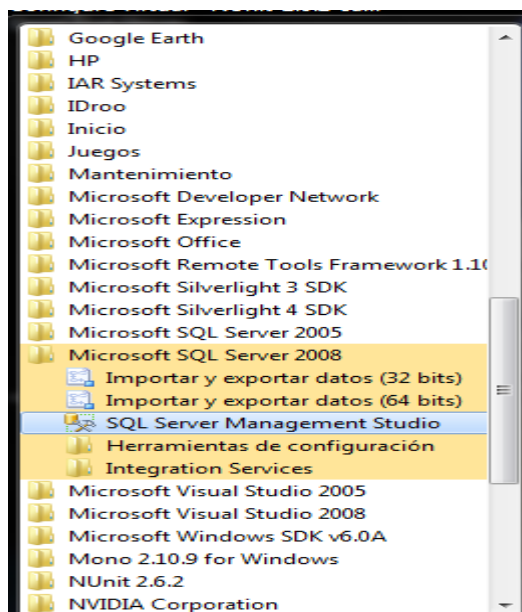
Cuando la instalación termina click *Next*.



Y por último hacemos click en *Close*.



Podemos verificar la instalación en el menú inicio como se ve en la siguiente imagen



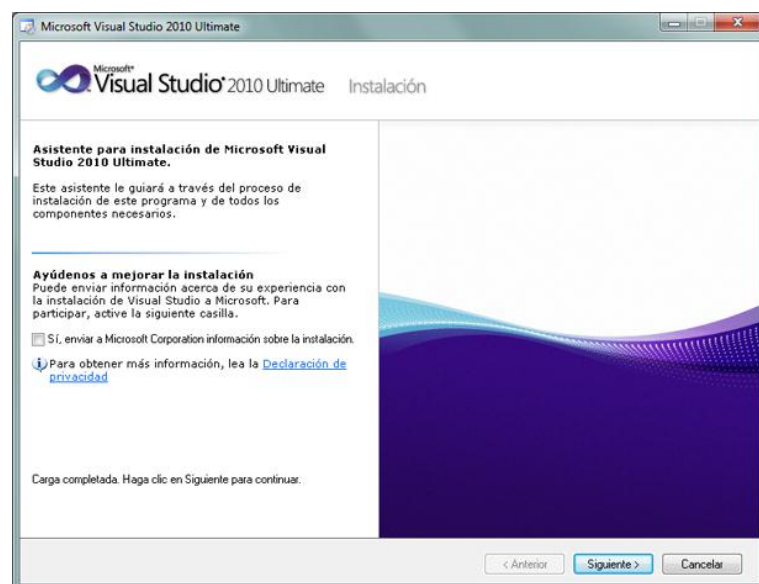
8.02 Instalación de Visual Studio 2010

Se iniciará el asistente para la instalación de Microsoft Visual Studio 2010

Pulsaremos en "Instalar Microsoft Visual Studio 2010":

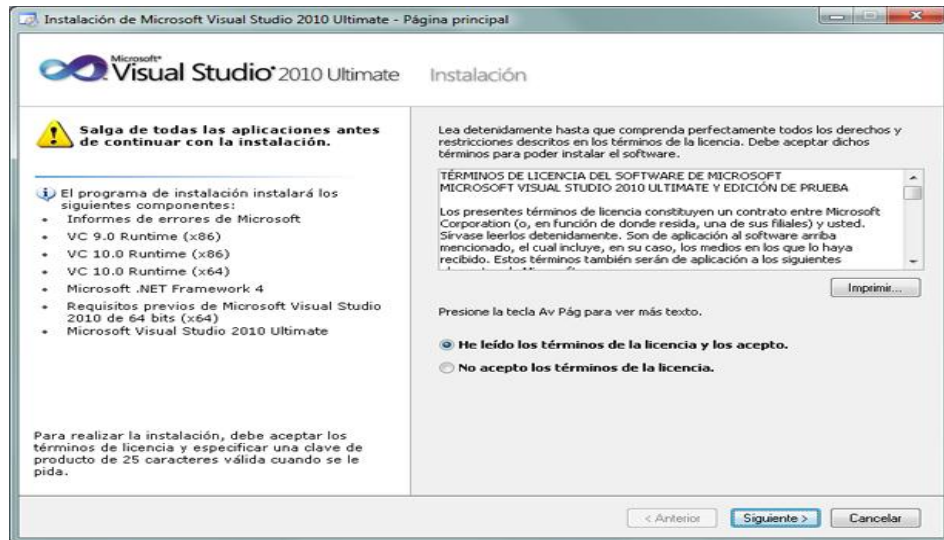


En el primer paso de la instalación de Microsoft Visual Studio 2010 Ultimate, desmarcaremos "Sí, enviar a Microsoft Corporation información sobre la instalación" (si no queremos enviar esta información) y pulsaremos "Siguiente":

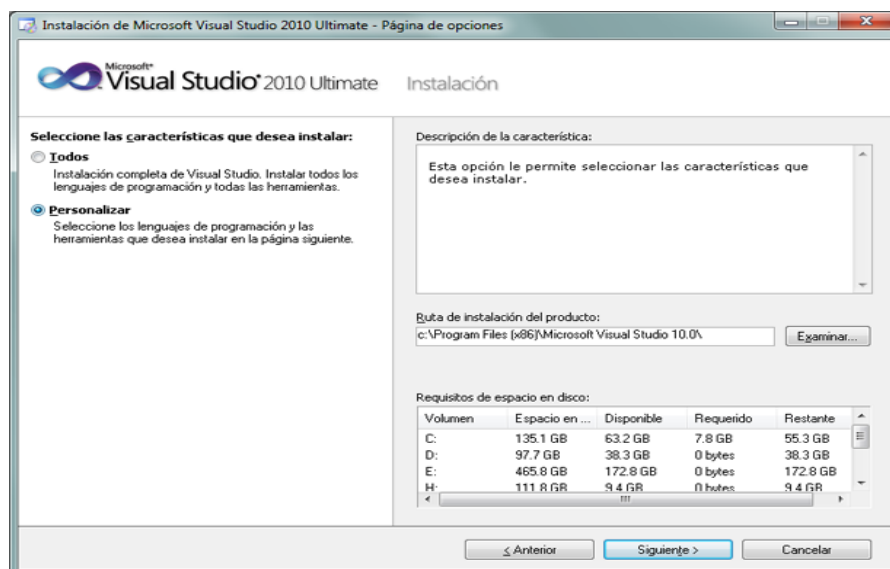


Leeremos los términos de licencia del software de Microsoft Visual Studio 2010

Ultimate. Si estamos de acuerdo marcaremos "He leído los términos de la licencia y los acepto". Pulsaremos "Siguiente" para continuar:



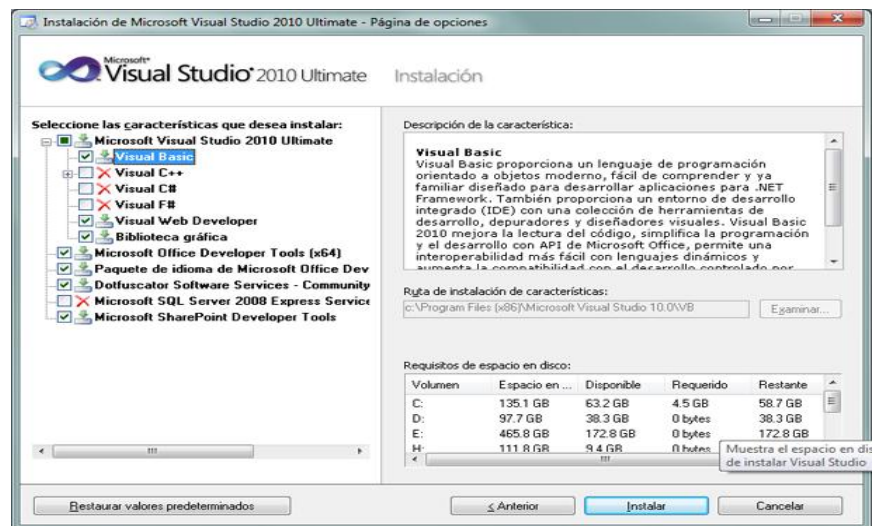
A continuación marcaremos "Personalizar" para seleccionar los lenguajes de programación y las herramientas que se quieran instalar. En "Ruta de instalación del producto" indicaremos la unidad y carpeta de destino de la instalación:



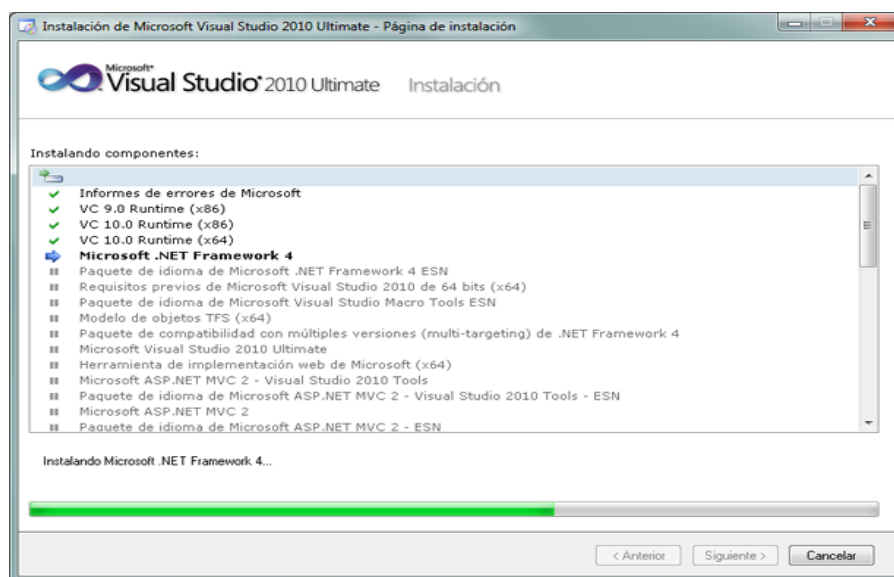
Seleccionaremos los lenguajes a instalar:

Seleccionaremos también las características a instalar:

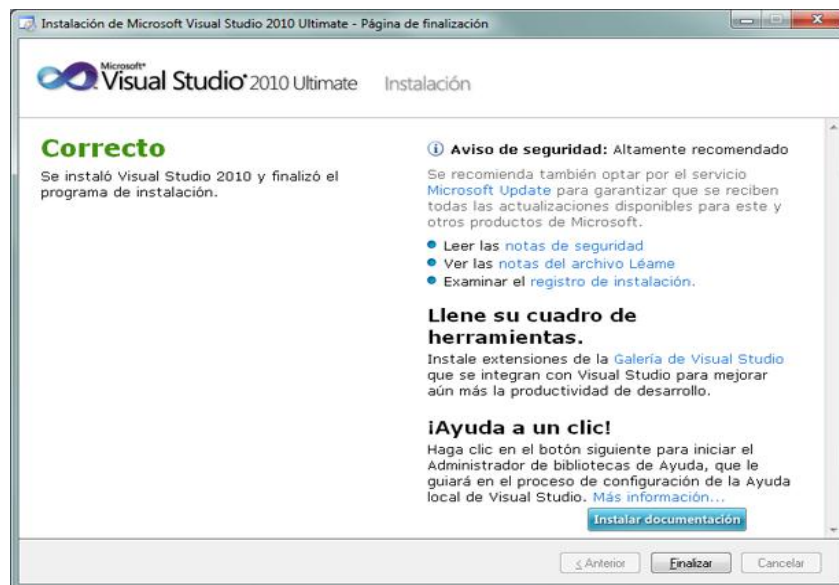
Una vez seleccionadas las características a instalar pulsaremos en el botón "Instalar":



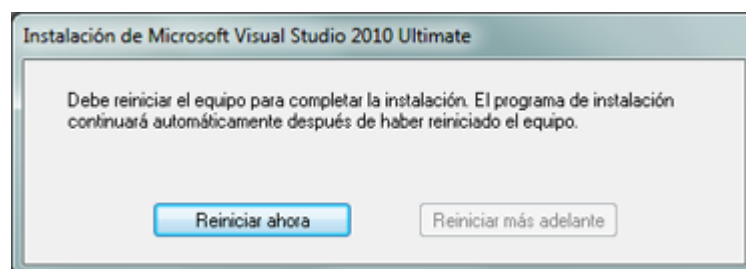
Se iniciará la instalación de Microsoft Visual Studio 2010 Ultimate: informe de errores de Microsoft, VC 9.0 Runtime, VC 10.0 Runtime, Microsoft .Net Framework 4, Microsoft Visual Studio 2010 Ultimate, Microsoft ASP .Net, etc.:



Tras la instalación, el asistente para instalar Visual Studio mostrará la siguiente ventana, con el texto "Correcto. Se instaló Visual Studio 2010 y finalizó el programa de instalación". Pulsaremos "Finalizar":

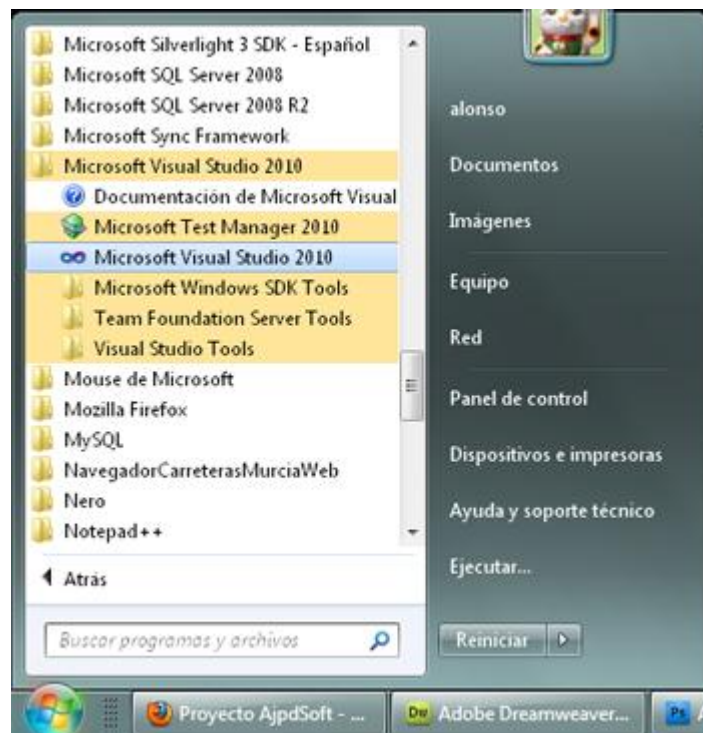


Tras el proceso de instalación puede que pida reiniciar el equipo:

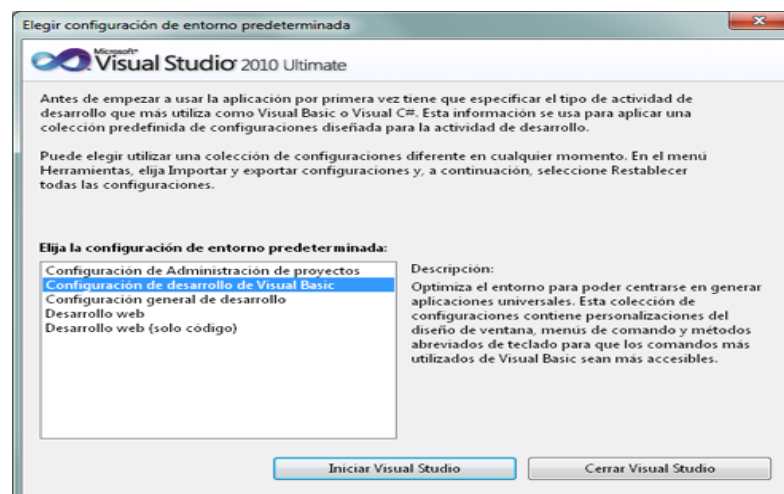


Con el texto: "Debe reiniciar el equipo para completar la instalación. El programa de instalación continuará automáticamente después de haber reiniciado el equipo".

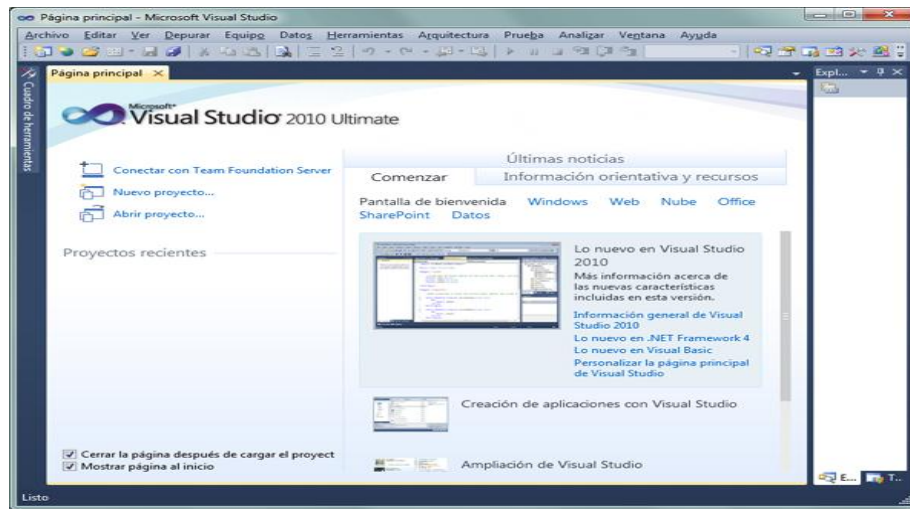
Desde el botón "Iniciar" - "Todos los programas" podremos ver el grupo de programas "Microsoft Visual Studio 2010":



Tras iniciar por primera vez Microsoft Visual Studio 2010 Ultimate, seleccionaremos la configuración de entorno predeterminada, por ejemplo "Configuración de desarrollo de Visual Basic":



Y ya podremos disponer de este entorno de programación:



8.03 Script de la Base de Datos

/*

Navicat SQL Server Data Transfer

```
Source Server           : MSSQLSERVER
Source Server Version   : 105000
Source Host             : DEVELOPER-PC:1433
Source Database         : BD_CENDOCAP6
Source Schema          : dbo
```

```
Target Server Type      : SQL Server
Target Server Version   : 105000
File Encoding           : 65001
```

Date: 2014-04-08 19:35:03

*/

```
-- -----
-- Table structure for AudDTab
-- -----
DROP TABLE [dbo].[AudDTab]
GO
CREATE TABLE [dbo].[AudDTab] (
  [AudDTabId] smallint NOT NULL ,
  [AudDTabTabla] varchar(50) NOT NULL ,
  [AudDTabFh] datetime NOT NULL ,
  [AudDTabCampo] varchar(50) NOT NULL ,
  [AudDTabValor] varchar(200) NOT NULL ,
  [AudDTabUsr] varchar(20) NOT NULL ,
  [AudDTabIP] char(18) NOT NULL
)
```

GO



```
-- -----  
-- Table structure for AudmTab  
-- -----  
  
DROP TABLE [dbo].[AudmTab]  
GO  
CREATE TABLE [dbo].[AudmTab] (  
    [AudmTabId] smallint NOT NULL ,  
    [AudmTabTabla] varchar(50) NOT NULL ,  
    [AudmTabFh] datetime NOT NULL ,  
    [AudmTabCampo] varchar(20) NOT NULL ,  
    [AudmTabUsr] varchar(20) NOT NULL ,  
    [AudmTabVantiguo] varchar(200) NOT NULL ,  
    [AudmTabVnuevo] varchar(200) NOT NULL ,  
    [AudmTabIP] char(18) NOT NULL  
)  
  
GO  
  
-- -----  
-- Table structure for AulaTab  
-- -----  
  
DROP TABLE [dbo].[AulaTab]  
GO  
CREATE TABLE [dbo].[AulaTab] (  
    [InstTabId] smallint NOT NULL ,  
    [AulaTabId] int NOT NULL ,  
    [AulaTabDescrip] varchar(50) NOT NULL ,  
    [AulaTabEstado] char(1) NOT NULL ,  
    [AulaTabUsr] varchar(20) NOT NULL ,  
    [AulaTabFh] datetime NOT NULL  
)  
  
GO  
  
-- -----  
-- Table structure for CiudTab  
-- -----  
  
DROP TABLE [dbo].[CiudTab]  
GO  
CREATE TABLE [dbo].[CiudTab] (  
    [CiudTabId] varchar(10) NOT NULL ,  
    [ProvTabId] varchar(10) NOT NULL ,  
    [PaisTabId] varchar(10) NOT NULL ,  
    [CiudTabNom] varchar(50) NOT NULL ,  
    [CiudTabUsr] varchar(20) NOT NULL ,  
    [CiudTabFh] datetime NOT NULL  
)  
  
GO  
  
-- -----  
-- Table structure for CursTab  
-- -----  
  
DROP TABLE [dbo].[CursTab]  
GO  
CREATE TABLE [dbo].[CursTab] (  

```




```
[CursTabId] int NOT NULL ,
[InstTabId] smallint NOT NULL ,
[CursTabDesc] varchar(50) NOT NULL ,
[CursTabCosto] smallmoney NOT NULL ,
[CursTabInscrip] int NOT NULL ,
[CursTabUsr] varchar(20) NOT NULL ,
[CursTabFHR] datetime NOT NULL ,
[CursTabEstado] char(1) NOT NULL ,
[DocenteId] int NOT NULL ,
[CursTabCupos] smallint NULL
)

GO

-- -----
-- Table structure for FactsecTab
-- -----
DROP TABLE [dbo].[FactsecTab]
GO
CREATE TABLE [dbo].[FactsecTab] (
[InstTabId] smallint NOT NULL ,
[FactsecTabId] smallint NOT NULL ,
[FactsecTabAutoriz] char(13) NOT NULL ,
[FactsecTabFin] int NOT NULL ,
[FactsecTabIni] int NOT NULL ,
[FactsecTabEst] char(1) NOT NULL
)

GO

-- -----
-- Table structure for HoraTab
-- -----
DROP TABLE [dbo].[HoraTab]
GO
CREATE TABLE [dbo].[HoraTab] (
[HoraTabId] int NOT NULL ,
[InstTabId] smallint NOT NULL ,
[HoraTabDesc] char(1) NOT NULL ,
[HoraTabHorI] datetime NOT NULL ,
[HoraTabHorF] datetime NOT NULL ,
[HoraTabUsr] varchar(20) NOT NULL ,
[HoraTabEst] char(1) NOT NULL ,
[HoraTabFHR] datetime NOT NULL
)

GO

-- -----
-- Table structure for HSesTab
-- -----
DROP TABLE [dbo].[HSesTab]
GO
CREATE TABLE [dbo].[HSesTab] (
[HSesTabId] varchar(20) NOT NULL ,
[HSesTabFec] datetime NOT NULL ,
```




```
[HSesTabCont] int NOT NULL ,  
[HSesTabFH] datetime NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for InstTab  
-- -----
```

```
DROP TABLE [dbo].[InstTab]  
GO  
CREATE TABLE [dbo].[InstTab] (  
[InstTabId] smallint NOT NULL ,  
[InstTabRazS] varchar(100) NOT NULL ,  
[InstTabRuc] varchar(13) NOT NULL ,  
[InstTabPropietario] varchar(50) NOT NULL ,  
[InstTabDir] varchar(100) NOT NULL ,  
[InstTabTelf1] varchar(20) NOT NULL ,  
[InstTabTelf2] varchar(20) NOT NULL ,  
[InstTabLogo] varbinary(MAX) NOT NULL ,  
[InstTabEstado] char(1) NOT NULL ,  
[InstTabUsr] varchar(20) NOT NULL ,  
[InstTabFh] datetime NOT NULL ,  
[PaistabId] varchar(10) NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for LogUTab  
-- -----
```

```
DROP TABLE [dbo].[LogUTab]  
GO  
CREATE TABLE [dbo].[LogUTab] (  
[InstTabId] smallint NOT NULL ,  
[SociPrintTabId] int NOT NULL ,  
[LogUTabId] char(1) NOT NULL ,  
[LogUTabLogin] varchar(20) NOT NULL ,  
[LogUTabPass] varchar(200) NOT NULL ,  
[LogUTabTipo] char(1) NOT NULL ,  
[LogUTabEstado] char(1) NOT NULL ,  
[LogUTabUsuario] varchar(20) NOT NULL ,  
[LogUTabFecha] datetime NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for MatricTab  
-- -----
```

```
DROP TABLE [dbo].[MatricTab]  
GO  
CREATE TABLE [dbo].[MatricTab] (  
[InstTabId] smallint NOT NULL ,  
[SociPrintTabId] int NOT NULL ,  
[CursTabId] int NOT NULL ,
```



```
[HoraTabId] int NOT NULL ,
[AulaTabId] int NOT NULL ,
[MatricTabNumrFactr] varchar(10) NOT NULL ,
[MatricTabId] int NOT NULL ,
[PeriTabId] int NOT NULL ,
[MatricTabHorF] datetime NOT NULL ,
[MatricTabUsr] varchar(20) NOT NULL ,
[MatricTabVal] smallmoney NOT NULL ,
[MatricTabFc] datetime NOT NULL
)
```

GO

```
-- -----
-- Table structure for PagoTab
-- -----
DROP TABLE [dbo].[PagoTab]
GO
CREATE TABLE [dbo].[PagoTab] (
[PagoTabId] int NOT NULL ,
[MatricTabId] int NOT NULL ,
[PagoTabFh] datetime NOT NULL ,
[PagoTabVal] smallmoney NOT NULL ,
[PagoTabUsr] varchar(20) NOT NULL ,
[PagoTabFhr] datetime NOT NULL ,
[PeriTabId] int NOT NULL ,
[MatricTabNumrFactr] varchar(10) NOT NULL ,
[AulaTabId] int NOT NULL ,
[HoraTabId] int NOT NULL ,
[CursTabId] int NOT NULL ,
[SociPrintTabId] int NOT NULL ,
[InstTabId] smallint NOT NULL
)
```

GO

```
-- -----
-- Table structure for PaisTab
-- -----
DROP TABLE [dbo].[PaisTab]
GO
CREATE TABLE [dbo].[PaisTab] (
[PaisTabId] varchar(10) NOT NULL ,
[PaisTabNom] varchar(50) NOT NULL ,
[PaisTabUsr] varchar(20) NOT NULL ,
[PaisTabFh] datetime NOT NULL
)
```

GO

```
-- -----
-- Table structure for PaisTabProvTab
-- -----
DROP TABLE [dbo].[PaisTabProvTab]
GO
CREATE TABLE [dbo].[PaisTabProvTab] (
```



```
[PaisTabId] varchar(10) NOT NULL ,  
[ProvTabId] varchar(10) NOT NULL ,  
[ProvTabNom] varchar(50) NOT NULL ,  
[ProvTabUsr] varchar(50) NOT NULL ,  
[ProvTabFh] datetime NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for ParmUtab  
-- -----
```

```
DROP TABLE [dbo].[ParmUtab]  
GO  
CREATE TABLE [dbo].[ParmUtab] (  
[ParmUtabId] smallint NOT NULL ,  
[ParmUtabSite] varchar(60) NOT NULL ,  
[ParmUtabPass] varchar(200) NOT NULL ,  
[ParmUtabLang] char(1) NOT NULL ,  
[ParmUtabPath] varchar(512) NOT NULL ,  
[ParmUtabLogo] varbinary(MAX) NOT NULL ,  
[ParmUtabBirt] varchar(30) NOT NULL ,  
[ParmUtabTemp] varchar(512) NOT NULL ,  
[ParmUtabLic] varchar(200) NOT NULL ,  
[ParmUtabTiem] smallint NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for PeriTab  
-- -----
```

```
DROP TABLE [dbo].[PeriTab]  
GO  
CREATE TABLE [dbo].[PeriTab] (  
[PeriTabId] int NOT NULL ,  
[InstTabId] smallint NOT NULL ,  
[PeriTabDesc] varchar(40) NOT NULL ,  
[PeriTabIni] datetime NOT NULL ,  
[PeriTabFin] datetime NOT NULL ,  
[PeriTabEstado] char(1) NOT NULL ,  
[PeriTabUsu] varchar(20) NOT NULL ,  
[PeriTabFH] datetime NOT NULL  
)
```

GO

```
-- -----  
-- Table structure for SessTab  
-- -----
```

```
DROP TABLE [dbo].[SessTab]  
GO  
CREATE TABLE [dbo].[SessTab] (  
[SessTabLog] varchar(40) NOT NULL ,  
[SessTabIp] varchar(18) NOT NULL ,  
[SessTabFH] datetime NOT NULL
```

```
)

GO

-- -----
-- Table structure for SociPrintTab
-- -----
DROP TABLE [dbo].[SociPrintTab]
GO
CREATE TABLE [dbo].[SociPrintTab] (
[SociPrintTabId] int NOT NULL ,
[SociPrintTabRazS] varchar(50) NOT NULL ,
[SociPrintTabTipoIden] char(1) NOT NULL ,
[SociPrintTabIden] varchar(20) NOT NULL ,
[SociPrintTabTipo] char(1) NOT NULL ,
[SociPrintTabDir] varchar(100) NOT NULL ,
[SociPrintTabTelf1] varchar(20) NOT NULL ,
[SociPrintTabTelf2] varchar(20) NOT NULL ,
[SociPrintTabEmail] varchar(50) NOT NULL ,
[SociPrintTabEstado] char(1) NOT NULL ,
[SociPrintTabUsr] varchar(20) NOT NULL ,
[SociPrintTabFh] datetime NOT NULL ,
[CiudTabId] varchar(10) NOT NULL ,
[ProvTabId] varchar(10) NOT NULL ,
[PaísTabId] varchar(10) NOT NULL
)

GO

-- -----
-- Indexes structure for table AudDTab
-- -----

-- -----
-- Primary Key structure for table AudDTab
-- -----
ALTER TABLE [dbo].[AudDTab] ADD PRIMARY KEY ([AudDTabId],
[AudDTabTabla], [AudDTabFh], [AudDTabCampo])
GO

-- -----
-- Indexes structure for table AudmTab
-- -----

-- -----
-- Primary Key structure for table AudmTab
-- -----
ALTER TABLE [dbo].[AudmTab] ADD PRIMARY KEY ([AudmTabId],
[AudmTabTabla], [AudmTabFh], [AudmTabCampo])
GO

-- -----
-- Indexes structure for table AulaTab
-- -----

-- -----
-- Primary Key structure for table AulaTab
```



```
-- -----  
ALTER TABLE [dbo].[AulaTab] ADD PRIMARY KEY ([InstTabId],  
[AulaTabId])  
GO  
  
-- -----  
-- Indexes structure for table CiudTab  
-- -----  
CREATE INDEX [ICIUDTAB1] ON [dbo].[CiudTab]  
([PaisTabId] ASC, [ProvTabId] ASC)  
GO  
  
-- -----  
-- Primary Key structure for table CiudTab  
-- -----  
ALTER TABLE [dbo].[CiudTab] ADD PRIMARY KEY ([CiudTabId],  
[ProvTabId], [PaisTabId])  
GO  
  
-- -----  
-- Indexes structure for table CursTab  
-- -----  
CREATE INDEX [ICURSTAB2] ON [dbo].[CursTab]  
([InstTabId] ASC)  
GO  
  
-- -----  
-- Primary Key structure for table CursTab  
-- -----  
ALTER TABLE [dbo].[CursTab] ADD PRIMARY KEY ([CursTabId],  
[InstTabId])  
GO  
  
-- -----  
-- Indexes structure for table FactsecTab  
-- -----  
  
-- -----  
-- Primary Key structure for table FactsecTab  
-- -----  
ALTER TABLE [dbo].[FactsecTab] ADD PRIMARY KEY ([InstTabId],  
[FactsecTabId])  
GO  
  
-- -----  
-- Indexes structure for table HoraTab  
-- -----  
CREATE INDEX [IHORATAB1] ON [dbo].[HoraTab]  
([InstTabId] ASC)  
GO  
  
-- -----  
-- Primary Key structure for table HoraTab  
-- -----  
ALTER TABLE [dbo].[HoraTab] ADD PRIMARY KEY ([HoraTabId],  
[InstTabId])  
GO  
  
-- -----
```

```
-- Indexes structure for table HSesTab
-- -----
CREATE INDEX [UHSESTAB] ON [dbo].[HSesTab]
([HSesTabFec] DESC)
GO

-- -----
-- Primary Key structure for table HSesTab
-- -----
ALTER TABLE [dbo].[HSesTab] ADD PRIMARY KEY ([HSesTabId],
[HSesTabFec])
GO

-- -----
-- Indexes structure for table InstTab
-- -----
CREATE INDEX [IINSTTAB1] ON [dbo].[InstTab]
([PaisTabId] ASC)
GO

-- -----
-- Primary Key structure for table InstTab
-- -----
ALTER TABLE [dbo].[InstTab] ADD PRIMARY KEY ([InstTabId])
GO

-- -----
-- Indexes structure for table LogUTab
-- -----
CREATE INDEX [ILOGUTAB1] ON [dbo].[LogUTab]
([SociPrintTabId] ASC)
GO

-- -----
-- Primary Key structure for table LogUTab
-- -----
ALTER TABLE [dbo].[LogUTab] ADD PRIMARY KEY ([InstTabId],
[SociPrintTabId], [LogUTabId])
GO

-- -----
-- Indexes structure for table MatricTab
-- -----
CREATE INDEX [IMATRICTAB1] ON [dbo].[MatricTab]
([PeriTabId] ASC, [InstTabId] ASC)
GO
CREATE INDEX [IMATRICTAB2] ON [dbo].[MatricTab]
([InstTabId] ASC, [AulaTabId] ASC)
GO
CREATE INDEX [IMATRICTAB3] ON [dbo].[MatricTab]
([HoraTabId] ASC, [InstTabId] ASC)
GO
CREATE INDEX [IMATRICTAB4] ON [dbo].[MatricTab]
([CursTabId] ASC, [InstTabId] ASC)
GO
CREATE INDEX [IMATRICTAB5] ON [dbo].[MatricTab]
([SociPrintTabId] ASC)
GO
```



```
-- -----  
-- Primary Key structure for table MatricTab  
-- -----  
ALTER TABLE [dbo].[MatricTab] ADD PRIMARY KEY ([InstTabId],  
[SociPrintTabId], [CursTabId], [HoraTabId], [AulaTabId],  
[MatricTabNumrFactr], [MatricTabId], [PeriTabId])  
GO  
  
-- -----  
-- Indexes structure for table PagoTab  
-- -----  
CREATE INDEX [IPAGOTAB1] ON [dbo].[PagoTab]  
([InstTabId] ASC, [SociPrintTabId] ASC, [CursTabId] ASC, [HoraTabId]  
ASC, [AulaTabId] ASC, [MatricTabNumrFactr] ASC, [MatricTabId] ASC,  
[PeriTabId] ASC)  
GO  
  
-- -----  
-- Primary Key structure for table PagoTab  
-- -----  
ALTER TABLE [dbo].[PagoTab] ADD PRIMARY KEY ([PagoTabId],  
[InstTabId], [SociPrintTabId], [CursTabId], [HoraTabId],  
[AulaTabId], [MatricTabNumrFactr], [MatricTabId], [PeriTabId])  
GO  
  
-- -----  
-- Indexes structure for table PaisTab  
-- -----  
  
-- -----  
-- Primary Key structure for table PaisTab  
-- -----  
ALTER TABLE [dbo].[PaisTab] ADD PRIMARY KEY ([PaisTabId])  
GO  
  
-- -----  
-- Indexes structure for table PaisTabProvTab  
-- -----  
  
-- -----  
-- Primary Key structure for table PaisTabProvTab  
-- -----  
ALTER TABLE [dbo].[PaisTabProvTab] ADD PRIMARY KEY ([PaisTabId],  
[ProvTabId])  
GO  
  
-- -----  
-- Indexes structure for table ParmUtab  
-- -----  
  
-- -----  
-- Primary Key structure for table ParmUtab  
-- -----  
ALTER TABLE [dbo].[ParmUtab] ADD PRIMARY KEY ([ParmUtabId])  
GO  
  
-- -----  
-- Indexes structure for table PeriTab  
-- -----
```

```
CREATE INDEX [IPERITAB1] ON [dbo].[PeriTab]
([InstTabId] ASC)
GO

-- -----
-- Primary Key structure for table PeriTab
-- -----
ALTER TABLE [dbo].[PeriTab] ADD PRIMARY KEY ([PeriTabId],
[InstTabId])
GO

-- -----
-- Indexes structure for table SessTab
-- -----

-- -----
-- Primary Key structure for table SessTab
-- -----
ALTER TABLE [dbo].[SessTab] ADD PRIMARY KEY ([SessTabLog])
GO

-- -----
-- Indexes structure for table SociPrintTab
-- -----
CREATE INDEX [ISOCIPRINTTAB1] ON [dbo].[SociPrintTab]
([CiudTabId] ASC, [ProvTabId] ASC, [PaisTabId] ASC)
GO
CREATE INDEX [USOCIPRINTTAB] ON [dbo].[SociPrintTab]
([SociPrintTabId] ASC, [SociPrintTabRazS] ASC)
GO

-- -----
-- Primary Key structure for table SociPrintTab
-- -----
ALTER TABLE [dbo].[SociPrintTab] ADD PRIMARY KEY ([SociPrintTabId])
GO

-- -----
-- Foreign Key structure for table [dbo].[AulaTab]
-- -----
ALTER TABLE [dbo].[AulaTab] ADD FOREIGN KEY ([InstTabId]) REFERENCES
[dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON UPDATE NO
ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[CiudTab]
-- -----
ALTER TABLE [dbo].[CiudTab] ADD FOREIGN KEY ([PaisTabId],
[ProvTabId]) REFERENCES [dbo].[PaisTabProvTab] ([PaisTabId],
[ProvTabId]) ON DELETE NO ACTION ON UPDATE NO ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[CursTab]
-- -----
ALTER TABLE [dbo].[CursTab] ADD FOREIGN KEY ([InstTabId]) REFERENCES
[dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON UPDATE NO
ACTION
```




GO

```
-- -----  
-- Foreign Key structure for table [dbo].[FactsecTab]  
-- -----  
ALTER TABLE [dbo].[FactsecTab] ADD FOREIGN KEY ([InstTabId])  
REFERENCES [dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON  
UPDATE NO ACTION  
GO  
  
-- -----  
-- Foreign Key structure for table [dbo].[HoraTab]  
-- -----  
ALTER TABLE [dbo].[HoraTab] ADD FOREIGN KEY ([InstTabId]) REFERENCES  
[dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON UPDATE NO  
ACTION  
GO  
  
-- -----  
-- Foreign Key structure for table [dbo].[InstTab]  
-- -----  
ALTER TABLE [dbo].[InstTab] ADD FOREIGN KEY ([PaisTabId]) REFERENCES  
[dbo].[PaisTab] ([PaisTabId]) ON DELETE NO ACTION ON UPDATE NO  
ACTION  
GO  
  
-- -----  
-- Foreign Key structure for table [dbo].[LogUTab]  
-- -----  
ALTER TABLE [dbo].[LogUTab] ADD FOREIGN KEY ([InstTabId]) REFERENCES  
[dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON UPDATE NO  
ACTION  
GO  
ALTER TABLE [dbo].[LogUTab] ADD FOREIGN KEY ([SociPrintTabId])  
REFERENCES [dbo].[SociPrintTab] ([SociPrintTabId]) ON DELETE NO  
ACTION ON UPDATE NO ACTION  
GO  
  
-- -----  
-- Foreign Key structure for table [dbo].[MatricTab]  
-- -----  
ALTER TABLE [dbo].[MatricTab] ADD FOREIGN KEY ([PeriTabId],  
[InstTabId]) REFERENCES [dbo].[PeriTab] ([PeriTabId], [InstTabId])  
ON DELETE NO ACTION ON UPDATE NO ACTION  
GO  
ALTER TABLE [dbo].[MatricTab] ADD FOREIGN KEY ([InstTabId],  
[AulaTabId]) REFERENCES [dbo].[AulaTab] ([InstTabId], [AulaTabId])  
ON DELETE NO ACTION ON UPDATE NO ACTION  
GO  
ALTER TABLE [dbo].[MatricTab] ADD FOREIGN KEY ([HoraTabId],  
[InstTabId]) REFERENCES [dbo].[HoraTab] ([HoraTabId], [InstTabId])  
ON DELETE NO ACTION ON UPDATE NO ACTION  
GO  
ALTER TABLE [dbo].[MatricTab] ADD FOREIGN KEY ([CursTabId],  
[InstTabId]) REFERENCES [dbo].[CursTab] ([CursTabId], [InstTabId])  
ON DELETE NO ACTION ON UPDATE NO ACTION  
GO
```

```

ALTER TABLE [dbo].[MatricTab] ADD FOREIGN KEY ([SociPrintTabId])
REFERENCES [dbo].[SociPrintTab] ([SociPrintTabId]) ON DELETE NO
ACTION ON UPDATE NO ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[PagoTab]
-- -----
ALTER TABLE [dbo].[PagoTab] ADD FOREIGN KEY ([InstTabId],
[SociPrintTabId], [CursTabId], [HoraTabId], [AulaTabId],
[MatricTabNumrFactr], [MatricTabId], [PeriTabId]) REFERENCES
[dbo].[MatricTab] ([InstTabId], [SociPrintTabId], [CursTabId],
[HoraTabId], [AulaTabId], [MatricTabNumrFactr], [MatricTabId],
[PeriTabId]) ON DELETE NO ACTION ON UPDATE NO ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[PaisTabProvTab]
-- -----
ALTER TABLE [dbo].[PaisTabProvTab] ADD FOREIGN KEY ([PaisTabId])
REFERENCES [dbo].[PaisTab] ([PaisTabId]) ON DELETE NO ACTION ON
UPDATE NO ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[PeriTab]
-- -----
ALTER TABLE [dbo].[PeriTab] ADD FOREIGN KEY ([InstTabId]) REFERENCES
[dbo].[InstTab] ([InstTabId]) ON DELETE NO ACTION ON UPDATE NO
ACTION
GO

-- -----
-- Foreign Key structure for table [dbo].[SociPrintTab]
-- -----
ALTER TABLE [dbo].[SociPrintTab] ADD FOREIGN KEY ([CiudTabId],
[ProvTabId], [PaisTabId]) REFERENCES [dbo].[CiudTab] ([CiudTabId],
[ProvTabId], [PaisTabId]) ON DELETE NO ACTION ON UPDATE NO ACTION
GO

```

8.04 Diccionario de Datos

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Estos diccionarios se desarrollan durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

III PDM Diagrams

III.1 Model level diagrams

III.1.1 Diagram PhysicalDiagram_1

Name	CENDOCAP
Code	CENDOCAP
Comment	
DBMS	Microsoft SQL Server 2008
Database	<None>

Model level references

List of references

Name	Code	Parent Table	Child Table
FK__CiudTab__4E88ABD4	FK__CiudTab__4E88ABD4	PaisTabProvTab	CiudTab
FK__CursTab__InstTab__4F7CD00D	FK__CursTab__InstTab__4F7CD00D	InstTab	CursTab
FK__HoraTab__InstTab__52593CB8	FK__HoraTab__InstTab__52593CB8	InstTab	HoraTab
FK__InstTab__PaisTab__571DF1D5	FK__InstTab__PaisTab__571DF1D5	PaisTab	InstTab
FK__LogUTab__InstTab	FK__LogUTab__InstTab	InstTab	LogUTab



__59063A47	__59063A47		
FK__LogUTab__SociPri__5812160E	FK__LogUTab__SociPri__5812160E	SociPrintTab	LogUTab
FK__PaisTabPr__PaisT__5BE2A6F2	FK__PaisTabPr__PaisT__5BE2A6F2	PaisTab	PaisTabProvTab
FK__SociPrintTab__5DCAEF64	FK__SociPrintTab__5DCAEF64	CiudTab	SociPrintTab
IAULATAB1	IAULATAB1	InstTab	AulaTab
IMATRICTAB1	IMATRICTAB1	PeriTab	MatricTab
IMATRICTAB2	IMATRICTAB2	AulaTab	MatricTab
IMATRICTAB3	IMATRICTAB3	HoraTab	MatricTab
IMATRICTAB4	IMATRICTAB4	CursTab	MatricTab
IMATRICTAB5	IMATRICTAB5	SociPrintTab	MatricTab
IPERITAB1	IPERITAB1	InstTab	PeriTab
Reference_16	Reference_16	MatricTab	PagoTab

Reference FK__CiudTab__4E88ABD4

Card of reference FK__CiudTab__4E88ABD4

Name	FK__CiudTab__4E88ABD4
Code	FK__CiudTab__4E88ABD4
Parent Table	PaisTabProvTab
Child Table	CiudTab

Code preview of reference FK__CiudTab__4E88ABD4

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('dbo.CiudTab') and o.name = 'FK__CiudTab__4E88ABD4')

alter table dbo.CiudTab

drop constraint FK__CiudTab__4E88ABD4

go



```
alter table dbo.CiudTab
```

```
add constraint FK__CiudTab__4E88ABD4 foreign key (PaisTabId, ProvTabId)
```

```
references dbo.PaisTabProvTab (PaisTabId, ProvTabId)
```

```
go
```

List of diagrams containing the reference FK__CiudTab__4E88ABD4

Name	Code
PhysicalDiagram_1	PHYSICALDIAGRAM_1

List of reference joins of the reference FK__CiudTab__4E88ABD4

Parent Table Column	Child Table Column
PaisTabId	PaisTabId
ProvTabId	ProvTabId

List of extended attributes of the reference FK__CiudTab__4E88ABD4

Name	Data Type	Value	Target Name
ExtFkNotForReplication	(Boolean)	false	Microsoft SQL Server 2008

Reference FK__CursTab__InstTab__4F7CD00D

Card of reference FK__CursTab__InstTab__4F7CD00D

Name	FK__CursTab__InstTab__4F7CD00D
Code	FK__CursTab__InstTab__4F7CD00D
Parent Table	InstTab
Child Table	CursTab

Code preview of reference FK__CursTab__InstTab__4F7CD00D

```
if exists (select 1
```

```
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
```

```
where r.fkeyid = object_id('dbo.CursTab') and o.name = 'FK__CursTab__InstTab__4F7CD00D')
```

```
alter table dbo.CursTab
```

```
drop constraint FK__CursTab__InstTab__4F7CD00D
go

alter table dbo.CursTab
add constraint FK__CursTab__InstTab__4F7CD00D foreign key (InstTabId)
references dbo.InstTab (InstTabId)
go
```

List of diagrams containing the reference FK__CursTab__InstTab__4F7CD00D

Name	Code
PhysicalDiagram_1	PHYSICALDIAGRAM_1

List of reference joins of the reference FK__HoraTab__InstTab__52593CB8

Parent Table Column	Child Table Column
InstTabId	InstTabId

List of extended attributes of the reference FK__HoraTab__InstTab__52593CB8

Name	Data Type	Value	Target Name
ExtFkNotForReplication	(Boolean)	false	Microsoft SQL Server 2008

Reference FK__InstTab__PaisTab__571DF1D5

Card of reference FK__InstTab__PaisTab__571DF1D5

Name	FK__InstTab__PaisTab__571DF1D5
Code	FK__InstTab__PaisTab__571DF1D5
Parent Table	PaisTab
Child Table	InstTab

Code preview of reference FK__InstTab__PaisTab__571DF1D5

```
if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('dbo.InstTab') and o.name = 'FK__InstTab__PaisTab__571DF1D5')
alter table dbo.InstTab
```

```
drop constraint FK__InstTab__PaisTab__571DF1D5
go

alter table dbo.InstTab

add constraint FK__InstTab__PaisTab__571DF1D5 foreign key (PaisTabId)

references dbo.PaisTab (PaisTabId)
go
```

List of diagrams containing the reference FK__InstTab__PaisTab__571DF1D5

Name	Code
PhysicalDiagram_1	PHYSICALDIAGRAM_1

List of reference joins of the reference FK__InstTab__PaisTab__571DF1D5

Parent Table Column	Child Table Column
PaisTabId	PaisTabId

List of extended attributes of the reference FK__InstTab__PaisTab__571DF1D5

Name	Data Type	Value	Target Name
ExtFkNotForReplication	(Boolean)	false	Microsoft SQL Server 2008

Reference FK__LogUTab__InstTab__59063A47

Card of reference FK__LogUTab__InstTab__59063A47

Name	FK__LogUTab__InstTab__59063A47
Code	FK__LogUTab__InstTab__59063A47
Parent Table	InstTab
Child Table	LogUTab

Code preview of reference FK__LogUTab__InstTab__59063A47

```
if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F'))
```

```

where r.fkeyid = object_id('dbo.LogUTab') and o.name = 'FK__LogUTab__InstTab__59063A47')

alter table dbo.LogUTab

drop constraint FK__LogUTab__InstTab__59063A47

go

alter table dbo.LogUTab

add constraint FK__LogUTab__InstTab__59063A47 foreign key (InstTabId)

references dbo.InstTab (InstTabId)

go

```

List of diagrams containing the reference FK__LogUTab__InstTab__59063A47

Name	Code
PhysicalDiagram_1	PHYSICALDIAGRAM_1

8.05 Manual Técnico

En el Caso de Matriculación

Reglas:

```

/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
    parm(in:&Mode, in:&InstTabId, in:&MatricTabId, in:&SociPrintTabId,
    in:&AulaTabId, in:&HoraTabId, in:&CursTabId, in:&PeriTabId,
    in:&MatricTabFact);

    InstTabId = &InstTabId if not &InstTabId.IsEmpty();
    noaccept(InstTabId) if not &InstTabId.IsEmpty();
    MatricTabId = &MatricTabId if not &MatricTabId.IsEmpty();
    noaccept(MatricTabId) if not &MatricTabId.IsEmpty();
    SociPrintTabId = &SociPrintTabId if not &SociPrintTabId.IsEmpty();
    noaccept(SociPrintTabId) if not &SociPrintTabId.IsEmpty();
    AulaTabId = &AulaTabId if not &AulaTabId.IsEmpty();
    noaccept(AulaTabId) if not &AulaTabId.IsEmpty();
    HoraTabId = &HoraTabId if not &HoraTabId.IsEmpty();
}

```




```
noaccept(HoraTabId) if not &HoraTabId.IsEmpty();
CursTabId = &CursTabId if not &CursTabId.IsEmpty();
noaccept(CursTabId) if not &CursTabId.IsEmpty();
PeriTabId = &PeriTabId if not &PeriTabId.IsEmpty();
noaccept(PeriTabId) if not &PeriTabId.IsEmpty();
MatricTabFact = &MatricTabFact if not &MatricTabFact.IsEmpty();
noaccept(MatricTabFact) if not &MatricTabFact.IsEmpty();
}
/* Generated by Work With Pattern [End] - Do not change */
//Titulo de Formulario
Form.Caption = 'Creación de Alumnos' If Insert;
Form.Caption = 'Modificación de Alumnos' If Update;
Form.Caption = 'Eliminación de Alumnos' If Delete;

//Valores por Default
Default(MatricTabUsr, &LogUTabLogin);
Default(MatricTabFhr, ServerNow());
Default(AsistenTabFec, ServerNow());
Default(AsistenTabFhr, ServerNow());
Default(PagoTabFH, ServerNow());
Default(PagoTabFhr, ServerNow());

ContMatricula.Call(MatricTabId) If Insert On AfterInsert;

Prompt(SelecAula, &InstTabId, AulaTabId, &AulaTabDescrip);
Error('') If &Flag = True On AfterInsert;
```

Eventos:

```
Event Start
  for each
    Form.Caption = DmRecDes.ToString()
  endfor
EndEvent
```

Eventos:

```
Event Start
  Autorizacion.Call(&Autorizacion)
  If (not &Autorizacion)
    DeniedAcces.Call()
  Endif

  &LogUTabLogin = getCookie('LogUTabLogin')

  /* Generated by Work With Pattern [Start] - Do not change */
  [web]
  {
    // ERROR: Security check is set to true in Work With Settings,
    but either the 'Check' procedure or the 'Not Authorized' webpanel
    has not been set.

    &TrnContext.FromXml(&WebSession.Get(!"TrnContext"))
  }
  /* Generated by Work With Pattern [End] - Do not change */
```



```
EndEvent

Event After Trn
Return
/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
If (&Mode = TrnMode.Delete and not &TrnContext.CallerOnDelete)
WWMatricTab.Call()
Endif

Return
}
/* Generated by Work With Pattern [End] - Do not change */
EndEvent
```

Procedimiento de validación de la cedula

```
&Largo= Len(Trim(&CedRuc))
&CodErr= 0

//for Each DmCliIdF
//    where DmCliIdF = &CedRuc
//    &DmCliTipId = DmCliTipId
//endfor

Do Case
    Case &DmCliTipId = 'R'
        &TipDoc= 1
    Case &DmCliTipId = 'C'
        &TipDoc= 2
    case &DmCliTipId = 'P'
        &TipDoc= 9
EndCase

If &TipDoc = 1 Or &TipDoc = 2
    // ----- Valida Provincia
    Digits 1 y 2
    &AuxStr= SubStr(&CedRuc, 1, 2)
    &AuxInt= Val(&AuxStr)
    If &AuxInt < 1 Or &AuxInt > 22
        &CodErr= 11
        &DesErr= 'Provincia Incorrecta..'
    EndIf

    // ----- Valida Tercer
    Dígito
    &AuxStr= SubStr(&CedRuc, 3, 1)
    &AuxInt= Val(&AuxStr)
    If &AuxInt = 7 Or &AuxInt = 8
        &CodErr= 12
        &DesErr= 'Tercer Dígito Inválido..'
    EndIf

    // ----- Valida 3 últimos
    dígitos
```



```
If &TipDoc = 1
    &AuxStr= SubStr(&CedRuc, 11, 3)
    If &AuxStr <> '001'
        &CodErr= 13
        &DesErr= 'Tres últimos dígitos deben ser 001'
    EndIf
EndIf

//----- Dígito Verificador
//----- Cédulas
&Coeficiente= 2
&Acum= 0

For &Cnt = 1 To 9
    &AuxStr= SubStr(&CedRuc, &Cnt, 1)
    &AuxInt= Val(&AuxStr)

    &AuxInt= &AuxInt* &Coeficiente

    If &AuxInt > 9
        &AuxStr= LTrim(Str(&AuxInt))
        &AuxInt= 0
        For &Cnt1 = 1 To Len(&AuxStr)
            &AuxInt= &AuxInt+ Val(SubStr(&AuxStr, &Cnt1, 1))
        EndFor
    EndIf

    &Acum= &Acum+ &AuxInt

    If &Coeficiente = 1
        &Coeficiente= 2
    Else
        &Coeficiente= 1
    EndIf
EndFor

&AuxInt= Int(&Acum/ 10)
&Acum= &Acum- &AuxInt* 10
If &Acum <> 0
    &Acum= 10- &Acum
EndIf

&AuxInt= Val(SubStr(&CedRuc, 10, 1))
If &Acum <> &AuxInt
    &CodErr= 99
    &DesErr= 'Dígito Verificador Incorrecto...'
Else
    If &Largo = 10
        &CodErr= 1
        &DesErr= 'Cédula de Identidad Válida...'
    Else
        &CodErr= 2
        &DesErr= 'RUC Personal Válido... (Cédula + 001)'
    EndIf
EndIf

//----- RUC
If &CodErr = 99
```



```
&AuxInt= Val(SubStr(&CedRuc, 3, 1))

    If &AuxInt = 9                                // Personas Juridicas y
Extranjeros sin Cedula
    &Coeficiente= 4
    &Acum= 0

    For &Cnt = 1 To 9
        &AuxStr= SubStr(&CedRuc, &Cnt, 1)
        &AuxInt= Val(&AuxStr)

        &AuxInt= &AuxInt* &Coeficiente
        &Acum= &Acum+ &AuxInt

        &Coeficiente= &Coeficiente- 1
        If &Coeficiente = 1
            &Coeficiente= 7
        EndIf
    EndFor

    &AuxInt= Int(&Acum/ 11)
    &Acum= &Acum- &AuxInt* 11
    If &Acum <> 0
        &Acum= 11- &Acum
    EndIf

    &AuxInt= Val(SubStr(&CedRuc, 10, 1))
    If &Acum <> &AuxInt
        &CodErr= 99
        &DesErr= 'Dígito Verificador Incorrecto...'
        Return
    Else
        &CodErr= 3
        &DesErr= 'RUC Personas Jurídicas Válido.'
    EndIf

Else
    &Coeficiente= 3
    &Acum= 0

    For &Cnt = 1 To 8
        &AuxStr= SubStr(&CedRuc, &Cnt, 1)
        &AuxInt= Val(&AuxStr)

        &AuxInt= &AuxInt* &Coeficiente
        &Acum= &Acum+ &AuxInt

        &Coeficiente= &Coeficiente- 1
        If &Coeficiente = 1
            &Coeficiente= 7
        EndIf
    EndFor

    &AuxInt= Int(&Acum/ 11)
    &Acum= &Acum- &AuxInt* 11
    If &Acum <> 0
        &Acum= 11- &Acum
    EndIf
```



```

        &AuxInt= Val(SubStr(&CedRuc, 9, 1))
    If &Acum <> &AuxInt
        &CodErr= 99
        &DesErr= 'Dígito Verificador Incorrecto...'
    Else
        &CodErr= 4
        &DesErr= 'RUC Empresas Públicas Válido.'
    EndIf
EndIf
EndIf
EndIf

```

En el caso de Clientes y Empleados

EVENTOS

```

Event Start
    Autorizacion.Call(&Autorizacion)
    If (not &Autorizacion)
        DeniedAcces.Call()
    Endif

    &InstTabId = Val(Trim(GetCookie('InstTabId'))))
    &LogUTabLogin = getCookie('LogUTabLogin')

    /* Generated by Work With Pattern [Start] - Do not change */
    [web]
    {
        // ERROR: Security check is set to true in Work With Settings,
        but either the 'Check' procedure or the 'Not Authorized' webpanel
        has not been set.

        &TrnContext.FromXml(&WebSession.Get(!"TrnContext"))
        &Insert_CiudTabId.SetEmpty()
        &Insert_ProvTabId.SetEmpty()
        &Insert_PaisTabId.SetEmpty()

        If (&TrnContext.TransactionName = &Pgmname and &Mode =
TrnMode.Insert)
            For &TrnContextAtt in &TrnContext.Attributes
                Do Case
                    // When inserting with instantiated
CiudTabId
                    Case &TrnContextAtt.AttributeName =
!"CiudTabId"

                        &Insert_CiudTabId.FromString(&TrnContextAtt.AttributeValue)
                    // When inserting with instantiated
ProvTabId
                    Case &TrnContextAtt.AttributeName =
!"ProvTabId"

                        &Insert_ProvTabId.FromString(&TrnContextAtt.AttributeValue)
                    // When inserting with instantiated
PaisTabId
                    Case &TrnContextAtt.AttributeName =
!"PaisTabId"

```



```
        &Insert_PaisTabId.FromString(&TrnContextAtt.AttributeValue)
        Endcase
    Endfor
Endif
}
/* Generated by Work With Pattern [End] - Do not change */
EndEvent

Event After Trn
Return
/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
    If (&Mode = TrnMode.Delete and not &TrnContext.CallerOnDelete)
        WWClieTab.Call()
    Endif

    Return
}
/* Generated by Work With Pattern [End] - Do not change */
EndEvent
```

REGLAS

```
/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
    parm(in:&Mode, in:&SociPrintTabId);

    SociPrintTabId = &SociPrintTabId if not &SociPrintTabId.IsEmpty();
    noaccept(SociPrintTabId) if not &SociPrintTabId.IsEmpty();
    noprompt(SociPrintTabId);

    CiudTabId = &Insert_CiudTabId if &Mode = TrnMode.Insert and not
    &Insert_CiudTabId.IsEmpty();
    noaccept(CiudTabId) if &Mode = TrnMode.Insert and not
    &Insert_CiudTabId.IsEmpty();
    ProvTabId = &Insert_ProvTabId if &Mode = TrnMode.Insert and not
    &Insert_ProvTabId.IsEmpty();
    noaccept(ProvTabId) if &Mode = TrnMode.Insert and not
    &Insert_ProvTabId.IsEmpty();
    PaisTabId = &Insert_PaisTabId if &Mode = TrnMode.Insert and not
    &Insert_PaisTabId.IsEmpty();
    noaccept(PaisTabId) if &Mode = TrnMode.Insert and not
    &Insert_PaisTabId.IsEmpty();}
/* Generated by Work With Pattern [End] - Do not change */
//Titulo de Formulario
Form.Caption = 'Creación de Cliente' If Insert;
Form.Caption = 'Modificación de Cliente' If Update;
Form.Caption = 'Eliminación de Cliente' If Delete;

//Valores por Default
Default(SociPrintTabUsr, &LogUTabLogin);
Default(SociPrintTabFh, ServerNow());
Default(SociPrintTabTipo, 'E');
```

```
//Cargar Variables
&SDTPAIS      = DPPAIS(PaisTabId);
&PaisTabNom   = &SDTPAIS.PaisTabNom;
&ProvTabNom   = &SDTPAIS.ProvTab.ProvTabNom;
&SDTCIUDAD    = DPCIUDAD(CiudTabId, ProvTabId, PaisTabId);
&CiudTabNom   = &SDTCIUDAD.CiudTabNom;

SecSoci.Call(SociPrintTabId) If Insert on AfterValidate;
Prompt(SelecCiudad, PaisTabId, ProvTabId, CiudTabId, &CiudTabNom,
&ProvTabNom, &PaisTabNom);

VerifIden.Call(SociPrintTabIden, 1, SociPrintTabTipoIden , &CodErr,
&DesErr) If Insert;
VerifIden.Call(SociPrintTabIden, 2, SociPrintTabTipoIden , &CodErr,
&DesErr) If Update;
Error("cedula Incorrecta") if &CodErr<>1 On AfterInsert;
SociPrintTabIden = "" if &CodErr<>1 on AfterInsert;

Error("cedula Incorrecta") if &CodErr<>1 On AfterUpdate;
SociPrintTabIden = "" if &CodErr<>1 On AfterUpdate;

Error("Identificacion de Socio No puede ser Nulo") If
SociPrintTabIden.IsEmpty();
Error("Razon Social No puede ser Nulo") If
SociPrintTabRazS.IsEmpty();
Error("Direccion no puede ser Nulo")If SociPrintTabDir.IsEmpty();

//Auditoria de Eliminacion de Cliente
&SociPrintTabRazS = Trim(SociPrintTabRazS);
TabDel.Call(&InstTabId, 'Clientes', 'Nombre', &SociPrintTabRazS) if
delete on afterdelete;

//Auditoria de modificacion
//Modificacion de nombre
&SociPrintTabRazSAnt = Trim(old(SociPrintTabRazS)) If Update on
AfterValidate;
&SociPrintTabRazSNuev = Trim(SociPrintTabRazS) If Update on
AfterValidate;
TabMod.Call(&InstTabId, 'Clientes', 'Nombre', &SociPrintTabRazSAnt,
&SociPrintTabRazSNuev ) if update and &SociPrintTabRazSAnt <>
&SociPrintTabRazSNuev on AfterValidate;

//Modificacion de Identificacion de Cliente
&SociPrintTabIdenAnt = Trim(old(SociPrintTabIden)) If Update on
AfterValidate;
&SociPrintTabIdenNuev = Trim(SociPrintTabIden) If Update on
AfterValidate;
TabMod.Call(&InstTabId, 'Clientes', 'Identificación',
&SociPrintTabIdenAnt, &SociPrintTabIdenNuev ) if update and
&SociPrintTabIdenAnt <> &SociPrintTabIdenNuev on AfterValidate;

//Modificacion de direccion de cliente
&SociPrintTabDirAnt = Trim(old(SociPrintTabDir)) If Update on
AfterValidate;
&SociPrintTabDirNuev = Trim(SociPrintTabDir) If Update on
AfterValidate;
```

```

TabMod.Call(&InstTabId,'Clientes','Dirección', &SociPrintTabDirAnt,
&SociPrintTabDirNuev ) if update and &SociPrintTabDirAnt <>
&SociPrintTabDirNuev on AfterValidate;

//Modificacion telefono cliente
&SociPrintTabTelf1Ant = Trim(old(SociPrintTabTelf1)) If Update on
AfterValidate;
&SociPrintTabTelf1Nuev = Trim(SociPrintTabTelf1) If Update on
AfterValidate;
TabMod.Call(&InstTabId,'Clientes','Telefono1',
&SociPrintTabTelf1Ant, &SociPrintTabTelf1Nuev ) if update and
&SociPrintTabTelf1Ant <> &SociPrintTabTelf1Nuev on AfterValidate;

&SociPrintTabTipoAnt = Trim(old(SociPrintTabTipo)) If Update on
AfterValidate;
&SociPrintTabTipoNew = Trim(SociPrintTabTipo) If Update on
AfterValidate;
TabMod.Call(&InstTabId,'Clientes','Tipo de Cliente',
&SociPrintTabTipoAnt, &SociPrintTabTipoNew ) if update and
&SociPrintTabTipoAnt <> &SociPrintTabTipoNew on AfterValidate;

&SociPrintTabTelf2Ant =Trim(old(SociPrintTabTelf2)) If Update on
AfterValidate;
&SociPrintTabTelf2New =Trim(SociPrintTabTelf2) If Update on
AfterValidate;
TabMod.Call(&InstTabId,'Clientes','Telefono2',
&SociPrintTabTelf2Ant, &SociPrintTabTelf2New ) if update and
&SociPrintTabTelf2Ant <> &SociPrintTabTelf2New on AfterValidate;

```

8.06 Manual de Usuario

Figura 33 Ingreso General de acceso del sistema

Figura: 33 Pantalla de la aplicación



Una vez que el usuario se autentifique podrá tener acceso al menú que se muestra en la imagen.

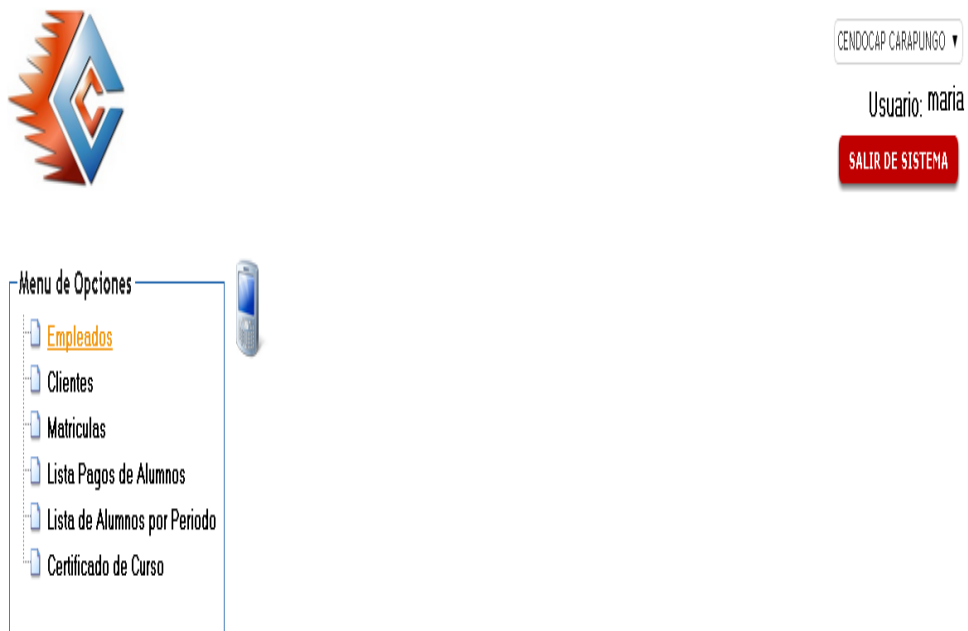
Aquí podrá dar clic para logearse.

Figura 34 Menú de Login de Usuario



Menú de login de usuario ingresa el usuario y la clave e ingresa al sistema.

Figura: 35 Menú de Opciones del Usuario



En el menú de usuario tenemos todas las opciones del trabajo que desempeña la secretaria.


Figura: 36 Registros de Empleados


Listado de Empleados

Busqueda

Cedula/Ruc:

Nombre:

 **NUEVO**

CIRUC	NOMBRE	DIRECCIÓN	TELÉFONO	TIPO DE PERSONA
 0602527756	MARIA DEL PILAR ZAMBRANO VARGAS	CARAPUNGO 3RA ETAPA EN LA IGLESIA	2421982	Profesor

Nos muestra los empleados ingresados

Figura: 37 Registro de Nuevo Empleado

Creación de Empleados

Maestro de empleados

Razon Social

Tipo Identificación Cedula ▼

Identificación

Tipo de Persona Profesor


Dirección

Teléfono1

Teléfono 2

Email

Estado ACTIVO ▼

Código de Ciudad 


Código de Provincia

Código de país

Confirmar
Cancelar

Se ingresa el registro de nuevo Empleado

Figura: 38 Modificar empleado



Maestro de empleados

Razon Social: MARIA DEL PILAR ZAMBRANO VARGAS

Tipo Identificacion: Cedula

Identificacion: 0602527756

Tipo de Persona: Profesor

Direccion: CARAPUNGO 3RA ETAPA EN LA IGLESIA

Telefono1: 2421982

Telefono 2: 8884-243-53

Email:


Estado: ACTIVO

Codigo de Ciudad: 1 QUITO

Codigo de Provincia: 1 PICHINCHA

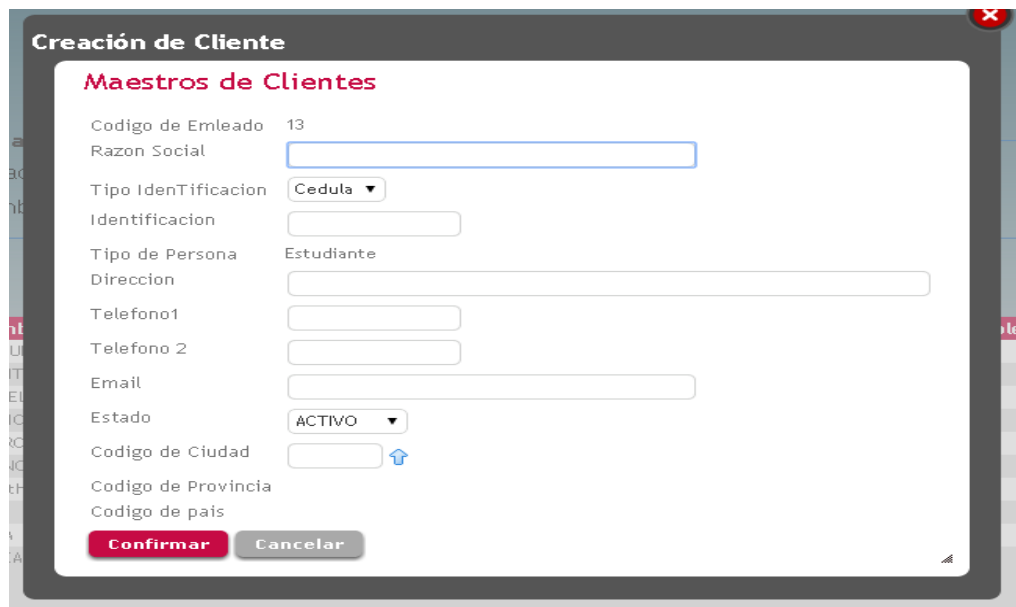
Codigo de pais: 1 ECUADOR

Confirmar **Cancelar**

Si se desea modificar la información solo se debe dar Tap sobre la imagen y seguidamente nos aparecerá la siguiente imagen en la cual se podrá modificar los datos. 

si los datos que se modificaron son correctos, dar Tap sobre la imagen y seguidamente los datos se modificaran en la base de datos. **CONFIRMAR**

Figura: 39 Registro de Nuevo Empleado



Creación de Cliente

Maestros de Clientes

Codigo de Empleado: 13

Razon Social:

Tipo Identificacion: Cedula

Identificacion:

Tipo de Persona: Estudiante

Direccion:

Telefono1:

Telefono 2:

Email:

Estado: ACTIVO

Codigo de Ciudad:

Codigo de Provincia:

Codigo de pais:

Confirmar **Cancelar**

Se ingresa el registro de nuevo Cliente

Figura:40 Modificar Cliente




Si se desea modificar la información solo se debe dar Tap sobre la imagen y seguidamente nos aparecerá la siguiente imagen en la cual se podrá modificar los datos.  si los datos que se modificaron son correctos, dar Tap sobre la imagen y seguidamente los datos se modificaran en la base de datos. **CONFIRMAR**

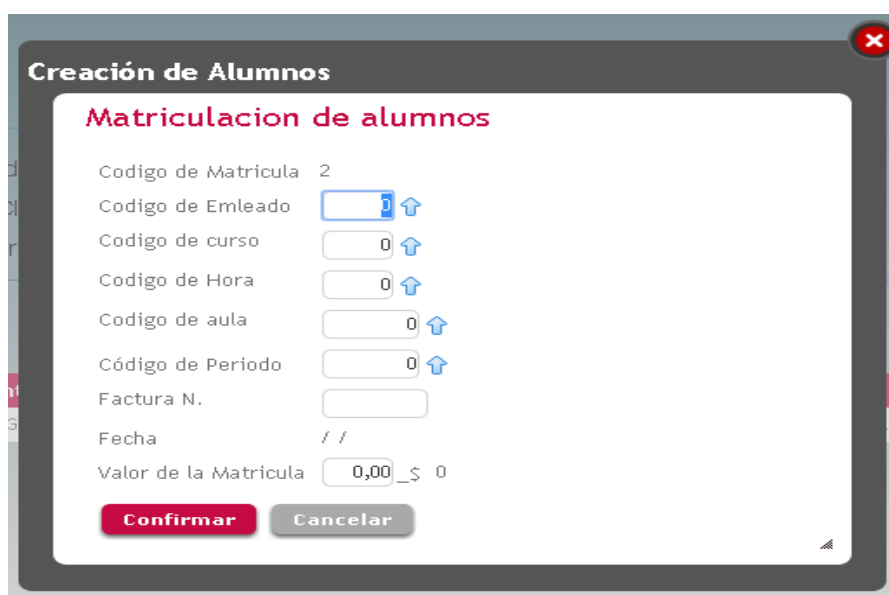
Figura: 41 Matricula



Identificación	Nombre de Cliente	Aula	Curso	Jornada	Periodo
  YDFG5JCG5H34	MIGUEL AGLONY	AULA DE CENESYT	cencyt	Noche	FEBRERO-MARZO

Datos de los alumnos que se matricularon

Figura: 42 Registro de Nueva Matricula



Creación de Alumnos

Matriculación de alumnos

Código de Matricula 2

Código de Empleado ↑

Código de curso ↑

Código de Hora ↑

Código de aula ↑

Código de Periodo ↑

Factura N.

Fecha / /

Valor de la Matricula _\$ 0

Confirmar **Cancelar**

Se ingresa el registro de nuevo Cliente

Figura: 43 Modificar de Matricula



Modificación de Alumnos

Matriculación de alumnos

Código de Matricula 1

Código de Empleado 3 MIGUEL AGLONY

Código de curso 3 cencyt

Código de Hora 2 N

Código de aula 3 AULA DE CENESYT

Código de Periodo 2 FEBRERO-MARZO

Factura N. 123

Fecha / /

Valor de la Matricula _\$ 15

Confirmar **Cancelar**


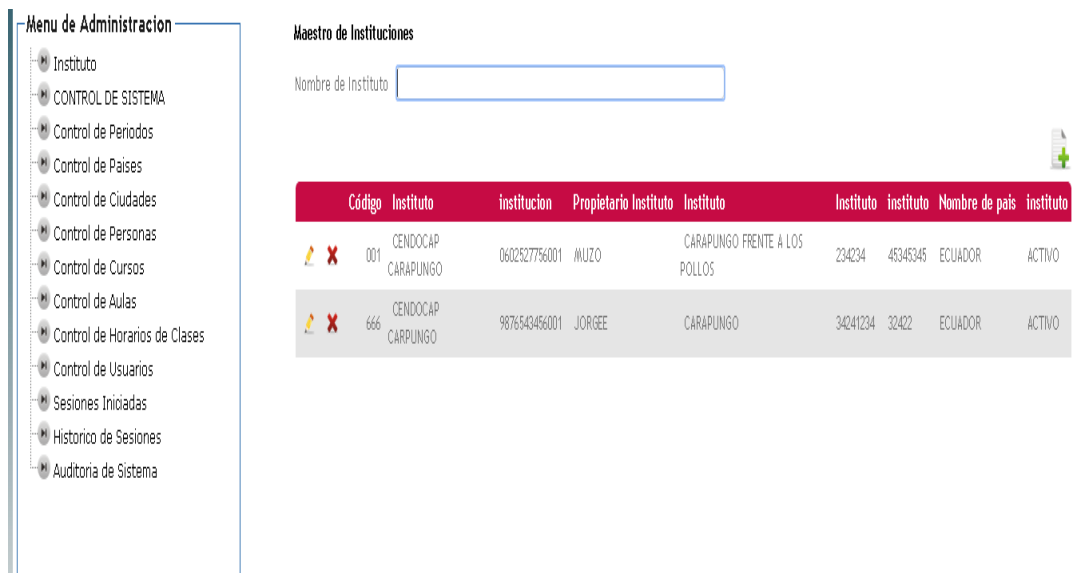
Si se desea modificar la información solo se debe dar Tap sobre la imagen y seguidamente nos aparecerá la siguiente imagen en la cual se podrá modificar los datos.  si los datos que se modificaron son correctos, dar Tap sobre la imagen y seguidamente los datos se modificaran en la base de datos. **CONFIRMAR**

Figura:44 Menú de Opciones del Administrador



En el menú de usuario tenemos todas las opciones del trabajo que desempeña la secretaria..

Figura: 45 Maestro de Institutos.



Esta pantalla nos muestra los institutos ingresados o registrados.

Figura: 46 Nuevo instituto



Registro de Instituciones

Creación de Instituciones

Maestro de Instituciones

Nombre:

Ruc:

Propietario:

Direccion:

Telefono1: **Por Favor Ingrese Telefn 1**

Telefono2:

Logo: Ningún archivo cargado

Pais:

Estado:

Esta pantalla nos permite ingresar un nuevo instituto.

Figura: 47 Periodos de clases



Menu de Administracion

- Instituto
- CONTROL DE SISTEMA
- Control de Periodos
- Control de Paises
- Control de Ciudades
- Control de Personas
- Control de Cursos
- Control de Aulas
- Control de Horarios de Clases
- Control de Usuarios
- Sesiones Iniciadas
- Historico de Sesiones
- Auditoria de Sistema

Periodo de Clases

Descripción de Periodo:

Fecha Inicial:

Fecha Final:

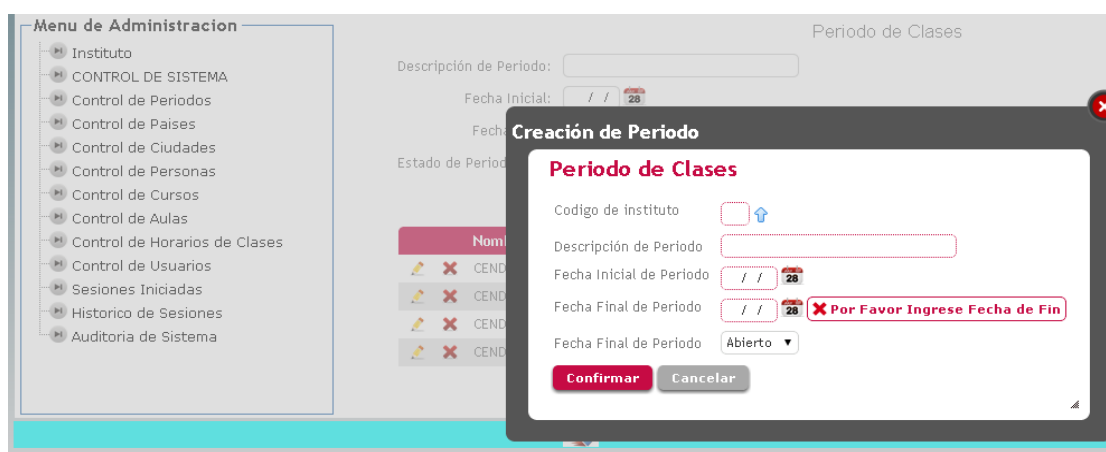
Estado de Periodo:

NombreInstituto	Código de Periodo	Descripción de Periodo	Fecha Inicial de Periodo	Fecha Final de Periodo	de Periodo
CENDOCAP CARPUNGO	1	abril-mayo	03/02/14	01/01/14	Abierto
CENDOCAP CARPUNGO	3	cenecyt	02/03/14	31/03/14	Abierto
CENDOCAP CARAPUNGO	2	FEBRERO-MARZO	27/02/14	27/03/14	Abierto
CENDOCAP CARAPUNGO	34	MARZO-ABRIL	02/03/14	31/03/14	Abierto

Copyright 2013 - 2014 Powered by Cendocap - Centro de alta Capacitación

Esta pantalla nos muestra los periodos de clases registrados en el Sistema.

Figura: 48 Creación de periodo de clases



Menu de Administracion

- Instituto
- CONTROL DE SISTEMA
- Control de Periodos
- Control de Paises
- Control de Ciudades
- Control de Personas
- Control de Cursos
- Control de Aulas
- Control de Horarios de Clases
- Control de Usuarios
- Sesiones Iniciadas
- Historico de Sesiones
- Auditoria de Sistema

Periodo de Clases

Descripción de Periodo:

Fecha Inicial:

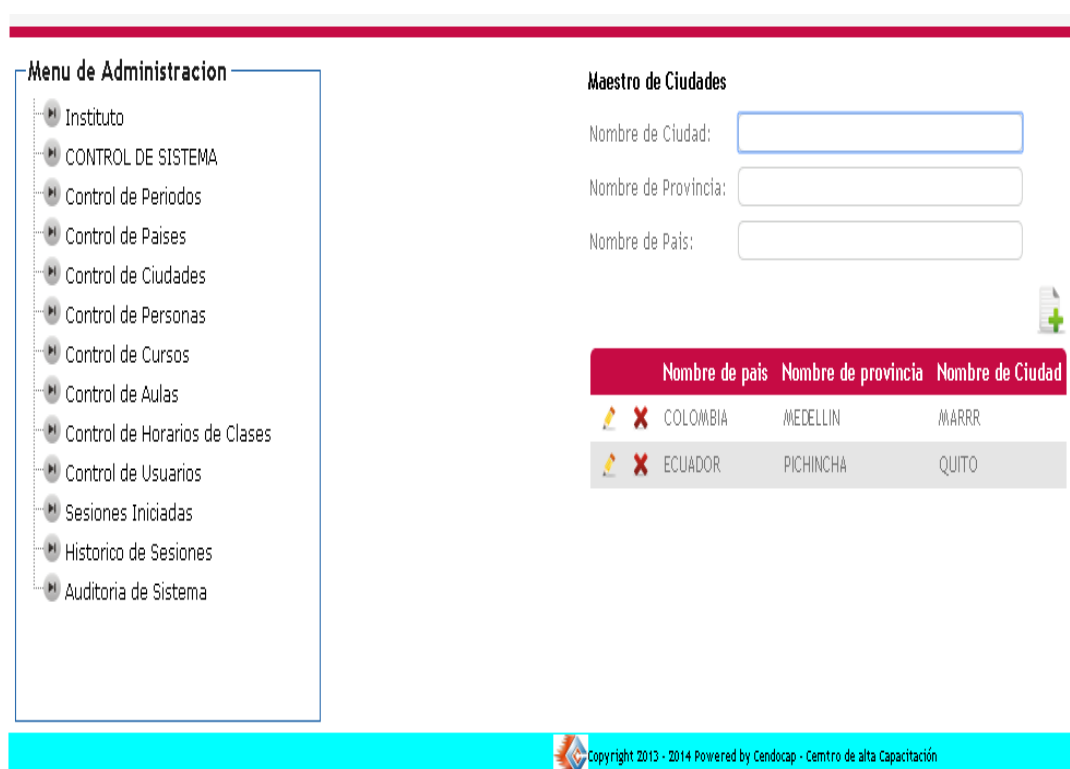
Fecha Final de Periodo: **Por Favor Ingrese Fecha de Fin**

Fecha Final de Periodo:

Confirmar **Cancelar**

Esta pantalla nos permite crear los periodos de clase.

Figura: 49 Maestro de Ciudades



Menu de Administracion

- Instituto
- CONTROL DE SISTEMA
- Control de Periodos
- Control de Paises
- Control de Ciudades
- Control de Personas
- Control de Cursos
- Control de Aulas
- Control de Horarios de Clases
- Control de Usuarios
- Sesiones Iniciadas
- Historico de Sesiones
- Auditoria de Sistema

Maestro de Ciudades

Nombre de Ciudad:

Nombre de Provincia:

Nombre de Pais:

Nombre de pais **Nombre de provincia** **Nombre de Ciudad**

COLOMBIA	MEDELLIN	MARRR
ECUADOR	PICHINCHA	QUITO

Copyright 2013 - 2014 Powered by Cendocap - Centro de alta Capacitación

Esta pantalla nos permite visualizar las ciudades ingresadas en el sistema.

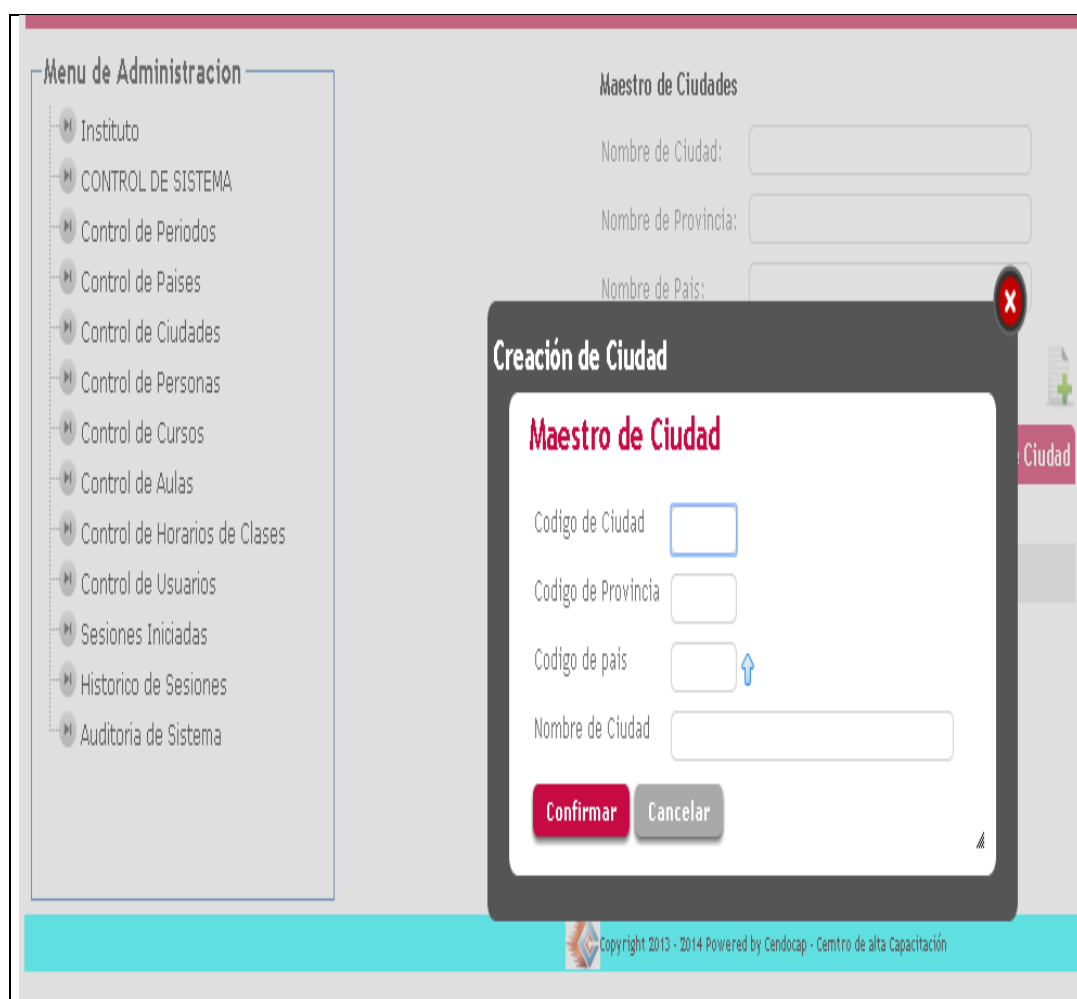


Figura: 50 Registro de Ciudades

Esta pantalla nos permite crear las ciudades.

8.07 GLOSARIO

Actor.- Algo o alguien externo al sistema en desarrollo pero que interactúa con él.

Arquitectura.- Estructura lógica y física de un sistema empleado para diseñar todas las estrategias y tácticas aplicadas durante el desarrollo.

Atributo.- Definición de dato simple o compuesto perteneciente a un objeto de clase.

Clase.- Descripción de un grupo de objetos con atributos, conducta y relaciones comunes.

Caso de uso.- Representación de un proceso del negocio. Representa el modelo de diálogo entre un actor y el sistema

Diagrama de casos de uso.- Representación gráfica que representa algunos o todos los actores, casos de uso y sus interacciones en el sistema.

Diagrama de clases.- Representación gráfica que permite visualizar algunas o todas las clases de un modelo

Diagrama de secuencias.- Representación gráfica que describe interacciones de secuencia de objetos.

Diagrama de iteración.- Representación gráfica de un proceso aplicado en el desarrollo de la plataforma virtual

IEEE.- Corresponde a las siglas de the institute of electrical and electronics engineers, el instituto de ingenieros eléctricos y electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, ingenieros en sistemas e ingenieros en telecomunicación....

UML (lenguaje de modelamiento unificado).- Lenguaje usado para especificar, visualizar y documentar un sistema en desarrollo orientado a objetos.

Plataforma virtual.- Las plataformas virtuales, se refieren, a la tecnología utilizada para la creación y desarrollo de cursos o módulos didácticos en la Web que se usan de manera más amplia en la Web 2.0. Mejora de la comunicación aprendizaje-enseñanza.

Base de datos.-Estructura de software que colecciona información muy variada de diferentes personas y cosas (es decir, de una realidad determinada), cada una de las cuales tiene algo en común o campos comunes con todos o con algunos. Se diseñó con la finalidad de solucionar y agilizar la administración de los datos que se almacenan en la memoria del computador.

Hardware.- Todos aquellos componentes físicos de un computador, todo lo visible y tangible. Por extensión, se aplica también a otros componentes electrónicos que no necesariamente forman parte de un computador.

Home pages.- En el web se refiere a las páginas de inicio que enlazan con otras páginas relacionadas.

Informática.- Ciencia del tratamiento automático y racional de la información, considerada como soporte de los conocimientos y comunicaciones, a través de los ordenadores.

Internet.-Proyecto que ya está en marcha para mejorar internet que se trata de la posibilidad de navegar en la red a una velocidad de 622 megabits por segundo, más de 1000 veces la velocidad actual disponible.

JavaScript.- Un lenguaje de comandos multiplataforma del WWW desarrollado por Netscape Communications. El código de JavaScript se inserta directamente en una página HTML.

8.08 WEBGRAFÍA

<http://www.desarrolloweb.com/articulos/jquery4php.html>

<http://www.youtube.com/watch?v=-RdUv3OamvE>

<http://mx.answers.yahoo.com/question/index?qid=20090420182420AAy0mgR>

<http://www.softonic.com/s/descargar-css>

<http://www.freecsstemplates.org/>

<http://getbootstrap.com/2.3.2/javascript.html#tabs>