



INSTITUTO TECNOLÓGICO  
“CORDILLERA”

CARRERA ANÁLISIS DE SISTEMAS

”AUTOMATIZACIÓN DE LOS PROCESOS ACADÉMICOS DE  
INSTITUCIONES EDUCATIVAS DE NIVEL MEDIO – BACHILLERATO  
MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN EDUCATIVA:  
MÓDULO GESTIÓN ACADÉMICA.”

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

Autor: Cevallos Padilla Carlos Andres

Tutor: Ing. Hugo Heredia M., Msc.

Quito, Abril 2015

---

### **Declaración De Aprobación Tutor Y Lector**

En mi calidad de tutor del trabajo sobre el tema: **“AUTOMATIZACIÓN DE LOS PROCESOS ACADÉMICOS DE INSTITUCIONES EDUCATIVAS DE NIVEL MEDIO – BACHILLERATO MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN EDUCATIVA MÓDULO GESTIÓN ACADÉMICA”**, presentado por el ciudadano: Cevallos Padilla Carlos Andres, estudiante de la Escuela de Análisis de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril del 2014

---

**Ing. Hugo Heredia**

TUTOR

---

**Ing. Diana Terán**

LECTOR

---

## **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 CEVALLOS PADILLA CARLOS ANDRES, 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 “AUTOMATIZACIÓN DE LOS PROCESOS ACADÉMICOS DE INSTITUCIONES EDUCATIVAS DE NIVEL MEDIO – BACHILLERATO MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN EDUCATIVA MÓDULO GESTIÓN ACADÉMICA”, 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 reconvenición, 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 6 días del mes de Abril del dos mil quince.

---

C.C. 1723002059

CEDENTE

---

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERAR

CESIONARIO

---

### Declaración De Autoría Del Estudiante

Declaro que la investigación los contenidos y los resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del Título de **Tecnólogo en Análisis de Sistemas** es netamente original, personal, y que se han citado las fuentes correspondientes.

---

Cevallos Padilla Carlos Andres

C.C. 172300205-9

## **Agradecimiento**

Agradezco a mis Padres, por darme su amor, por esos abrazos de consuelo,  
por celebrar con migo mis  
triunfos, la oportunidad de existir,  
reconocimientos a ellos que son  
parte de esta etapa de mi vida.

### **Dedicatoria**

A mis Padres por su apoyo  
incondicional y son los principales  
gestores de este Proyecto.

Cevallos Padilla Carlos Andres

---



---

## Índice General

Contenido	Pág.
Declaración de aprobación de Tutor y Lector.....	ii
Contrato de Cesión sobre derechos propiedad Intelectual.....	iii
Declaración de autoría del estudiante.....	vi
Agradecimiento .....	vii
Dedicatoria.....	viii
Resumen Ejecutivo .....	xv
Abstract.....	xvi
Capítulo I: Antecedentes .....	1
1.01 Contexto .....	1
1.02 Justificación .....	2
1.03 Definición de problema central .....	3
Capítulo II: Análisis de Involucrados .....	7
2.01 Requerimientos.....	7
2.01.1 Descripción del sistema actual.....	7
2.01.2 Visión y Alcance.....	7
2.01.3 Entrevista.....	8
2.01.4 Matriz de requerimiento.....	9
2.01.5 Descripción detallada.....	10
2.02 Mapeo de involucrados .....	13
2.02.01 Análisis del Mapeo de involucrado.....	14

---

	Pág.
2.03 Matriz de análisis de involucrados.....	15
Capítulo III: Problemas y Objetivos.....	16
3.01 Árbol de Problemas.....	16
3.02 Árbol de Objetivos.....	17
3.03 Diagrama casos de uso.....	18
3.04 Casos de uso de realización.....	19
3.05 Diagrama de Secuencia del Sistema.....	23
Capítulo IV: Análisis de Alternativas. ....	24
4.01 Matriz de Análisis de Alternativas.....	24
4.02 Matriz de Impacto de los Objetivos.....	25
4.03 Estándares para el Diseño de Clases.....	26
4.04 Diagrama de Clases.....	26
4.05 Modelo Lógico – Físico.....	26
4.06 Diagrama de Componentes.....	27
4.07 Diagrama de Estrategias.....	28
4.08 Matriz de Marco lógico.....	29
4.09 Vistas Arquitectónicas.....	30
4.09.01 Vista Física.....	30
4.09.02 Vista Logica.....	31
Capítulo V: Propuesta.....	32
5.01 Especificación de Estándares de Programación.....	34

---

---

	Pág.
5.01.1 Declaración de Variables.....	33
5.02 Diseño de Interfaces de Usuarios.....	34
5.03 Especificación de Pruebas de Unidad.....	39
5.04 Especificación de Pruebas de Aceptación.....	41
5.05 Especificación de Pruebas de Carga.....	43
5.06 Configuración de Ambiente Mínima/Ideal.....	45
Capítulo VI: Aspectos Administrativos.....	47
6.01 Recursos.....	47
6.02 Presupuesto .....	48
6.03 Cronograma .....	49
Capítulo VII: Conclusiones y Recomendaciones .....	50
7.01 Conclusión.....	50
7.02 Recomendación .....	50

---

	Pág.
ANEXOS .....	49
A.01 Matriz de Involucrados.....	49
A.02 Matriz de Alternativas.....	50
A.03 Estándares para el Diseño de clase.....	51
A.04 Diagrama de Clases.....	52
A.05 Modelo Lógico – físico.....	53
A.06 Cronograma.....	54
A.07 Script de la Base de Datos .....	55
A.08 Manual de Instalación .....	73
A.09 Manual de Usuario .....	92
A.10 Manual Técnico.....	95
WEBGRAFÍA .....	106

---

---

## ÍNDICE TABLAS

	Pág.
Tabla 1: Matriz T.....	3
Tabla 2: Entrevista.....	8
Tabla 3: Matriz de Requerimientos.....	9
Tabla 4: Detalle de Requerimientos Mejorar Procesos.....	10
Tabla 5: Detalle de Requerimientos Registro de Notas.....	11
Tabla 6: Detalle de Requerimientos Entrega de Aprobaciones.....	12
Tabla 7: Especificación de caso de uso Realización Quimestres.....	19
Tabla 8: Especificación de caso de uso Quimestres.....	20
Tabla 9: Especificación de caso de uso Realización Ingreso.....	21
Tabla 10: Especificación de caso de uso Ingreso.....	21
Tabla 11: Especificación de caso de uso Realización Recepción.....	22
Tabla 12: Especificación de caso de uso Recepción.....	23
Tabla 13: Matriz de análisis de impacto de los objetivos.....	25
Tabla 14: Descripción de la Variable.....	32
Tabla 15: Descripción Tipos de Variable.....	32
Tabla 16 Descripción Tipos de Control.....	33
Tabla 17: Descripción Declaración de Clases .....	33
Tabla 18: Descripción de Pruebas de Unidad Interface PRU_UNI01.....	39
Tabla 19: Descripción de Pruebas de Unidad Reportes PRU_UNI02.....	40
Tabla 20: Descripción de Pruebas de Unidad Pruebas Código PRU_UNI03.....	40
Tabla 21: Descripción de Pruebas de Unidad almacenamientoPRU_UNI04.....	41
Tabla 22: Detalle Prueba de Aceptación PRU_ACE01 Usuarios.....	42

---

---

Tabla 23: Detalle Prueba de Aceptación PRU_ACE02 Notas.....	42.
Tabla 24: Detalle Tipo de Prueba de Carga más Baja.....	43
Tabla 25: Detalle Tipo de Prueba de con Mínimo de Carga.....	43
Tabla 26: Detalle Tipo de Prueba de Carga Máxima.....	44
Tabla 27: Detalle de Gastos Realizados en el Proyecto.....	48

## ÍNDICE DE FIGURAS

	Pág.
Figura 1: Mapeo de involucrado.....	13
Figura 2: Árbol de Problemas.....	16
Figura 3: Árbol de Objetivos.....	17
Figura 4: Diagrama Casos de Uso Comportamiento del sistema.....	18
Figura 5: Diagrama de Realización-CUR01 Interfaz quimestres.....	19
Figura 6: Diagrama de Realización-CU02 Interfaz ingreso.....	20
Figura 7: Diagrama de Realización-CU03 Interfaz Recepción.....	22
Figura 8: Diagrama de Secuencia Interacción de registro de notas.....	23
Figura 9: Diagrama de Componentes .....	27
Figura 10: Diagrama de Estrategias .....	28
Figura 11: Diagrama Vista Física.....	30
Figura 12: Vista Física Diagrama de Despliegue.....	31
Figura 13: Vista Logica.....	31
Figura 14: Diseño General de Interfaz de usuario.....	34
Figura 15: Diseño General de Ingreso de usuario.....	35
Figura 16: Diseño General de Ingreso de Notas del Docente.....	36
Figura 17: Diseño General de Ingreso de Asistencia alumnos.....	36
Figura 18: Diseño Listado de Docente por Materia.....	37
Figura 19: Diseño Listado de Todos los Docentes.....	38
Figura 20: Diagrama de Pruebas de Aceptación .....	41
Figura 21: Especificación de Pruebas de Aceptación .....	44
Figura 22: Especificación de Pruebas de Aceptación 2.....	41

## RESUMEN EJECUTIVO

El presente proyecto propone implementar un software para unidades educativas de nivel medio – bachillerato mediante el módulo de gestión académica.

Esta propuesta se ha llevado a cabo para mejorar el sistema actual con el que cuentan las instituciones de nivel medio – bachillerato mejorando procesos de notas para los estudiantes que se lleva a cabo dentro de las instituciones.

Con la falta de herramientas tecnológicas se aumenta el índice de pérdidas de información hasta llegar a la duplicidad de información ya que los procesos no son claros para los usuarios y causa mal estar en los estudiantes y padres de familia.

En el presente proyecto se presenta el desarrollo del módulo de gestión académica orientado a la web, esto quiere decir que tendremos una interfaz amigable a los padres de familia ya que podrá tener la información de las notas de los estudiantes.

El módulo de gestión académica permitirá al docente ingresar las notas por deberes, lecciones o trabajos en cada periodo de los quimestres.



---

## ABSTRACT

This project proposes to implement a software for mid-level educational units - high school through academic management module.

This proposal has been carried out to improve the current system that have mid-level institutions - high school grades improved processes for students takes place within institutions.

With the lack of technological tools index data loss is increased to reach the duplication of information and processes are not clear to users and cause illness in students and parents.

In this project we present the development of academic management module oriented web, this means that we will have a friendly interface to parents as they may have information of student grades.

The academic management module allows the teacher to input notes for homework, lessons or work in each period of quimestres.

---

## Capítulo I: Antecedentes

### 1.01. Contexto

El Ministerio de Educación ha dado grandes cambios en lo que respecta al campo educativo la educación de nivel Medio-Bachillerato ha implementado nuevos parámetros para este nuevo año en lo que respecta a notas con la implementación de los quimestres que serán de 2 quimestres comprendidos cada uno en 3 periodos cada uno. (Educacion, 2012)

El continuo desarrollo de la tecnología, ha llevado a las instituciones educativas del Ecuador a contar con sistemas informáticos que permitan mejorar la gestión de los procesos, garanticen la solidez de la información y en general, proporcionen nuevas formas de control.

En otras palabras, la disponibilidad de un sistema de información avanzado implica la adopción de un cambio en los procesos tradicionales y permite poner la información relevante a disposición de todos los miembros de la Institución a través de la Web.

El objeto de estudio se centra en el registro de matrículas, control de notas y asistencia, además con la emisión de información mediante generación de reportes para cada consulta que se requiera y ayuda en la planificación curricular docente, dicha información, en la actualidad se maneja de forma automática, limitando la entrega de información veraz y en tiempo real.

---

## 1.02 Justificación

Al implementar el sistema propuesto, tanto los estudiantes como el personal administrativo podrán eliminar los problemas mencionados anteriormente y obtener los beneficios como:

- Disminuir los inconvenientes causados tanto a los estudiantes como a los administrativos en el momento de requerir alguna información.
- Brindar satisfacción a los estudiantes ya que en el sistema que se implementara será el encargado de realizar las labores tan tediosas como son las consultas y la entrega de certificados.
- Niveles de seguridad de la información, para que sea accesible solo al personal autorizado. (Educacion, 2012)

Mediante la utilización del sistema vamos a lograr una buena toma de decisiones, ya que el objetivo en general que se busca es controlar los procesos académicos con la utilización de herramientas de última tecnología para mejorar y agilizar la gestión académica con esto se pretende mejorar la emisión de datos correcta, y no tener como resultado una información errónea.

Cabe resaltar que la viabilidad del mismo y su factibilidad son posibles pues se cuenta con datos reales y relevantes para aplicarlos a la investigación, además contamos con las herramientas necesarias y mínimas para desarrollar un sistema de estas características que brinde un servicio de calidad para aplicarlo directamente a la institución.




## 1.03 Definición Del Problema Central

**Tabla 1**

*Análisis de Matriz de Fuerzas T*

Problemática Agravada	Problemática Central				Problemática Resuelta
Sanción del Ministerio de Educación.	Procesos inadecuados de la gestión académica en las instituciones de nivel medio.				Mejorar los procesos de gestión académica.
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Software de interfaz amigable para las unidades educativas.	3	4	3	4	Desconocimiento del manejo del software por parte de los usuarios.
Información segura de las notas registradas por el docente.	3	4	3	4	Carencia de información sobre las notas de los estudiantes.
Cumplimiento de las normativas, reglamentos que otorga el Ministerio de Educación a la gestión académica	3	3	3	3	Poca colaboración en las obligaciones y responsabilidades en la gestión académica.
Permitir que el representante acceda a la información de su hijo por medio de una página web informativa	3	4	3	4	Poco uso de la tecnología por parte de los representantes

Nota: I= Intensidad PC=Potencial de cambio.

Escalas: Rojo  , 1 a 2 Impacto negativo importante, Amarillo  , 3 Impacto negativo medio Verde  , 4 a 5 Impacto positivo.

## Fuerzas impulsadoras

Software de interfaz amigable para instituciones educativas.

Intensidad	3	A causa de una interfaz complicada el usuario no comprende los procesos.
Potencial de Cambio	4	Al tener una interfaz interactiva los usuarios no tendrán problemas en entender los procesos del sistema

Información segura de las notas registradas por el docente.

Intensidad	3	La falta de generar reportes del estado de notas genera malestar en los estudiantes al no saber las notas de que se generaron.
Potencial de Cambio	4	Al generar los reportes los estudiantes podrá ver el estado de sus notas a través de una página web.

Cumplimiento de las normativas, reglamentos que otorga el Ministerio de Educación a la gestión académica

Intensidad	3	La falta de información de los nuevos cambios de calificaciones Generan malestar en todos los usuarios.
Potencial de Cambio	3	La información se la dará mediante charlas de los nuevos cambios que tienen las calificaciones.

Permitir que el representante acceda a la información de su hijo por medio de una página web informativa

Intensidad	3	El representante debe estar informado con lo que pasa en las Instituciones.
Potencial de Cambio	4	La información el cliente podrá acceder por medio de una página web por medio de un usuario y una contraseña accederá a toda la información de los estudiantes.

### Fuerzas bloqueadoras

Desconocimiento del manejo del software por parte de los usuarios

Intensidad	4	Los usuarios no comprenden las distintas partes del software Con lo que no ingresan la información correcta
Potencial de Cambio	3	El software no tiene muchas páginas para que el usuario tenga mayor rapidez en los procesos

Carencia de información sobre las notas de los estudiantes.

Intensidad	3	Las notas ingresadas por los docentes carecen de Información segura o veraz.
Potencial de Cambio	4	Los técnicos a cargo deber tomar por partes iguales los equipos para que la carga de trabajo no se acumule.

Poca colaboración en las obligaciones y responsabilidades en la gestión académica

Intensidad	3	Los involucrados no poseen obligaciones con respecto al modulo
Potencial de Cambio	3	Los involucrados tendrán responsabilidades en el modulo

Poco uso de la tecnología por parte de los representantes

Intensidad	3	Los representantes no tiene tanto conocimiento de tecnología
Potencial de Cambio	4	Se puede dar una breve charla del tipo de software a utilizar en este proyecto beneficios a futuro

---

## Capítulo II: Análisis De Involucrados

### 2.01. Requerimientos

#### 2.01.1. Descripción del sistema actual

Aportes de aprobación de curso: Este proceso iría a la mano del registro de notas, lo cual después de haber sido registradas las notas, la secretaria se encarga de realizar reuniones a padres de familia conjuntamente con el/la tutor/a donde se hace la entrega de dichos aportes, lo cual a veces existen aportes duplicados o información de alumnos cambiada.

Registro de notas: Esto se realiza como ha sido costumbre en el establecimiento a través de los docentes antes de forma trimestral, ahora por dos quimestres donde estos a su vez se dividen en cuatro periodos quienes deben llenar las hojas designada para notas de los cursos que hayan sido designados, una vez terminado esto las hojas de notas son enviadas a la secretaria la cual se encarga de realizar los aportes respectivos de cada estudiante con las notas obtenidas por cada uno todo esto según el modelo de calificación del LOEI 2012.

#### 2.01.2. Visión y Alcance

Mejorar los procesos antes mencionados con un sistema informático a la web que los automatizara para un mejor funcionamiento de los mismos y lograr los objetivos expuestos, y además con esto satisfacer las necesidades de la institución.

Además, el sistema pretende mejorar el sistema actual para poder tener una rapidez y eficacia al momento de la consulta de información de cada usuario incluido en el sistema.



## Alcance

El sistema permitirá a los usuarios, tener datos reales de las notas, asistencia de los alumnos además de permitir al representante ver las notas de sus hijos, el docente tendrá la facilidad de registrar a los estudiantes con mayor rapidez y con una interfaz amigable.

### 2.01.03 Entrevista

**Tabla 2**

*Entrevista para conocer el estado actual de los procesos*

#### DISEÑO ENTREVISTA

Identificador:001

Preguntas	Objetivos	Análisis posterior
<b>¿Cuál es la necesidad o el problema que se presenta en la institución?</b>	Conocer las necesidades y problemas a resolver	Diseñar un sistema informático a la web para el control de los procesos académicos
<b>¿Cuáles son los procesos que desea automatizar?</b>	Identificar cómo funcionan dichos procesos actualmente	Automatizar los procesos que actualmente funcionan manualmente para la entrega correcta de información
<b>¿Cómo es el proceso de matriculación actualmente?</b>	Saber cómo se maneja el proceso de matriculación	Mejorar la demora de tiempo que se tiene en este proceso
<b>¿Cómo es el proceso de registro de notas actualmente?</b>	Determinar cómo se lleva a cabo el registro de las notas	Registro sin equivocaciones de notas
<b>¿Cómo es el proceso de aprobación de curso actualmente?</b>	Establecer la entrega de reportes de aprobación con la información correcta	Entrega de aprobaciones con la información y notas correctas obtenidas por el estudiante
<b>¿Quiénes lo manejarían?</b>	Obtener el listado de los usuarios que manejarían el sistema	Las personas que tendrán acceso al sistema son:  Secretaria Docentes Estudiantes

## 2.01.04 Matriz de Requerimientos

**Tabla 3**

*Matriz de requerimientos especificación de procesos del sistema.*

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
<b>REQUERIMIENTOS FUNCIONALES</b>						
RF001	Automatizar los procesos que actualmente funcionan para la entrega correcta de información	Rectora	Alta	Software	En ejecución	Docentes Alumnos Secretaria
RF002	Mejorar el tiempo que se tiene en el proceso de notas	Secretaria	Alta	Software	En ejecución	Secretaria Alumnos
RF003	Registro sin equivocaciones las notas	Docentes	Alta	Software	En ejecución	Secretaria Docentes
RF004	Entrega de aprobaciones con la información y notas correctas obtenidas por el estudiante	Secretaria	Alta	Usuario	En ejecución	Alumnos
<b>REQUERIMIENTOS NO FUNCIONALES</b>						
RNF001	Las personas que tendrán acceso al sistema	Secretaria	Media	Usuario	En ejecución	Secretaria Docentes Alumnos
RNF002	El software debe visualizarse en cualquier navegador web.	Secretaria	Media	Usuario	En ejecución	Secretaria Docentes Alumnos

## 2.01.05. Descripción detallada.

**Tabla 4**

*Detalle de requerimiento mejorar procesos*

**Mejorar los procesos que actualmente funcionan para la entrega correcta de información**

Estado

**En ejecución**

Creado por	Carlos Cevallos	Actualizado por	Carlos Cevallos
Fecha Creación	22/12/2014	Fecha de actualización	22/12/2014
Identificador	RF 001		
Tipo de requerimiento	Crítico	Tipo de requerimiento	Funcional
Datos de entrada	Problemas encontrados en los procesos realizados automáticamente erróneos.		
Descripción	Al momento de consulta de información no es la correcta las aprobaciones de curso no tienen los datos correctos de los alumnos		
Datos de Salida	Información correcta		
Resultados esperados	Procesos Automatizados listos para ser utilizados		
Origen	Secretaria		
Dirigido a	Secretaria Docentes Alumnos		
Prioridad	Alta		
Requerimientos asociados	Ninguno		

### ESPECIFICACIONES

Precondiciones	Entrevista para encontrar los problemas de los procesos
Poscondiciones	Ninguno
Criterios de aceptación	Los usuarios están satisfechos por la información presentada.

**Tabla 5**
*Detalle del requerimiento registro de notas*

Registro sin equivocaciones de notas		Estado	En ejecución
Creado por	Carlos Cevallos	Actualizado por	Carlos Cevallos
Fecha Creación	22/12/2014	Fecha de actualización	22/12/2014
Identificador	RF 002		
Tipo de requerimiento	Crítico	Tipo de requerimiento	Funcional
Datos de entrada	Obtención de notas por parte del docente		
Descripción	Luego de la obtención el docente se encargara de registrar las notas al sistema de cada alumno		
Datos de salida	Registro correcto con las notas obtenidas por cada alumno		
Resultados esperados	Cada nota ingresada que pertenezca al alumno		
Origen	Secretaria		
Dirigido a	Docente Alumno		
Prioridad	Alta		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	Ninguno		
Poscondiciones	Ninguno		
Criterios de aceptación	El alumno estará satisfecho por sus notas correctas ingresadas al sistema		

**Tabla 6**

*Detalle del requerimiento entrega de aprobaciones*

**Entrega de aprobaciones con la información y notas correctas obtenidas por el estudiante**

Estado

**En ejecución**

Creado por	Carlos Cevallos	Actualizado por	Carlos Cevallos
Fecha Creación	22/12/2014	Fecha de actualización	22/12/2014
Identificador	RF 004		
Tipo de requerimiento	Crítico	Tipo de requerimiento	Funcional
Datos de entrada	Información del alumno y notas obtenidas		
Descripción	Con la información se podrá generar un reporte en el cual se sabrá si el alumno es promovido al siguiente año		
Datos de salida	Aprobación de curso		
Resultados esperados	Generación de reporte		
Origen	Secretaria		
Dirigido a	Alumnos		
Prioridad	Alta		
Requerimientos /asociados	Ninguno		

#### ESPECIFICACIONES

Precondiciones	Registro de notas
Poscondiciones	Ninguno
Criterios de aceptación	Representante conforme por información actualizada

## 2.02 Mapeo de involucrados.

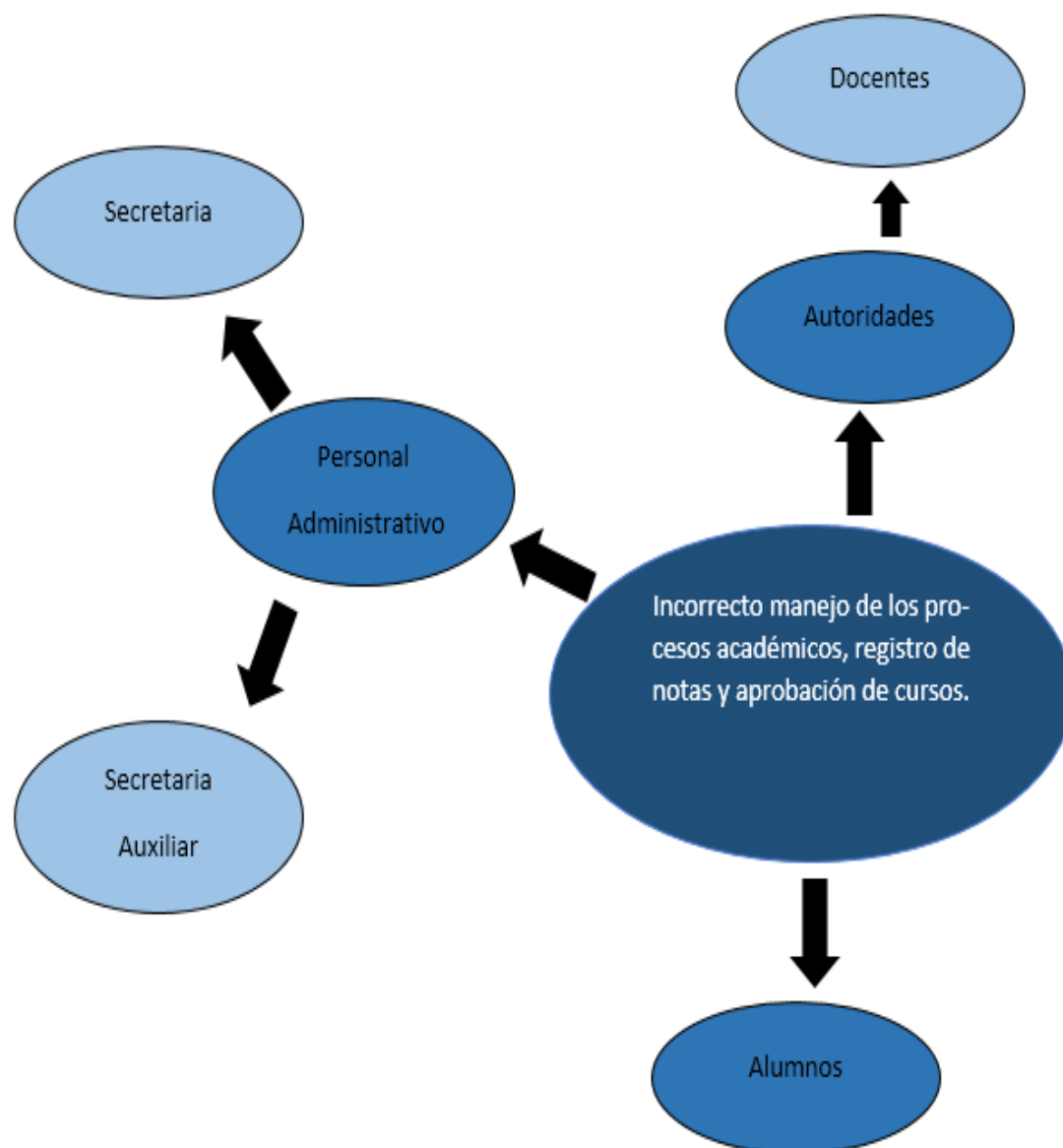


Figura 1: Mapeo de involucrados determinación del problema central.

### **2.02.01 Análisis mapeo de involucrados.**

Al analizar el mapeo de involucrados se determina el problema central que son los procesos inadecuados procesos de los sistemas académico registro de notas y aprobación de cursos en las instituciones de nivel medio se realiza una investigación de campo en la que se pueda determinar cuáles son los involucrados las instituciones privadas y públicas en el problema existente; por lo que se determina los siguientes:

El personal administrativo de la institución se ve afectado por el problema que le ocasiona el funcionamiento del actual sistema con el que cuentan ya que no entregan una información correcta de las notas de los alumnos.

Los docentes tienen pérdida de información por lo que los registros de notas no son los correctos para la entrega y aprobación de la misma al no tener un sistema que les ayude al ingreso de la misma.

Los estudiantes no obtienen la información esperada por parte de la institución ya que no cuentan una página que les entrega esta información.

### **2.03 Matriz de Análisis de Involucrados**

La matriz de involucrados de involucrados detalla los autores involucrados sobre el interés del problema el cual el sistema realizara procesos, serán detallados cada uno de los mismos para obtener información concreta sobre los cambios y conflictos de tenga el sistema aplicado en las instituciones públicas y privadas.

El aplicativo tendrá una solución para el conflicto que se presente en los establecimientos y poder cumplir leyes y normas otorgados para evitar el cierre de establecimientos educativos y brindar un excelente servicio a cada estudiante y representante que se encuentre ingresado en el sistema.

Además de una ayuda para el control de asistencia de los alumnos al establecimiento y horas de clase, el cual tiene como beneficio

Véase Anexo A.01



## Capítulo III: Problemas y Objetivos

### 3.01. Árbol de Problemas

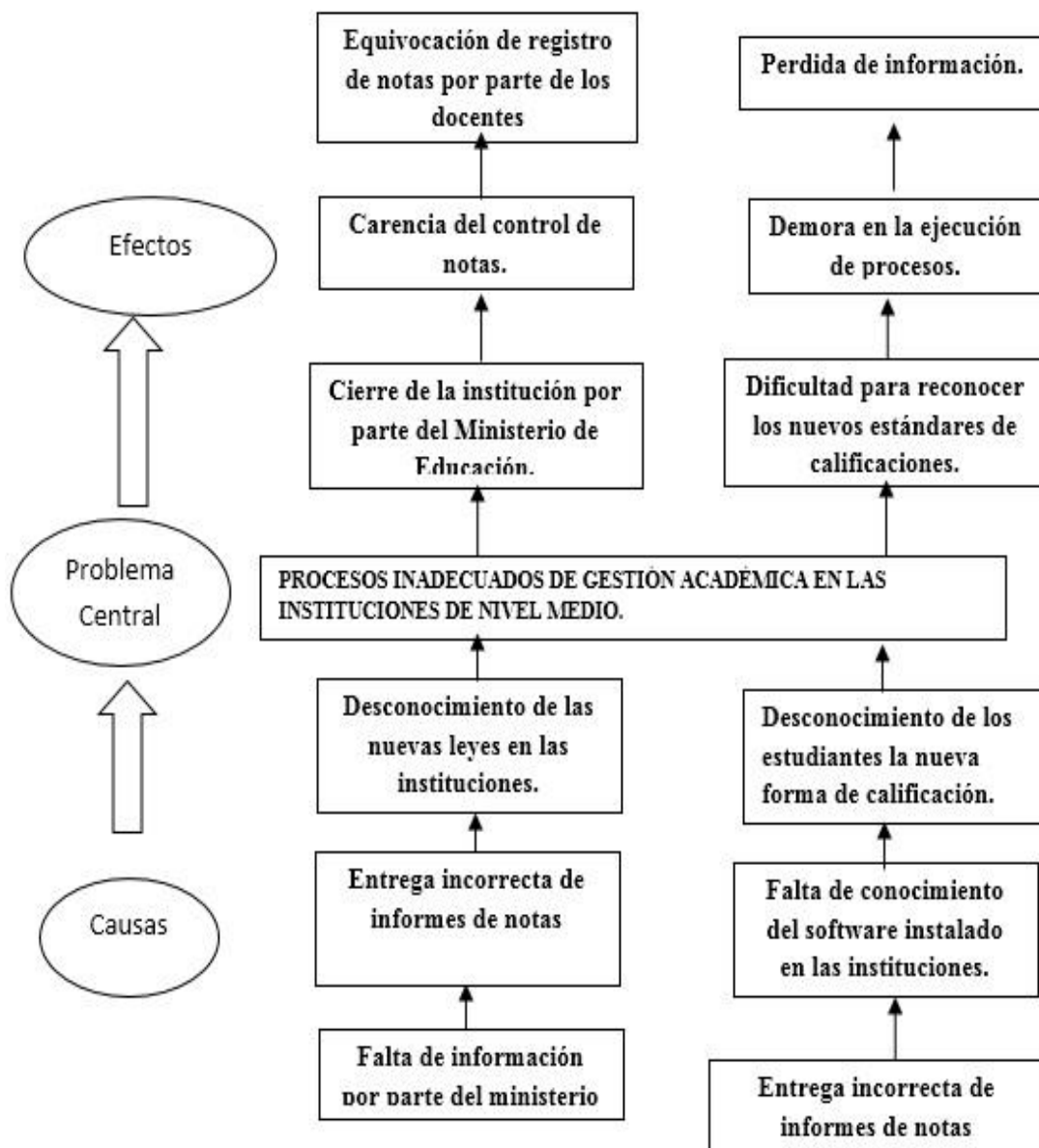


Figura 2: Árbol de Problemas identificación del problema más importante

### 3.02. Árbol de Objetivos

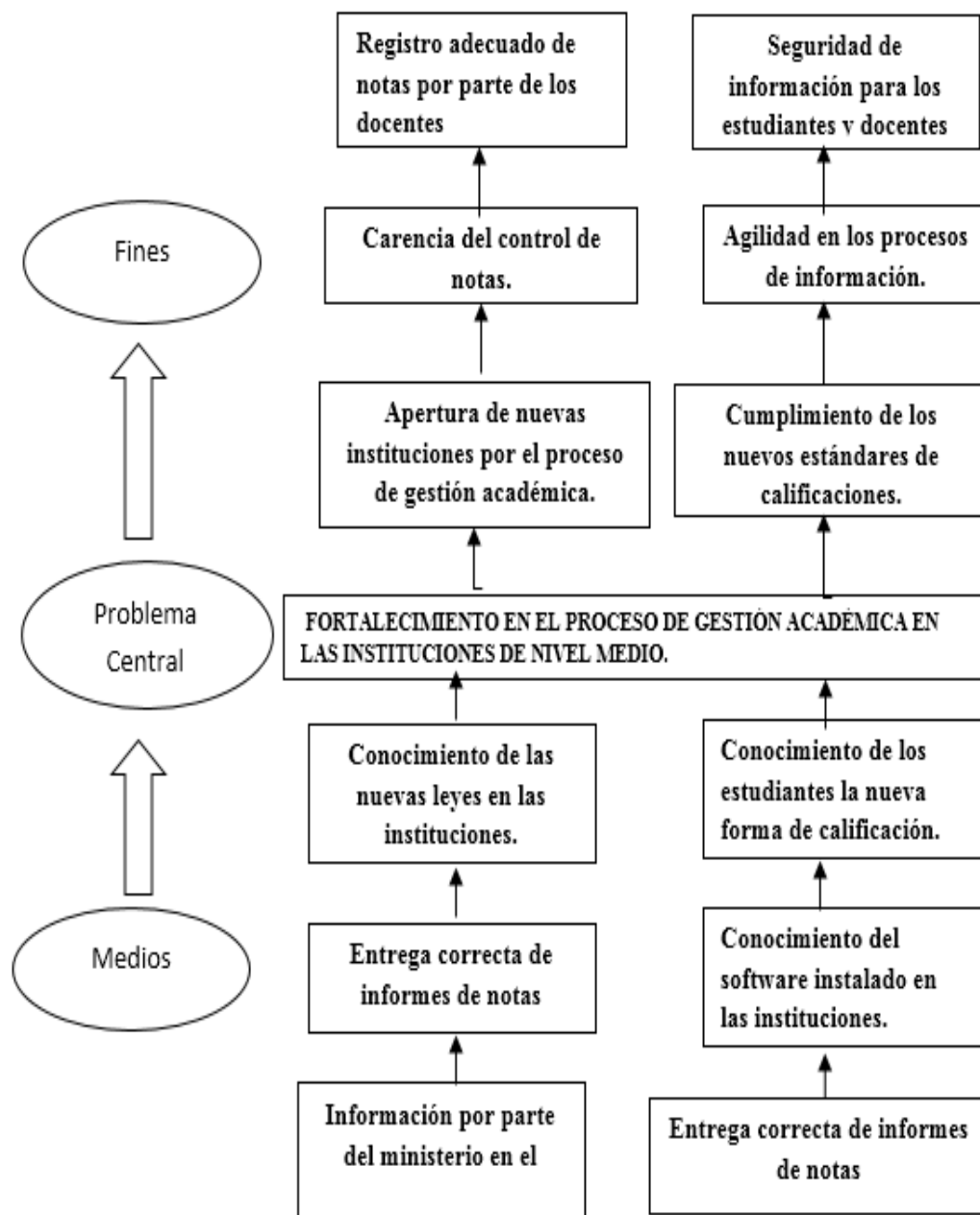


Figura 3: Árbol de Objetivos situación esperada

### 3.03. Diagramas de casos de uso

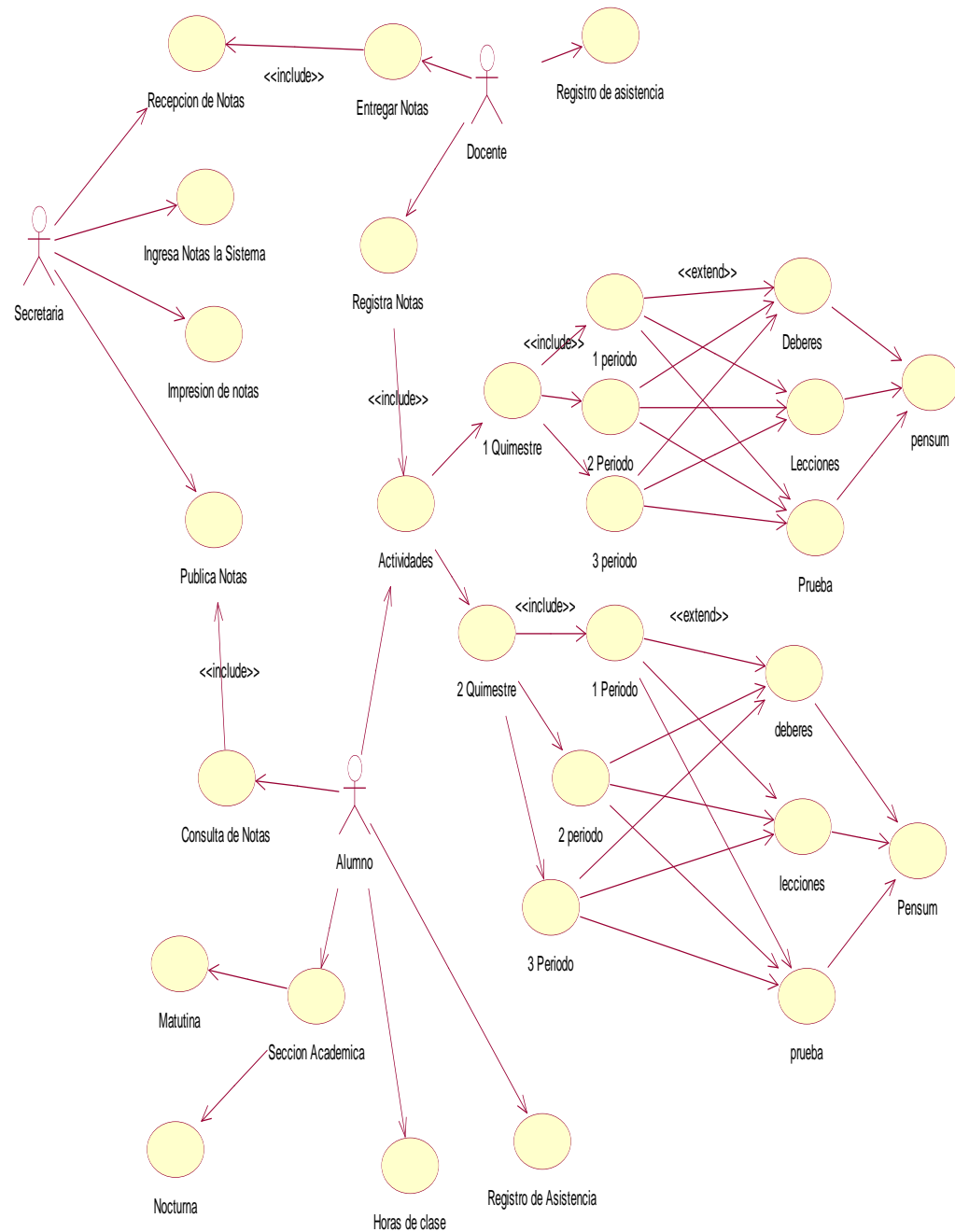


Figura 4: Diagrama de caso de uso comportamiento del sistema.

### 3.04 Casos de uso de realización

#### DIAGRAMA DE REALIZACION 1

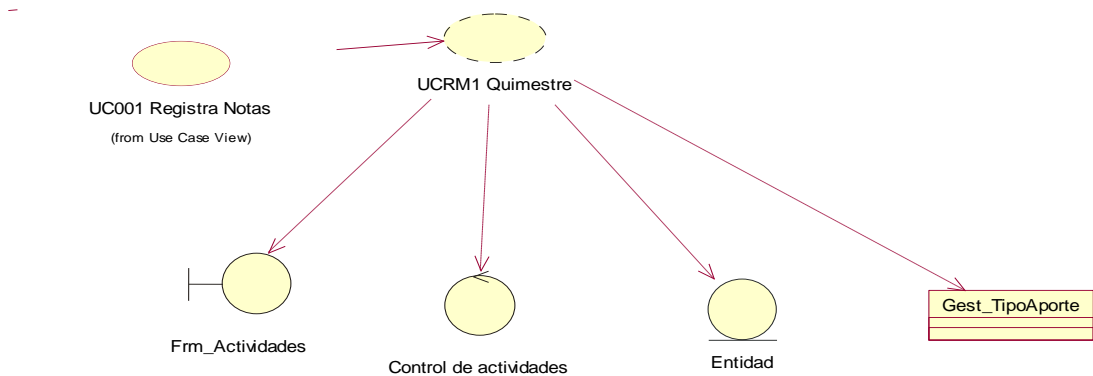


Figura 5: Diagrama de realización CUR01 interfaz Quimestres

**Tabla 7**

*Especificación de caso de uso de realización de quimestres.*

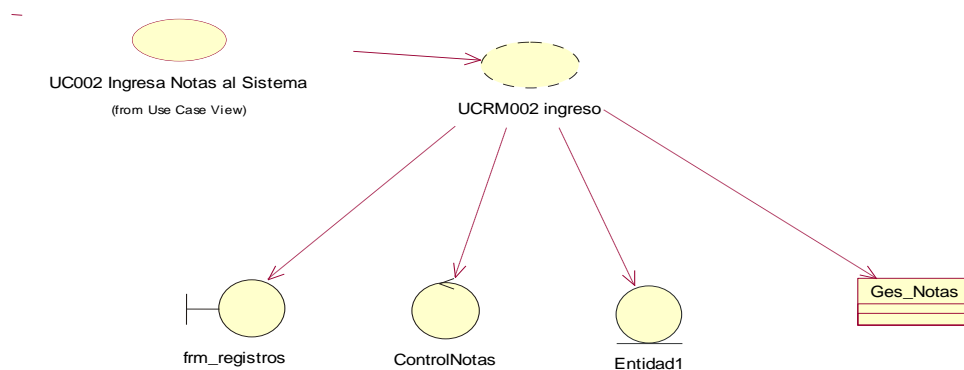
<b>Nombre</b>	Quimestres
<b>Identificador</b>	UC001Registro notas
<b>Responsabilidades</b>	RF1 El docente registra las notas del alumno según las actividades realizadas en los quimestres.
<b>Tipo</b>	Usuario
<b>Referencias Casos de Uso</b>	UC001
<b>Referencias Requisitos</b>	RF1
<b>PRECONDICIONES De Instancia</b>	
Se instancia con los periodos del alumno	
<b>De Relación</b>	
Tiene relación de interfaz con el docente y secretaria	
<b>POSCONDICIONES De Instancia</b>	
La instancia de periodos lo realizan los docentes.	
<b>De Relación</b>	
La tabla de detalles de notas interpreta los periodos académicos	
<b>SALIDAS PANTALLA</b>	
La información se despliega dando opciones para asignar la calificación por periodos de los quimestres	

**Tabla 8**

*Especificación caso de uso quimestres*

Caso de Uso	<i>Quimestres</i>
Identificador	<i>UC001Registro notas</i>
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
<p><i>1.- El docente registra las notas según el periodo y quimestres.</i></p> <p><i>1.- Guarda la información de lo realizado en el día.</i></p>	<p><i>1.- El formulario se despliega para que pueda ingresar las notas según periodo y quimestres.</i></p> <p><i>Actividad que ejecuta el sistema con identificador de orden</i></p>
CURSOS ALTERNATIVOS	
<p><i>Los docentes llenaran en orden los periodos por cada quimestres</i></p> <p><i>Los docentes sacan notas de quimestres después de a ver llenado todos los periodos</i></p>	

**DIAGRAMA DE REALIZACION 2**



*Figura 6: Diagrama de realización interfaz ingreso*

**Tabla 9**

*Especificación de caso de uso de realización de ingreso.*

<b>Nombre Identificador</b>	<i>Ingreso</i>
<b>Responsabilidades</b>	UC002 Ingreso de notas al sistema
<b>Tipo</b>	RF2 La secretaria ingresa las notas entregadas por el profesor de los alumnos.
<b>Referencias Casos de Uso</b>	Usuario
<b>Referencias Requisitos</b>	UC002
<b>PRECONDICIONES De Instancia</b>	RF2
<b>PRECONDICIONES De Instancia</b>	
<i>Esto se instancia con las notas que registro el docente</i>	
<b>De Relación</b>	
Tiene una relación grafica con la secretaria y el docente	
<b>POSCONDICIONES De Instancia</b>	
La instancia de ingreso lo realiza los docentes	
<b>De Relación</b>	
Tiene una relación grafica con los docentes	
<b>SALIDAS PANTALLA</b>	
Se despliega todos los campos para ingresar notas	

**Tabla 10**

*Especificación caso de uso ingreso*

<b>Caso de Uso Identificado</b>	<i>Ingreso</i>
<b>CURSO TÍPICO DE EVENTOS</b>	UC002 Ingreso de Notas al Sistema
<b>Usuario</b>	<b>Sistema</b>
1.- El docente registra las notas según el periodo y quimestres.	1.- El formulario se despliega para que pueda ingresar las notas según periodo y
1.- Guarda la información de lo realizado en el día.	Actividad que ejecuta el sistema con identificador de orden
<b>CURSOS ALTERNATIVOS</b>	
El docente esperar tener todas las notas de los estudiantes.	

### DIAGRAMA DE REALIZACION 3

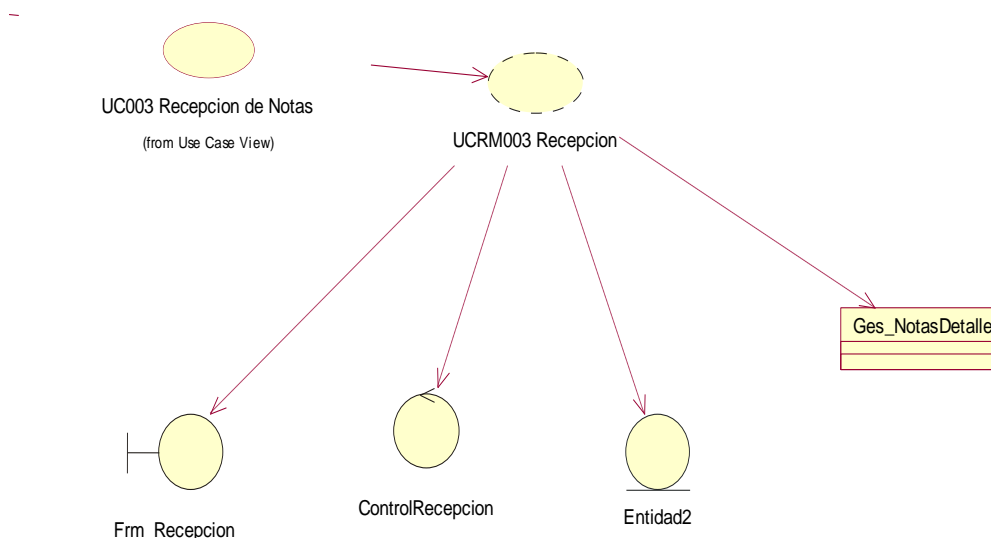


Figura 7: Diagrama de realización interfaz recepción.

**Tabla 11**

*Especificación de caso de uso de realización de recepción.*

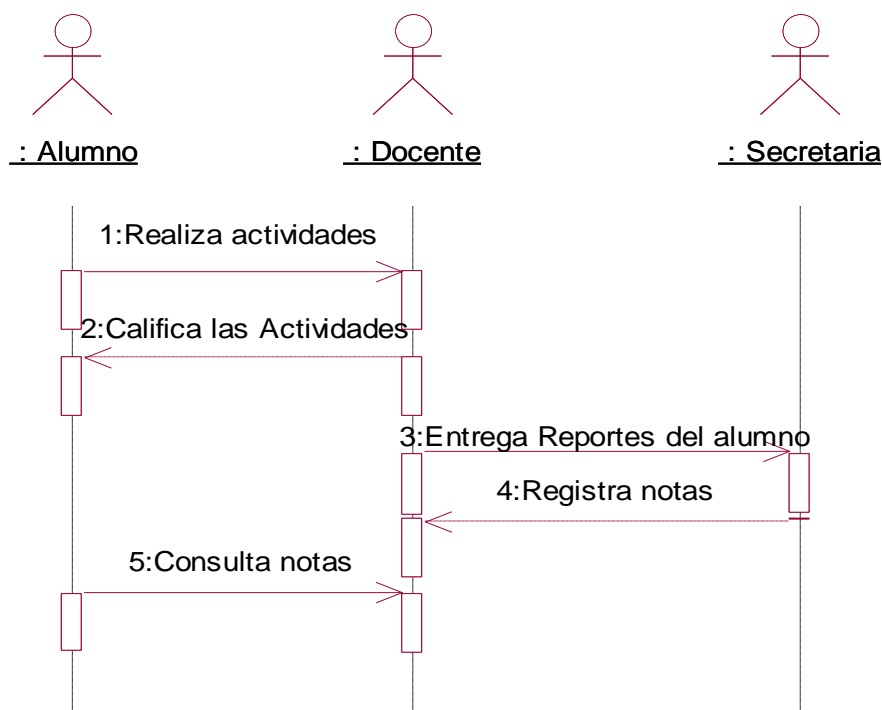
Nombre	Recepción
<b>Identificador</b>	UC003 Recepción de Notas
<b>Responsabilidad</b>	RF3 La secretaria recepta las notas entregadas por el docente.
<b>Tipo</b>	Usuario
<b>Referencias Casos de Uso</b>	UC003
<b>Referencias Requisitos</b>	RF3
<b>PRECONDICIONES</b>	
<b>De Instancia</b>	Se instancia con las notas que entrega el docente
<b>De Relación</b>	Tiene relación de interfaz con la secretaria
<b>POSCONDICIONES</b>	
<b>De Instancia</b>	La instancia lo realiza la secretaria
<b>De Relación</b>	Los procesos lo realiza la secretaria
<b>SALIDAS PANTALLA</b>	
Se despliega las notas ingresadas por el docente	

**Tabla 12**

*Especificación de caso de uso recepción.*

Caso de Uso	Recepción
Identificador	UC003 Recepción de notas
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
1.- El profesor entrega las notas de los estudiantes a la secretaria	1.- el profesor ingresa las notas en el sistema desplegable y entrega la hoja de quimestres respectivo.
2.- La secretaria recepa las notas para proceder con el siguiente proceso	2.- La secretaria verifica en el sistema las notas con la hoja quimestres.
CURSOS ALTERNATIVOS	
LA SECRETARIA ESPERA PARA PUBLICAR NOTAS	

### 3.05. Diagrama de secuencia del sistema



*Figura 8: Diagrama de Secuencia interacción del Registro de notas*



---

## Capítulo IV: Análisis de Alternativas

### 4.01. Matriz de Análisis de Alternativas

La matriz de análisis de alternativas es la encargada de la observación de objetivos del sistema que se aplican en los establecimientos educativos con sus diferentes impactos sobre el propósito, factibilidad financiera, técnica, social y política elevamos un porcentaje de (Baja, Media, Alta) para cada categoría y obtener un resultado total de cómo se ejecuta el aplicativo para establecimiento.

Los siguientes objetivos lo realizamos a través de una investigación:

Mejorar los resultados en el cumplimiento de la programación.

Optimizar el tiempo de respuesta.

Agilizar el proceso de registro de notas.

Controlar todos los procesos de registro de notas de los estudiantes.

Informar al representante de las notas de su representado.

Véase Anexo A.02

## 4.02. Matriz de Impacto de los Objetivos

**Tabla 13**

*Matriz de análisis de impacto de los objetivos*

	Factibilidad de Lograrse  (alta-media- baja) 4 - 2 - 1	Impacto de Género  (alta- media- baja) 4 - 2 - 1	Impacto Ambiental  (alta- media- baja) 4 - 2 - 1	Relevancia  (alta- media- baja) 4 - 2 - 1	Sostenibilid ad  (alta- media- baja) 4 - 2 - 1	Total
Objetivos	Los beneficios son los estudiantes de las instituciones de nivel medio.	Increment o de fuentes de Trabajo	Mejora el entorno cultural respecto al uso de la computador a	Beneficia a los estudiantes de los colegios	Fortalece la participació n del personal de la institución.	<b>75 Puntos</b>  15-35 BAJA
	Ayuda por parte de una institución de nivel medio en el desarrollo de la aplicación.	Mayor participaci ón del personal técnico	Mejora el ambiente de trabajo con tecnología	Efectividad para registrar notas.	El servicio inicia por parte del profesor.	36-56 MEDIA ALTA
	Colaboració n de los empleados en el análisis	Aumento de responsab ilidades en el personal de la empresa	Aumento de ingresos para la empresa	Disminució n de estudiantes inconforme s	Mejoramien to del software	57-75 ALTA
	<b>15 puntos</b>	<b>13 Puntos</b>	<b>10 Puntos</b>	<b>20 Puntos</b>	<b>17 Puntos</b>	

---

#### **4.03. Estándares para el Diseño de Clases**

Lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modelling Language) es un lenguaje estándar para la especificación, visualización, construcción y documentación de artefactos de sistemas de software, muy bueno para la modelación de negocios y otros sistemas que no son software. El UML representa una colección de las mejores prácticas de ingeniería que tienen una probación exitosa en la modelación de sistemas largos y complejos

El UML es una parte muy importante para el desarrollo de software orientados a objetos y en el proceso de desarrollo de software. Utiliza, en su mayor parte, notaciones gráficas para expresar para expresar los proyectos de diseño del software. Utilizando el ayudante del UML puede comunicar el equipo de proyecto, explorar el potencial de diseños, y validar el diseño de la arquitectura del software.

Véase Anexo A.03

#### **4.04. Diagrama de Clases**

El diagrama de clases describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Véase Anexo A.03

#### **4.05. Modelo Lógico – Físico**

El modelo lógico contiene el proceso de la información que utilizan las unidades educativas.

El modelo físico traduce el diseño lógico ya que es una solución para implementar en las Unidades Educativas

Véase Anexo A.04

#### 4.06. Diagrama de Componentes

Los diagramas de componentes tiene la estructura del sistema finalizado maostando los diferentes tipos de componentes utilizados.

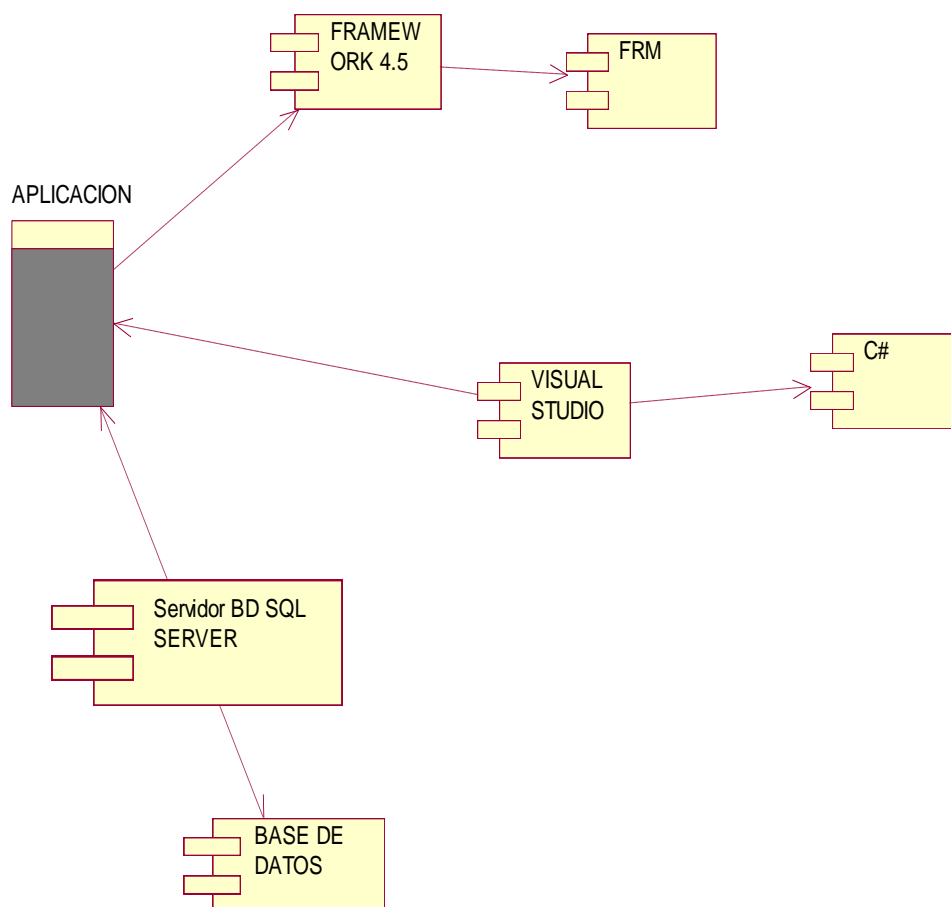


Figura 9: Diagrama de Componentes

#### 4.07. Diagrama de Estrategias

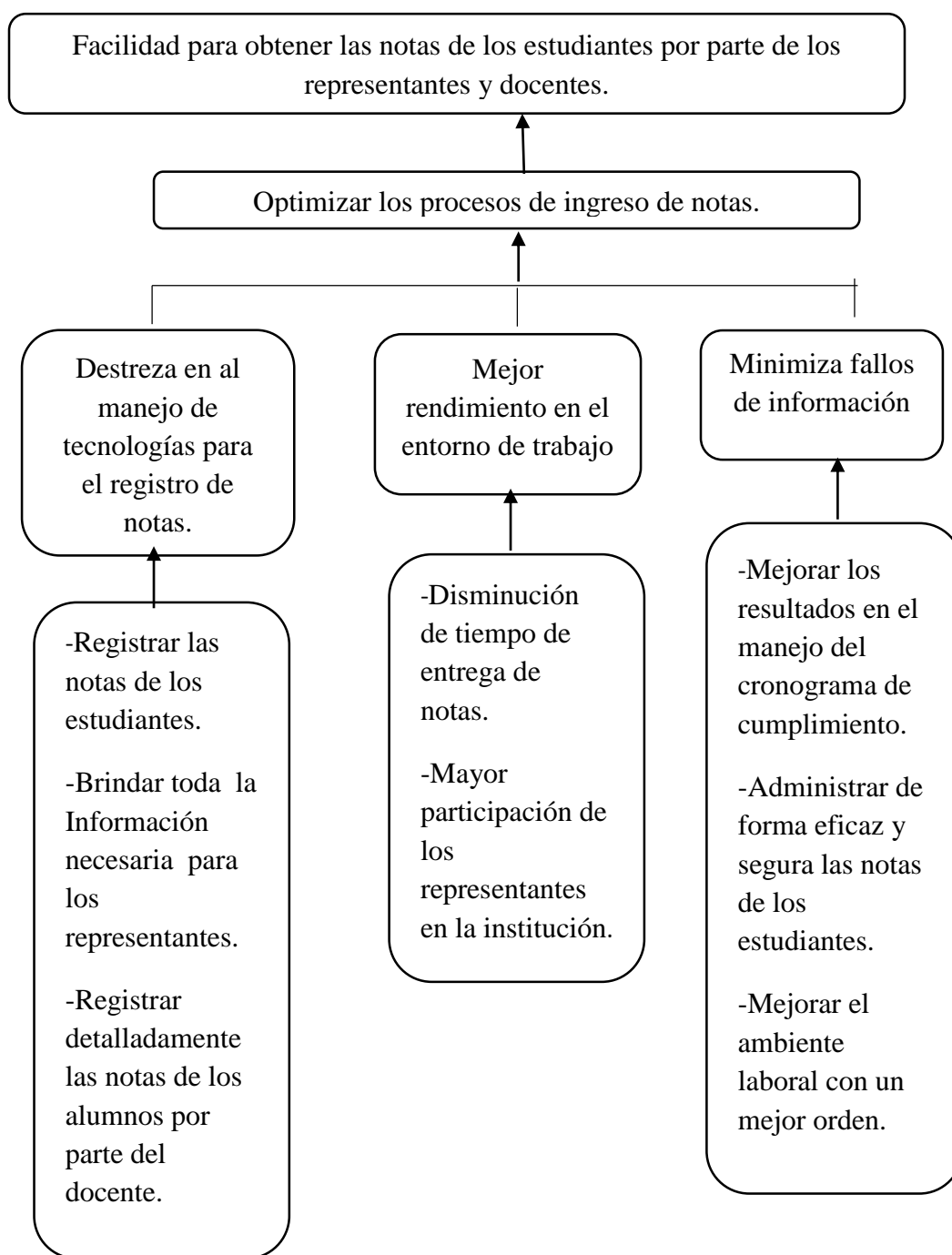


Figura 10: Diagrama de Estrategias

## 4.08. Matriz de Marco Lógico

**Tabla 14**

*Matriz del marco lógico*

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<b>Finalidad</b>  Mejor control y registro de las notas de los estudiantes, para optimizar tiempo de respuesta, logística.	Se implementara la aplicación en funcionamiento para las instituciones de nivel medio.	- Diagnóstico del Jefe de sistemas.	No existen los recursos necesarios para la realización de los soportes.
<b>Propósito</b>  Eficiencia en la gestión de soporte técnico.	La aplicación registrara todos los procesos realizados en este módulo.	- Informes de los usuarios del módulo.	Utilización inadecuada de esta herramienta por parte de los usuarios.
<b>Componentes</b>  - Registros y control de todos los procesos de notas.	- Porcentaje de representantes que realizan el servicio en línea.	- Reportes de las notas del sistema al final del día.	Clientes insatisfechos por el servicio.
<b>Actividades</b>  - Registrar las notas con seguridad. - Actualizar las técnicas de control de notas.	- Disminución en el tiempo de entrega de notas. - Efectividad en la entrega de notas.	- Establecer un cronograma con fechas y horarios para cumplir con los ingresos de notas.	- El personal utilice la herramienta apropiadamente.

## 4.09. Vistas Arquitectónicas

La arquitectura de software describe cómo un sistema es descompuesto en componentes, cómo éstos son interconectados, y la manera en que éstos se

comunican e interactúan entre sí. Cada vista representa un comportamiento particular del sistema.

#### 4.09.01 Vista Física

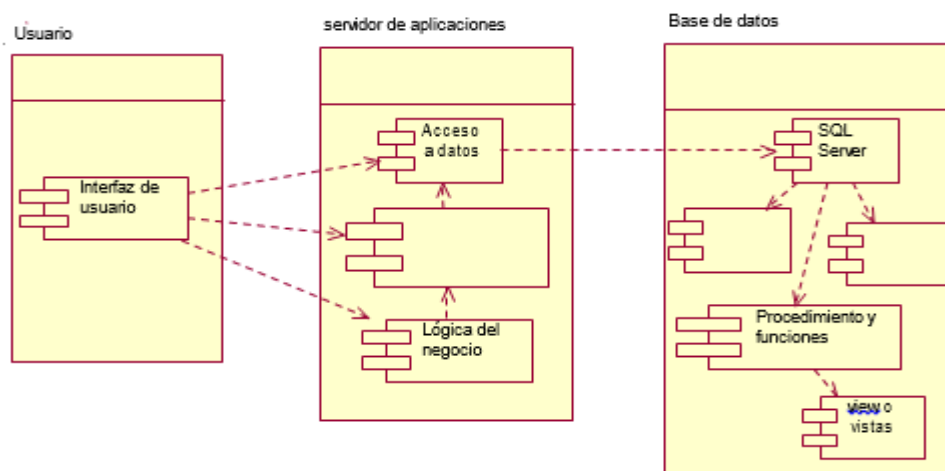


Figura 11: Diagrama vista Física

Se toma en cuenta los requisitos no funcionales del sistema tales como, disponibilidad, confiabilidad, desempeño entre otras más.

El sistema se ejecuta sobre varios nodos de procesamiento (hardware). Estos nodos son relacionados con los elementos identificados de las vistas anteriores.

En esta vista se especifican varias configuraciones físicas. Por ejemplo, una para el desarrollo y las pruebas, o para el despliegue del sistema en plataformas distintas.

Capa1 Presentacion.- Navegadores

Capa 2 De Negocio .- C#

Capa 3 De Datos.- Sql server

## Diagrama de Despliegue

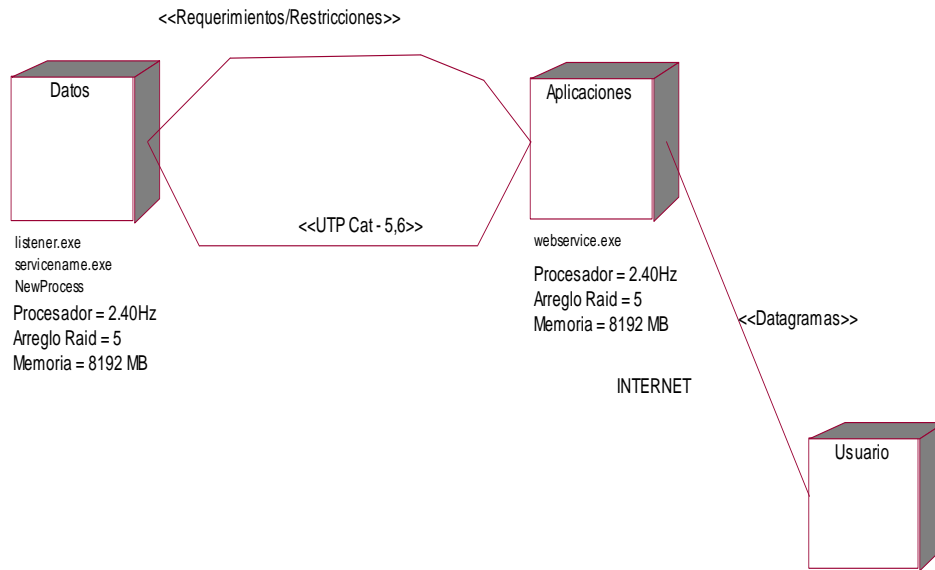


Figura 12: Vista Física diagrama de despliegue

## Vista Lógica

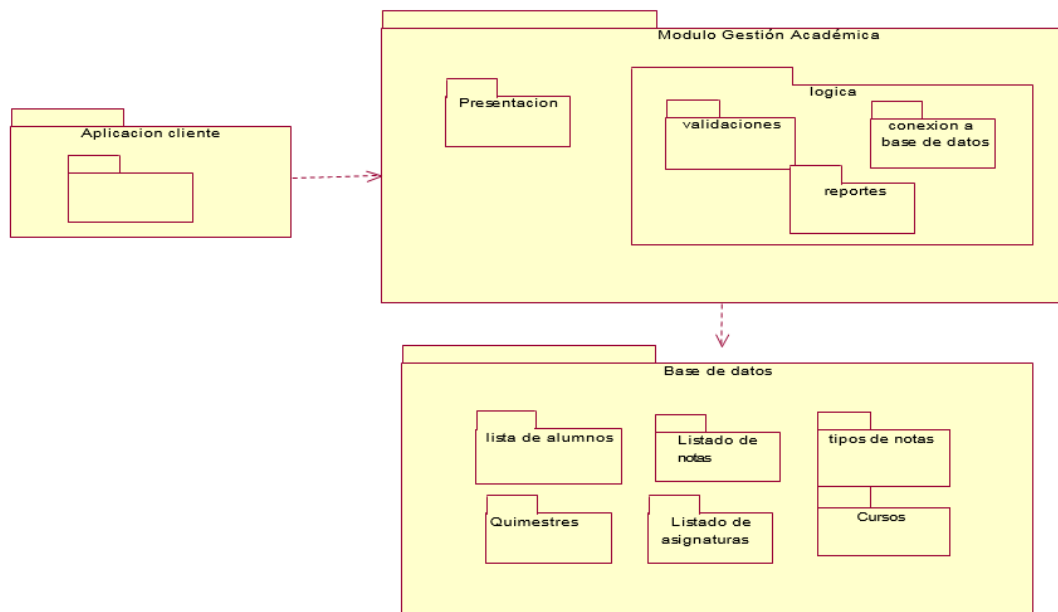


Figura 13: Vista Logica



## Capítulo V: Propuesta

### 5.01. Especificación de estándares de programación

Este punto tiene como objetivo reglamentar la forma en que se implementará el código fuente del proyecto, pasando, por las variable, controles, ficheros, archivos y todo aquello que esté implicado en el código.

#### 5.01.1. Declaraciones de Variables.

- La longitud debe ser lo más recomendable posible.
- El tipo de dato al que pertenece la variable.

**Tabla 14**

*Descripción de la variable*

Estructura	Descripción de la Variable
LONGITUD. MAX.	← 1 → ← 16 →
FORMATO	Todo con minúsculas
EJEMPLO	\$meses, \$a = 1; \$cadena="Hola amigo";

**Tabla 15**

*Descripción tipos de variables.*

Tipo de variable	Descripción
Integer	[FAC_NUMERO] [int] NULL,
Char	[PEN_MATERIA] [varchar] (150) NULL,
Numeric	[REP_CODIGO] [numeric] (30, 0) NOT NULL,
Date	[FAC_FECHA] [datetime] NULL,
Double	[QUI_NOTA] [decimal] (8, 2) NULL,

**Tabla 16**

*Descripción de tipo de control.*

Tipo de control	Prefijo	Ejemplo
<b>Label</b>	<b>Lbl</b>	<b>Label#label1</b>
<b>TextBox</b>	<b>Txt</b>	<b>TextBox#txtl2</b>
<b>Button</b>	<b>Btn</b>	<b>ButtonBtnRegistrar</b>
<b>Image</b>	<b>Img</b>	<b>Imge#Image1</b>
<b>GridView</b>	<b>Dgv</b>	<b>GridView#dgvEstudiante</b>
<b>DropDownList</b>	<b>Cmb</b>	<b>DropDownList#Cmbquimestre</b>
<b>Calendar</b>	<b>Dtp</b>	<b>Calendar#dtpFecha</b>

## Declaración de Clases

**Tabla 17**

*Descripción declaración de clases*

Título	Descripción
Sintaxis	<code>public partial class</code>
Descripción	El tipo de variables son Private, Public o Protected.
Observaciones	<p>En la declaración de clases no se deberá utilizar caracteres como:</p> <p>Letra Ñ o ñ.</p> <p>Caracteres especiales ¡^, #, \$, %, &amp;, /, (,), ¿, +, -, *, {, }, [, ].</p> <p>Caracteres tildados: á, é, í, ó, ú.</p>
Ejemplo	<p><code>public partial class</code></p> <p>Indica una clase parcial</p>

## 5.02. Diseño de Interfaces de Usuario.

La interfaz diseñada para el usuario es el proceso de determinar los distintos componentes, tanto de hardware como de software, sus características y su disposición, que se utilizarán para interactuar con una serie de usuarios determinados



Figura 14: Diseño general de interface de usuario.

1. Label#labelIngreso.- Nos indica que el ingreso al sistema para el usuario.
2. Menu#NavigationMenu.- Menú desplegable de las diferentes opciones que tiene el sistema.
3. Image#Image.- Es una imagen de bienvenida del sistema.

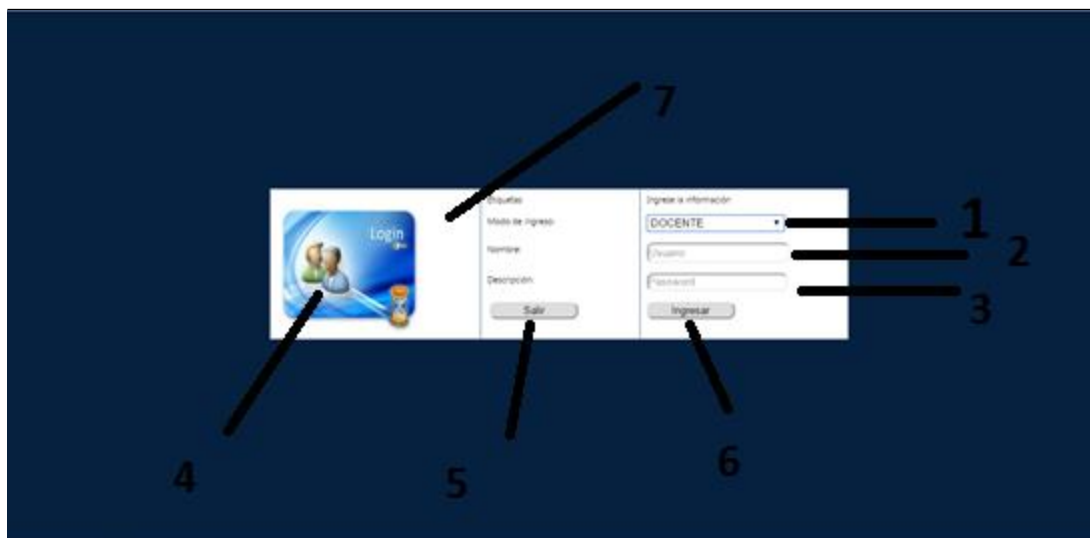


Figura 15: Diseño general de ingreso de usuarios.

- 1.- DropDownList.- Nos servirá para poder seleccionar el tipo de usuario
2. TextBox #txtUsuario.- Aquí escribiremos nuestro nombre de usuario
3. TextBox #txtContraseña.- dentro de esta caja de texto pondremos nuestra contraseña la cual será reemplazada por puntos para mayor seguridad.
4. Image #Image1.- Imagen para identificar el login de usuario del sistema.
5. Button#btnSalir.- Este botón nos permitirá regresar al inicio de la página.
6. Button#btnRegistrar.- El botón nos permitirá ingresar al sistema mientras ingresemos el usuario y contraseña correcta.

**SISTEMA DE MATRICULACIÓN MODULO DOCENTE**

Seleccione un Estudiante: **Perez 2 Luis Q** LISTAR CALIFICACIONES Seleccione un Cuatrimestre: **PRIMERO** **Buscar**

COD	N	T1	AC1	TA1	L1	PE1	P1	T2	AC2	TA2	L2	PE2	P2	T3	AC3	TA3	L3	PE3	P3	FE	EQ	PI
Calific	443	Física	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Calific	445	Literatura	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Calific	446	Matemáticas	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**INGRESAR CALIFICACIONES**

T1:  AC1:  TA1:  L1:  PE1:

T2:  AC2:  TA2:  L2:  PE2:

T3:  AC3:  TA3:  L3:  PE3:

**Guardar**

Figura 16: Diseño general de ingreso de notas del docente

- 1.- DropDownList nos servirá para poder seleccionar al estudiante.
- 2.- Button#btnBuscar el botón nos permitirá buscar el alumno
- 3.- DropDownList#Cmbquimestre se visualiza las notas ingresadas.
4. Table#tdingrso se ingresa las notas de los estudiantes.

**SISTEMA DE MATRICULACIÓN MODULO DOCENTE**

Principal Calificaciones Asistencia Salir del sistema

**ASISTENCIA MAXIMA**

**ASISTENCIA DEL ESTUDIANTE**

0

**Guardar asistencia**

Figura 17: Diseño general de ingreso de Asistencia alumnos

1. Menu#NavigationMenu nos despliega los diferentes opciones que tiene el sistema.
2. Tabla#td.- Contiene el calendario de fechas de asistencia de alumnos.

3. Calendar#dtpFecha.- Aquí se selecciona las fechas de asistencia de los alumnos
4. Button#btnRegistrar.- Registra las notas ingresadas por el docente.
5. Label#Label3.- Texto para identificar el contenido del menú.

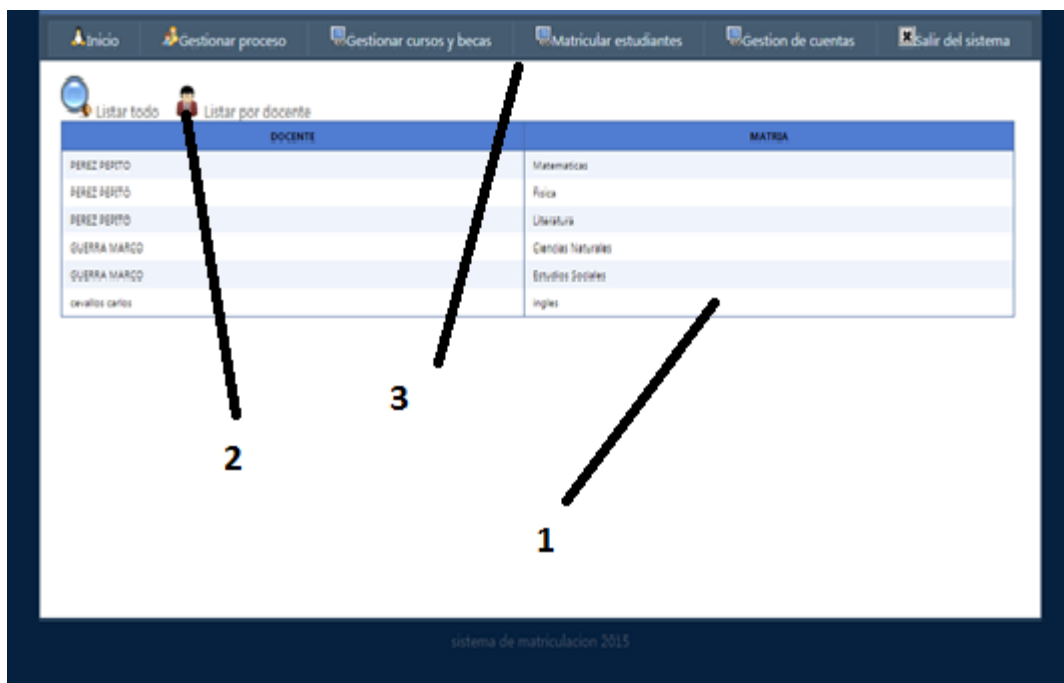


Figura 18: Diseño listado de docente por materia.

1. GridView#dgvTodoDocente.- Este grid nos permite visualizar los profesores y las materias que dictan.
2. ImageButton#btnTodo0.- Icono de listar los profesores
3. Menu#NavigationMenu.- Menú desplegable de las diferentes opciones que tiene el sistema.

Nombre del docente   LISTAR DOCENTE

CEDULA	APELLIDO	NOMBRE	DIRECCION	TELEFONO	CORREO	CONTRASEÑA
	PEREZ	PEPITO	EL SUR	0998765432	akhdhaviakhdas@hotmail.com	
	GUERRA	MARCO	KHASDKHVKDV	09876543	khsadihasdbk@hotmail.com	
	jghhj	gfthg	kbbhb	1234567	kj@hotmail.com	
1723002059	cevallos	carlos	el bosque	123456	coupermg99@gmail.com	1723002059
1723002059	cevallos	carlos	el bosque	123456	coupermg99@gmail.com	1723002059

Figura 19: Diseño listado de todos los docentes.

1. GridView#dgvDocente.- Se encuentra toda la información del docente.
2. Label#lbllistar.- Etiqueta de información de ventana.
3. Button#btnBuscar.- Boton que nos permite buscar a los profesores que se encuentran en el plantel.
4. textBox#txtBuscar.- Caja de texto que nos permite ingresar el nombre del docente a buscar.

### 5.03. Especificación de pruebas de Unidad.

Una prueba de unidad pretende probar cada función en un archivo de programa simple (una clase en terminología de objetos). Las librerías de pruebas de unidad formalizan este trabajo al proporcionar clases para pruebas.

**Tabla 18**

*Descripción de pruebas de unidad interface PRU\_UNI01*

Identificador de la Prueba:	PRU_UNI01
Método a Probar	<b>Interface</b>
Objetivo de la Prueba	Tiene como objetivo verificar la correcta inserción de notas del alumno por parte del docente
Datos de Entrada:	
<b>Ingreso al sistema login</b>	
<b>Identificación de los iconos de acceso al sistema</b>	
<b>Correcto despliegue de los diferentes menús del sistema.</b>	
Resultados Esperados	
<b>Encontrar errores que podrían detener el funcionamiento de la interfaz de usuario.</b>	
Comentarios	<b>Validaciones correctas de usuarios</b>
<b>Los iconos deben entendidos por los usuarios y no solo por el programador</b>	
<b>Adecuado formato de la interfaz</b>	



**Tabla 19**

*Descripción de pruebas de unidad reportes PRU\_UNI02*

Identificador de la Prueba:

PRU\_UNI02

Método a Probar	Reportes
Objetivo de la Prueba	Evaluar los reportes que realiza la secretaria.
Datos de Entrada:	
<b>Búsqueda del alumno</b>	
<b>Correcta información del alumno cuando busque su información.</b>	
Resultados Esperados	
<b>Corregir errores posibles al generar reportes</b>	
Comentarios	
Al buscar al alumno debe ser mediante su cedula de identidad	
La información del alumno no debe ser alterada por el sistema.	

**Tabla 20**

*Descripción de pruebas de unidad pruebas de código PRU\_UNI03*

Identificador de la Prueba:

PRU\_UNI03

Método a Probar	Pruebas de Código – Compilación
Objetivo de la Prueba	Evaluar los eventos de compilación del sistema
Datos de Entrada:	
<b>Compilación proceso paso a paso.</b>	
Resultados Esperados	
<b>Mantener el sistema en ejecución, corregir los errores al compilar.</b>	
Comentarios	
<b>Los procesos no deben saltarse los pasos que son.</b>	

**Tabla 21**

*Descripción de pruebas de unidad almacenamiento datos PRU\_UNI04*

Identificador de la Prueba:

PRU\_UNI04

Método a Probar

**Almacenamiento de datos en la base**

Objetivo de la Prueba

Realizar una evaluación de los datos ingresados en la base de datos

Datos de Entrada:

**Los datos deben estar en la tabla correcta**

**Las tablas deben tener los campos necesarios para los procesos**

Resultados Esperados

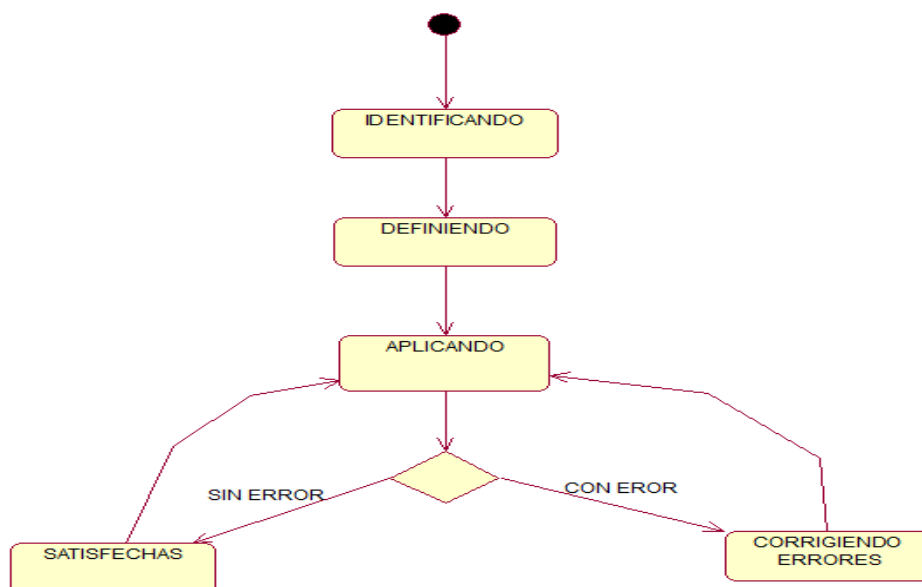
**Obtener datos consistentes y coherentes.**

Comentarios

**Los datos no deben irse a otras tablas para no tener pérdida de información**

**Las tablas no deben tener campos sin utilizar.**

**5.04. Especificación de pruebas de aceptación**



*Figura 20: Diagrama de pruebas de aceptación.*

**Tabla 22**
*Detalle prueba de aceptación PRU\_ACE01 usuarios*

Identificador de la Prueba:	PRU_ACE01
Caso de Uso	<b>Usuarios CU001</b>
Tipo de Usuario	<b>Administrador</b>
Objetivo de la Prueba	Probar el funcionamiento del proceso general de Usuarios.
Secuencia de Eventos	
<b>Login de usuario, ingresar Usuarios, Ejecutar cambios Eliminar.</b>	
Resultados Esperados	
<b>Que no tenga problemas con los eventos citados</b>	
Comentarios	
<b>Se realiza las pruebas ingresando nuevos usuarios y proporcionando mantenimiento a la tabla de usuarios.</b>	
Estado Aceptado/No aceptado	
<b>Aceptado</b>	

**Tabla 23**
*Detalle prueba de aceptación PRU\_ACE02 notas*

Identificador de la Prueba:	PRU_ACE02
Caso de Uso	<b>Notas CU002</b>
Tipo de Usuario	<b>Administrador</b>
Objetivo de la Prueba	Probar el funcionamiento del proceso de ingreso de notas.
Secuencia de Eventos	
<b>Login de usuario, ingresar notas, Eliminar, Guardar.</b>	
Resultados Esperados	
<b>Que no tenga inconsistencias con respecto a guardados de notas.</b>	
Comentarios	
<b>Se realiza las pruebas en el sistema ingresando nuevas notas a un estudiante.</b>	
Estado Aceptado/No aceptado	
<b>Aceptado</b>	

## 5.05. Especificación de pruebas de carga

**Tabla 24**

*Detalle de un tipo de prueba de carga más baja*

Identificador de la Prueba:	PRCA01
Tipo de Prueba	Prueba normal (Prueba de Carga)
Objetivo de la Prueba	Establecer los tiempos de respuesta cuando sólo un usuario está conectado a la aplicación.
Descripción:	
<b>Esta prueba establece medir el software entregado.</b>	
Resultados Esperados	
<b>Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta.</b>	
Comentarios	
No hubo complicaciones.	

**Tabla 25**

*Detalle de un tipo de prueba de carga con un número mínimo de usuarios*

Identificador de la Prueba:	PRCA02
Tipo de Prueba	<b>Prueba con número mínimo de usuarios</b>
Objetivo de la Prueba	Conocer si los procesos se están generando con normalidad y sin problemas.
Descripción:	
<b>Se realizan las pruebas del sistema con el número de usuarios mínimos.</b>	
Resultados Esperados	
<b>Validar la funcionalidad del sistema con un mínimo de usuarios logueados.</b>	
Comentarios	
<b>Problemas el login de algunas no funciona por motivos de cedula mal validada</b>	

**Tabla 26**

*Detalle de un tipo de prueba de carga con un número máximo de usuarios*

Identificador de la Prueba:

PRCA03

Tipo de Prueba

**Prueba con número máximo de usuarios**

Objetivo de la Prueba

Establecer los tiempos de respuesta cuando una gran cantidad de usuarios están conectados a la aplicación.

Descripción:

**Se realizan las pruebas del sistema con el número de usuarios máximo concurrentes establecido**

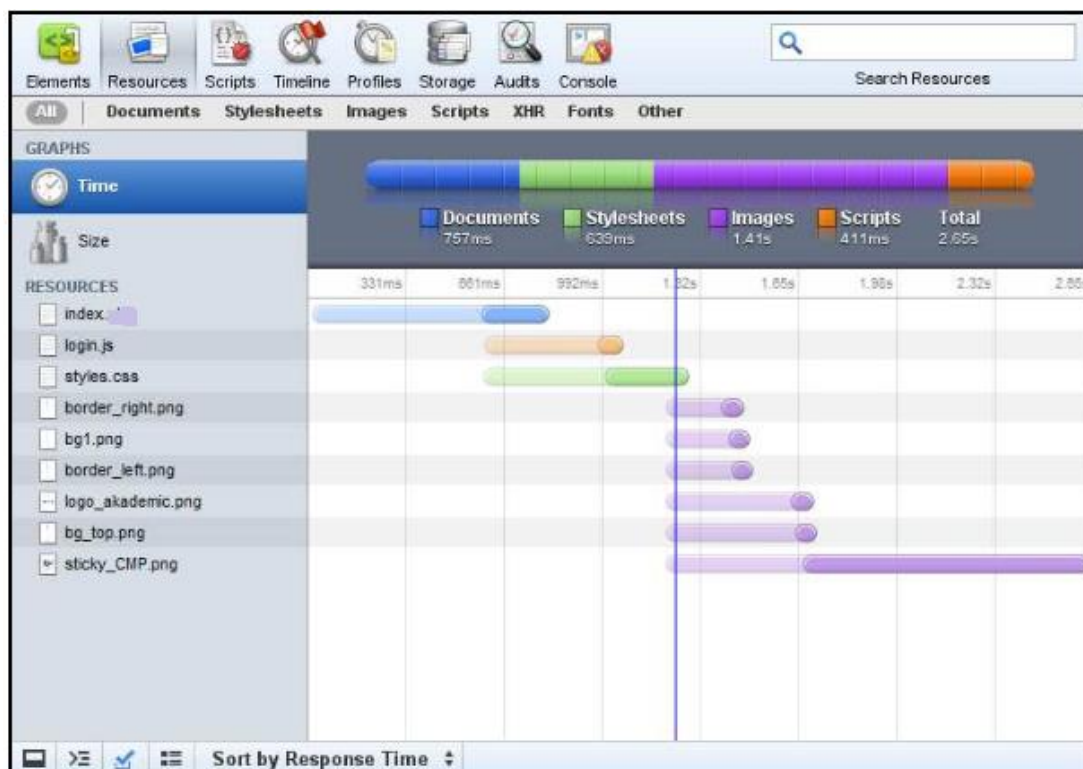
Resultados Esperados

**Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta cuando tengamos una gran cantidad de usuarios conectados en el Sistema.**

Comentarios

**Los problemas notados fueron que no hay buena respuesta de tiempo con muchos usuarios**

### Página de inicio



*Figura 21: Especificación de pruebas de aceptación*

## Ingreso de información

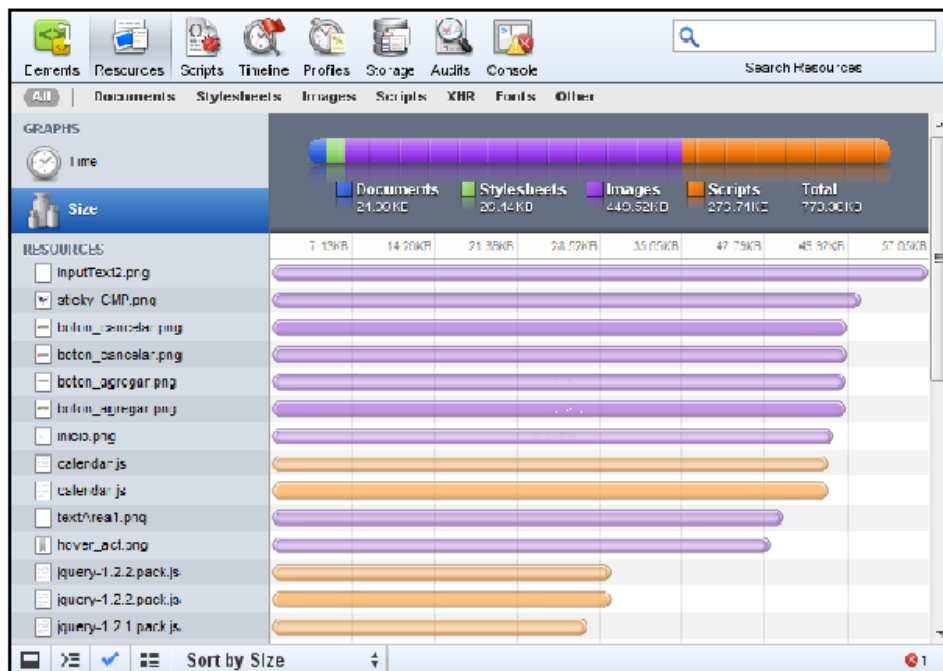


Figura 22: Especificación de pruebas de aceptación 2

### 5.06. Configuración del Ambiente mínima/ideal

#### Requerimientos Mínimos

#### Requerimientos de Hardware

#### Equipo Servidor

- Procesador: Intel Core 2 Dúo de 2 GHz ó 3Ghz
- Memoria RAM 4 GB. La cantidad de memoria está establecida para 15 usuarios usando el sistema simultáneamente.
- Disco Duro con 100 Gb LIBRES para datos

#### Equipo Cliente

- Procesador: Intel Core 2 Dúo de 2GHz.
- Memoria RAM 2 Gb.
- Disco Duro: 50 Gb de memoria libre son suficientes para alojar archivos temporales y reportes, además para instalación de navegadores Web

---

## Requerimientos de Software

### Equipo Servidor

#### Sistemas Operativos

- Sistema Operativo Windows Server 2008 o superior

#### Manejadores de Base de Datos

- MySQL 5.1 o superior

#### Paquete de Aplicaciones

Office 2007 para visualización de reportes y estadísticas.

### Requerimientos recomendados

Los requerimientos recomendados listados a continuación determinarán un buen funcionamiento del sistema, en la ejecución del mismo y en los procesos que realiza.

### Hardware

- Procesador: Intel Core i3 de 2GHz.
- Memoria RAM: 4 Gb.
- Disco Duro: 50 Gb de memoria libre son suficientes para alojar archivos temporales y reportes, además para instalación de navegadores Web.
- Pantalla de 17 pulgadas para una buena visualización.

### Software

- Sistema operativo: Windows 7 o superior.
- Navegador Web: Firefox o Google Chrome.

---

## Capítulo VI: Aspectos Administrativos

### 6.01 Recursos

#### Recursos Humanos

- Promotor del proyecto.- Carlos Cevallos
- Tutor. Ing. Hugo Heredia Msc.
- Lector Ing. Diana Terán

#### Recursos Materiales

- Computador
- Hojas de Papel Bond
- Servicios Básicos
- Servicios de Internet
- Alimentación
- Transporte
- Artículos de Oficina

#### Recurso Técnico.

- C#
- Sql Server 2008
- Office



## 6.02. Presupuesto.

**Tabla 27**

*Detalle de Gastos realizados en el Proyecto.*

<b>PRESUPUESTO DE GASTOS</b>			
<b>RUBROS</b>	<b>CANTIDAD</b>	<b>PRECIO UNITARIO</b>	<b>SUB TOTAL</b>
<b>1. Bienes</b>			
Hojas	1000	\$ 0.02	\$ 20
Cartuchos	2	\$ 35	\$ 70
Esferos	2	\$ 0.35	\$ 0.70
Impresiones B/N	900	\$0.10	\$ 90
Impresiones Color	100	\$0.25	\$ 25
<b>TOTAL</b>			<b>\$ 205.70</b>
<b>2. Servicios</b>			
Movilidad			\$ 80
Impresiones			\$ 60
Teléfono Celular			\$ 25
Luz			\$ 30
Agua			\$ 25
<b>TOTAL</b>			<b>\$ 220</b>
<b>RUBRO TOTAL</b>			<b>\$ 425.70</b>

---

### 6.03. Cronograma

El cronograma es la transcripción a tiempos de los procesos y acciones para llevar a cabo un proyecto. En él se establece cuánto tiempo va a costar a la organización que sus recursos lleven a cabo cada proceso. Además, sirve de guía para establecer el grado de avance en la consecución de objetivos tomando en cuenta las restricciones y las incertidumbres.

Comprende la realización de toda la secuencia lógica para hacer realidad los resultados.

#### *Ventajas de la planificación del cronograma de un proyecto*

La utilización de esta herramienta tan útil en el diseño de proyectos beneficia tanto a la organización como al personal que trabaja en el mismo. Aunque la responsabilidad, el compromiso y la motivación son esenciales para cumplir los objetivos de la empresa, la utilización de cronogramas hará de la planificación algo visual y accesible para todo el equipo.

Algunas de las ventajas que puede aportarte el uso de cronogramas:

- Determinar todas las actividades necesarias para hacer realidad los objetivos.
- Priorizar para determinar un orden lógico.
- Establecer los tiempos y asignar responsabilidades en el equipo de trabajo.
- Analizar qué recursos se necesitan para realizar la actividad.
- Permite estar seguro de que se está avanzando en la consecución de metas.
- Análisis de si las actividades.

Véase Anexo A.05

---

## Capítulo VII: Conclusiones y Recomendaciones

### 7.01. Conclusiones

El Proyecto de Tesis planteado nos brinda la oportunidad de desarrollar una Aplicación Web de Gestión Académica, así como implementar los conocimientos en el análisis y desarrollo de Software adquiridos a lo largo de toda nuestra carrera, profundizando en la investigación.

Se ha cumplido satisfactoriamente el desarrollo de este proyecto, permitiendo así a las instituciones de particulares de Nivel Medio Bachillerato.

Una de sus principales ventajas es la aplicación está orientada a la web para la Gestión Académica, bastara que su ordenador tenga acceso a Internet y cualquier navegador actual, la accesibilidad al Sistema Escolástico será inmediata basta que el usuario tenga su clave y contraseña.

El diseño cuenta con una interface amigable e interactiva de acuerdo a los requerimientos indicados por los beneficiarios.

### 7.02. Recomendaciones

- Para un correcto funcionamiento del proceso del sistema, todos los involucrados del sistema deben registrarse en la aplicación web, mediante su registro le permitirá a los usuarios ver la información necesaria según sus necesidades.
- La administración del sistema debe tener conocimientos en informática, ya que realizara cambios en el sistema según las necesidades que se requiera.
- Dar charla a los alumnos y padres de familia acerca de sistema y sus beneficios.
- Realizar actualizaciones de datos periódicamente.

## ANEXOS

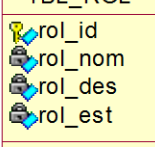
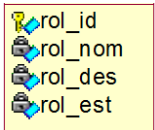

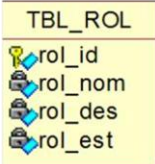
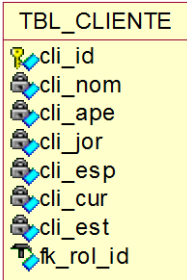
### A.01 Matriz de Involucrados.

ACTORES INVOLUCRADOS	INTERES SOBRE EL PROBLEMA CENTRAL	PROBLEMA PERCIBIDO	RECURSOS Y CAPACIDADES	INTERESES SOBRE EL PROYECTO	CONFLICTO POTENCIAL
<b>Docente</b>	Mejorar el ingreso de las notas correctas de cada alumno	Equivocación en el registro de notas obtenidas	Recurso Humano Recurso Tecnológico	Mejorar el ingreso el proceso de ingreso de notas	Falta de capacitación para el manejo del sistema
<b>Secretaria</b>	Brindar una mejor ayuda en el pedido de información de notas.	Demora de tiempo en pedido de información de alumnos y docentes	Recurso Humano Recurso Tecnológico	Va agilizar y mejorar el tiempo en la consulta de información	Falta de capacitación para el manejo del sistema
<b>Alumnos</b>	Obtener sus datos y notas correctas	Información errónea de cada alumno	Recurso Tecnológico	Tener un proceso en el cual se pueda consultar más rápido sus notas	Falta de capacitación para el manejo del sistema

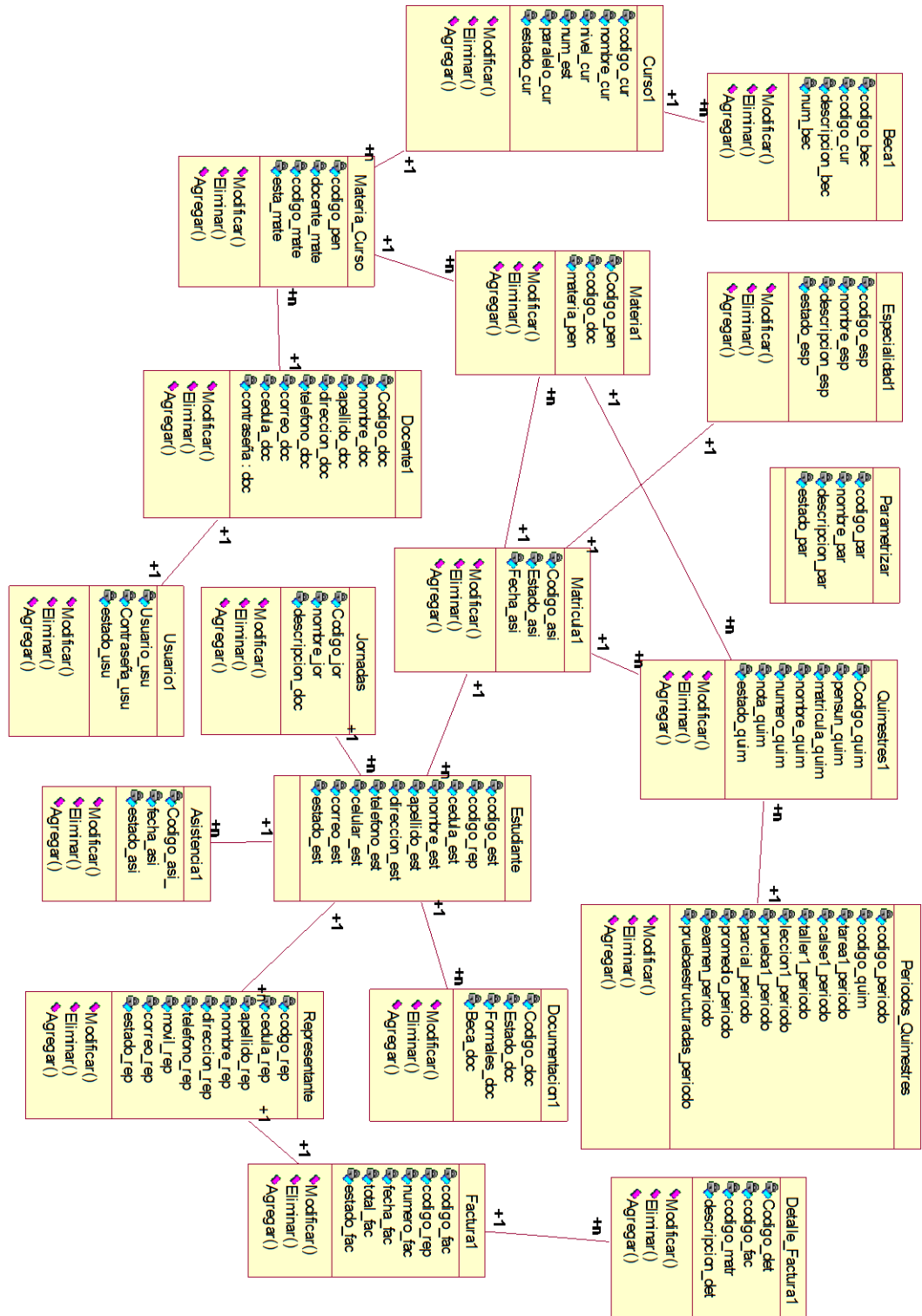
## A.02 Matriz de Alternativas.

Objetivos	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Mejorar los resultados en el cumplimiento de la programación	4	3	2	3	2	14	Media Alta
Optimizar el tiempo de respuesta.	4	2	3	3	3	15	Media Alta
Agilizar el proceso de registro de notas.	4	3	4	3	2	16	Media Alta
Controlar todos los registros de notas de los estudiantes.	4	4	2	3	2	15	Media Alta
Informar al representante de las notas de su representado.	4	2	1	2	2	11	Media Baja
<b>TOTAL</b>	<b>20</b>	<b>14</b>	<b>12</b>	<b>14</b>	<b>11</b>	<b>71</b>	

### A.03 Estándares para el Diseño de Clases

Nombre	Gráfico	Está
Clase		<p>El nombre de las clases se iniciara con el sufijo "TBL" seguido de un el nombre de la clase. Ej.: "TBL_ROL".</p> <p>El mismo se redactara en singular.</p>
Atributos		<p>Se tomara las tres primeras letras de la clase seguida de un guion bajo con el nombre del atributo, se recomienda escribir con letras minúsculas.</p> <p>Ej.: "rol_id".</p>
Métodos		<p>Las operaciones o métodos se redactaran en minúsculas y especificando la operación.</p> <p>Ej.: "guardar ()".</p>
Primary Key		<p>El atributo que se determina como primary key siempre estará marcado con una llave dorada alado, toda tabla deberá tener una llave primaria.</p>
Foreing Key		<p>EL foreing key se representa antecediendo el nombre del atributo con el sufijo "FK", seguido de un "_" y con el respectivo nombre.</p> <p>Ej.: fk_rol_id</p> <p>Como tal se muestra la herencia entre las tablas: TBL_ROL Y TBL_CLIENTE</p>

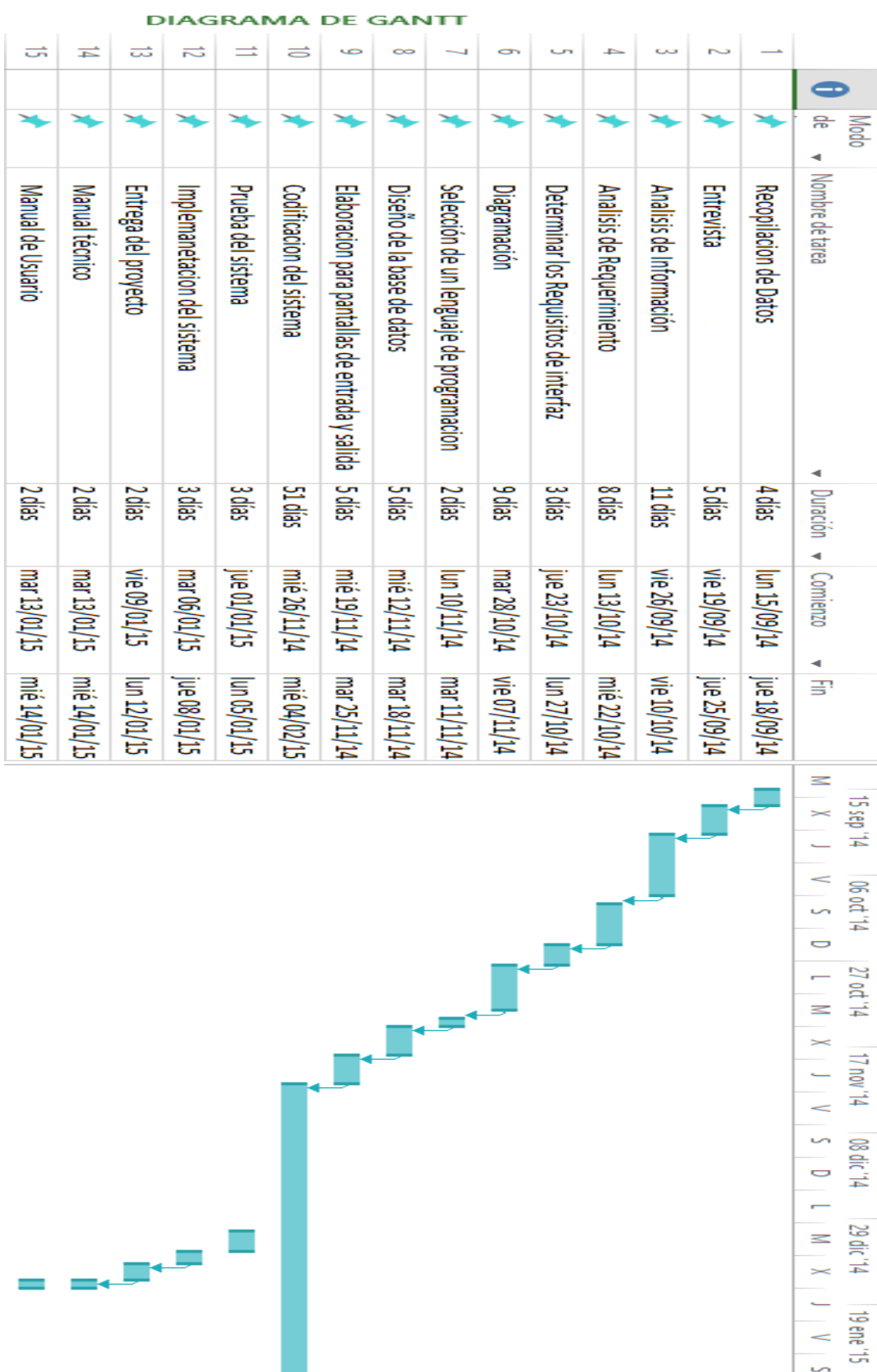
## A.04 Diagrama de Clases.







## A. 06 Cronograma



## A. 07 Script de la Base de Datos

```
CREATE DATABASE MATRICULACION
USE [MATRICULACION]
GO
/***** Object: Table [dbo].[USUARIO] Script Date: 03/24/2015
20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[USUARIO] (
    [USUARIO] [nchar] (50) NULL,
    [CONTRASENA] [nchar] (50) NULL,
    [ESTADO] [nchar] (10) NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[CURSOS] Script Date: 03/24/2015
20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[CURSOS] (
    [CUR_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [CUR_NOMBRE] [varchar] (150) NULL,
    [CUR_NIVEL] [varchar] (150) NULL,
    [CUR_NUM_ESTUDIANTES] [varchar] (150) NULL,
    [CUR_PARALELO] [varchar] (150) NULL,
    [CUR_ESTADO] [varchar] (50) NULL,
    CONSTRAINT [PK_CURSOS] PRIMARY KEY NONCLUSTERED
(
    [CUR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[DOCENTE] Script Date: 03/24/2015
20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DOCENTE] (
    [DOCEN_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [DOCEN_NOMBRE] [varchar] (150) NULL,
    [DOCEN_APELLIDO] [varchar] (150) NULL,
    [DOCEN_DIRECCION] [varchar] (150) NULL,
    [DOCEN_TELEFONO] [varchar] (20) NULL,
    [DOCEN_CORREO] [varchar] (150) NULL,
    [DOCEN_CEDULA] [varchar] (13) NULL,
```

```
[DOCEN_CONTRASENA] [varchar] (20) NULL,
CONSTRAINT [PK_DOCENTE] PRIMARY KEY NONCLUSTERED
(
    [DOCEN_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[JORNADA]      Script Date: 03/24/2015
20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[JORNADA] (
    [JOR_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [JOR_NOMBRE] [varchar] (150) NULL,
    [JOR_DESCRIPCION] [varchar] (150) NULL,
    CONSTRAINT [PK_JORNADA] PRIMARY KEY NONCLUSTERED
(
    [JOR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[ESPECIALIDAD]  Script Date:
03/24/2015 20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ESPECIALIDAD] (
    [ESP_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [ESP_NOMBRE] [varchar] (150) NULL,
    [ESP_DESCRIPCION] [varchar] (150) NULL,
    [ESP_ESTADO] [varchar] (20) NULL,
    CONSTRAINT [PK_ESPECIALIDAD] PRIMARY KEY NONCLUSTERED
(
    [ESP_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[REPRESENTANTE]  Script Date:
03/24/2015 20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[REPRESENTANTE] (
    [REP_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [REP_CEDULA] [varchar](13) NULL,
    [REP_APELLIDO] [varchar](100) NULL,
    [REP_NOMBRE] [varchar](100) NULL,
    [REP_DIRECCION] [varchar](150) NULL,
    [REP_TELEFONO] [varchar](20) NULL,
    [REP_MOVIL] [varchar](20) NULL,
    [REP_CORREO] [varchar](150) NULL,
    [REP_ESTADO] [varchar](20) NULL,
    CONSTRAINT [PK_REPRESENTANTE] PRIMARY KEY NONCLUSTERED
(
    [REP_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PARAMETRIZAR]      Script Date:
03/24/2015 20:23:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PARAMETRIZAR] (
    [CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [NOMBRE] [varchar](150) NULL,
    [DESCRIPCION] [varchar](150) NULL,
    [ESP_ESTADO] [varchar](20) NULL,
    CONSTRAINT [PK_PARAMETRIZAR] PRIMARY KEY NONCLUSTERED
(
    [CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: View [dbo].[representanteView]    Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[representanteView]
as
Select
r.REP_CEDULA,r.REP_APELLIDO,r.REP_NOMBRE,r.REP_TELEFONO,r.REP_MOVIL,
r.REP_DIRECCION,r.REP_CORREO  from Representante r
GO
/***** Object: Table [dbo].[BECA]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
```

```
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[BECA] (
    [BEC_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [CUR_CODIGO] [numeric](30, 0) NULL,
    [BEC_DESCRIPCION] [varchar](150) NULL,
    [BEC_NUM] [int] NULL,
    CONSTRAINT [PK_BECA] PRIMARY KEY NONCLUSTERED
(
    [BEC_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[FACTURA]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[FACTURA] (
    [FAC_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [REP_CODIGO] [numeric](30, 0) NOT NULL,
    [FAC_NUMERO] [int] NULL,
    [FAC_FECHA] [datetime] NULL,
    [FAC_TOTAL] [decimal](8, 3) NULL,
    [FAC_ESTADO] [varchar](20) NULL,
    CONSTRAINT [PK_FACTURA] PRIMARY KEY NONCLUSTERED
(
    [FAC_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_FACTURA_RELATIONS_REPRESEN_FK] ON
[dbo].[FACTURA]
(
    [REP_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: Table [dbo].[MATERIA]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
```

```
GO
CREATE TABLE [dbo].[MATERIA] (
    [PEN_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [DOCEN_CODIGO] [numeric](30, 0) NULL,
    [PEN_MATERIA] [varchar](150) NULL,
    CONSTRAINT [PK_MATERIA] PRIMARY KEY NONCLUSTERED
(
    [PEN_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_MATERIA_RELATIONS_DOCENTE_FK] ON
[dbo].[MATERIA]
(
    [DOCEN_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: Table [dbo].[ESTUDIANTE]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ESTUDIANTE] (
    [EST_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [REP_CODIGO] [numeric](30, 0) NULL,
    [EST_CEDULA] [varchar](13) NULL,
    [EST_NOMBRE] [varchar](100) NULL,
    [EST_APELLIDO] [varchar](100) NULL,
    [EST_DIRECCION] [varchar](150) NULL,
    [EST_TELEFONO] [varchar](20) NULL,
    [EST_CELULAR] [varchar](20) NULL,
    [EST_CORREO] [varchar](150) NULL,
    [EST_ESTADO] [varchar](20) NULL,
    CONSTRAINT [PK_ESTUDIANTE] PRIMARY KEY NONCLUSTERED
(
    [EST_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_ESTUDIAN_RELATIONS_REPRESEN_FK] ON
[dbo].[ESTUDIANTE]
(
    [REP_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
```

```
GO
/***** Object: Table [dbo].[DETALLE_FACTURA]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DETALLE_FACTURA] (
    [DET_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [FAC_CODIGO] [numeric](30, 0) NULL,
    [MATR_CODIGO] [numeric](30, 0) NULL,
    [DET_DESCRIPCION] [varchar](150) NULL,
    CONSTRAINT [PK_DETALLE_FACTURA] PRIMARY KEY NONCLUSTERED
(
    [DET_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_DETALLE__RELATIONS_FACTURA_FK] ON
[dbo].[DETALLE_FACTURA]
(
    [FAC_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FK_DETALLE__RELATIONS_MATRICUL_FK] ON
[dbo].[DETALLE_FACTURA]
(
    [MATR_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: View [dbo].[listarEstudiantesAllMatricula]
Script Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[listarEstudiantesAllMatricula]
as
select * from ESTUDIANTE E WHERE E.EST_ESTADO='Activo'
GO
/***** Object: View [dbo].[facturaDescendente]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[facturaDescendente]
as
select top (1) F.* from FACTURA F order by FAC_CODIGO desc
```

```
GO
/***** Object:  Table [dbo].[DOCUMENTACION]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DOCUMENTACION] (
    [DOC_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [EST_CODIGO] [numeric] (30, 0) NULL,
    [DOC_FORMALES] [varchar] (150) NULL,
    [DOC_BECA] [varchar] (150) NULL,
    CONSTRAINT [PK_DOCUMENTACION] PRIMARY KEY NONCLUSTERED
(
    [DOC_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object:  Table [dbo].[MATRICULA]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[MATRICULA] (
    [MATR_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [JOR_CODIGO] [numeric] (30, 0) NULL,
    [ESP_CODIGO] [numeric] (30, 0) NULL,
    [CUR_CODIGO] [numeric] (30, 0) NULL,
    [EST_CODIGO] [numeric] (30, 0) NULL,
    [MATR_FECHA_INCRIPCION] [datetime] NULL,
    [MATR_FECHA_PAGO] [datetime] NULL,
    [MATR_TIPO_MATRICULA] [varchar] (150) NULL,
    [MATR_COSTO] [money] NULL,
    [MATR_ESTADO] [varchar] (20) NULL,
    CONSTRAINT [PK_MATRICULA] PRIMARY KEY NONCLUSTERED
(
    [MATR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_MATRICUL_RELATIONS_CURSOS_FK] ON
[dbo].[MATRICULA]
(
    [CUR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
```



```
GO
CREATE NONCLUSTERED INDEX [FK_MATRICUL_RELATIONS_ESPECIAL_FK] ON
[dbo].[MATRICULA]
(
    [ESP_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FK_MATRICUL_RELATIONS_ESTUDIAN_FK] ON
[dbo].[MATRICULA]
(
    [EST_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FK_MATRICUL_RELATIONS_JORNADA_FK] ON
[dbo].[MATRICULA]
(
    [JOR_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: View [dbo].[MATERIA_DOCENTE]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[MATERIA_DOCENTE]
AS
select D.DOCEN_APELLIDO,D.DOCEN_NOMBRE,M.PEN_MATERIA from MATERIA M,
DOCENTE D WHERE M.DOCEN_CODIGO=D.DOCEN_CODIGO
GO
/***** Object: Table [dbo].[MATERIA_CURSO]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[MATERIA_CURSO] (
    [PEN_CODIGO] [numeric](30, 0) NOT NULL,
    [CUR_CODIGO] [numeric](30, 0) NOT NULL,
    [CUR_ESTADO] [varchar](50) NULL,
    [CUR_DIA] [varchar](50) NULL,
    [CUR_HORA_INI] [int] NULL,
    [CUR_HORA_FIN] [int] NULL,
    CONSTRAINT [PK_MATERIA_CURSO] PRIMARY KEY NONCLUSTERED
(
    [PEN_CODIGO] ASC,
    [CUR_CODIGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
```

```
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_MATERIA__RELATIONS_CURSOS_FK] ON
[dbo].[MATERIA_CURSO]
(
    [CUR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FK_MATERIA__RELATIONS_MATERIA_FK] ON
[dbo].[MATERIA_CURSO]
(
    [PEN_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: Table [dbo].[ASISTENCIA]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ASISTENCIA] (
    [ASI_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [EST_CODIGO] [numeric] (30, 0) NULL,
    [ASI_FECHA] [datetime] NULL,
    [ASI_ESTADO] [varchar] (20) NULL,
    CONSTRAINT [PK_ASISTENCIA] PRIMARY KEY NONCLUSTERED
(
    [ASI_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [FK_ASISTENC_RELATIONS_ESTUDIAN_FK] ON
[dbo].[ASISTENCIA]
(
    [EST_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: View [dbo].[ultimaMatriculaReporte]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[ultimaMatriculaReporte]
```

```
as
select TOP (1) M.*, R.REP_CODIGO, DT.DET_CODIGO FROM MATRICULA
M, ESTUDIANTE E, REPRESENTANTE R, DETALLE_FACTURA DT WHERE
E.REP_CODIGO=R.REP_CODIGO AND M.EST_CODIGO=E.EST_CODIGO AND
DT.MATR_CODIGO=M.MATR_CODIGO order by DT.DET_CODIGO desc
GO
/***** Object: View [dbo].[ultimaMatricula]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[ultimaMatricula]
as
select TOP (1) M.*, R.REP_CODIGO FROM MATRICULA M, ESTUDIANTE E,
REPRESENTANTE R WHERE E.REP_CODIGO=R.REP_CODIGO AND
M.EST_CODIGO=E.EST_CODIGO order by M.MATR_CODIGO desc
GO
/***** Object: StoredProcedure [dbo].[spVerificarCupos]      Script
Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[spVerificarCupos]
@codCurso int
as
DECLARE @numNormal int,
        @numBeca int,
        @normalCur int,
        @becaCur int

    SELECT @numNormal=COUNT(M.CUR_CODIGO) FROM MATRICULA M WHERE
M.MATR_TIPO_MATRICULA='NORMAL' AND M.CUR_CODIGO=@codCurso
    SELECT @numBeca=COUNT(M.CUR_CODIGO) FROM MATRICULA M WHERE
M.MATR_TIPO_MATRICULA='BECA' AND M.CUR_CODIGO=@codCurso
    SELECT @normalCur=C.CUR_NUM_ESTUDIANTES, @becaCur=B.BEC_NUM FROM
CURSOS C, BECA B WHERE C.CUR_CODIGO=B.CUR_CODIGO AND
C.CUR_CODIGO=@codCurso

select (@normalCur-@numNormal) as 'NUM_NORMAL', (@becaCur-@numBeca) AS
'NUM_BECA'
PRINT @numNormal
PRINT @numBeca
PRINT @normalCur
PRINT @becaCur
GO
/***** Object: View [dbo].[ESTUDIANTESMETRICULADOS]      Script
Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[ESTUDIANTESMETRICULADOS]
AS
select
ES.EST_CEDULA, ES.EST_APELLIDO, ES.EST_NOMBRE, J.JOR_NOMBRE, E.ESP_NOMBR
E, C.CUR_NIVEL+' '+C.CUR_PARALELO AS
'CURSO', M.MATR_FECHA_INCRIPCION, M.MATR_COSTO
from MATRICULA M, JORNADA J, ESPECIALIDAD E, CURSOS C, ESTUDIANTE ES
WHERE
```

```
M.CUR_CODIGO=C.CUR_CODIGO
AND M.ESP_CODIGO=E.ESP_CODIGO
AND M.EST_CODIGO=ES.EST_CODIGO
AND M.JOR_CODIGO= J.JOR_CODIGO
GO
/***** Object:  StoredProcedure
[dbo].[sp_insertar_document_estudiante]      Script Date: 03/24/2015
20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[sp_insertar_document_estudiante]
AS
BEGIN
declare @valorId int
select top(1)@valorId=E.EST_CODIGO from ESTUDIANTE E order by
E.EST_CODIGO desc
INSERT INTO [MATRICULACION].[dbo].[DOCUMENTACION]
        ([EST_CODIGO]
        , [DOC_FORMALES]
        , [DOC_BECA])
VALUES
        (@valorId, 'P', 'P')

END
GO
/***** Object:  Table [dbo].[QUIMESTRALES]      Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[QUIMESTRALES] (
    [QUI_CODIGO] [numeric] (30, 0) IDENTITY(1,1) NOT NULL,
    [PEN_CODIGO] [numeric] (30, 0) NULL,
    [MATR_CODIGO] [numeric] (30, 0) NULL,
    [QUI_NOMBRE] [varchar] (50) NULL,
    [QUI_NUMERO] [int] NULL,
    [QUI_NOTA] [decimal] (8, 2) NULL,
    [QUI_ESTADO] [varchar] (50) NULL,
    CONSTRAINT [PK_QUIMESTRALES] PRIMARY KEY NONCLUSTERED
(
    [QUI_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE NONCLUSTERED INDEX [RELATIONSHIP_18_FK] ON
[dbo].[QUIMESTRALES]
(
    [PEN_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
```

```
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [RELATIONSHIP_19_FK] ON
[dbo].[QUIMESTRALES]
(
    [MATR_CODIGO] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
/***** Object: View [dbo].[estudiantesPorDocente]    Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[estudiantesPorDocente]
as
select DISTINCT E.*,D.DOCEN_CEDULA from ESTUDIANTE E,MATRICULA
M,QUIMESTRALES Q, MATERIA MAT,DOCENTE D
WHERE E.EST_CODIGO=M.EST_CODIGO AND M.MATR_CODIGO=Q.MATR_CODIGO AND
Q.PEN_CODIGO=MAT.PEN_CODIGO
AND MAT.DOCEN_CODIGO=D.DOCEN_CODIGO
GO
/***** Object: Table [dbo].[PERIODOS_QUIMESTRE]    Script Date:
03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PERIODOS_QUIMESTRE] (
    [PERQ_CODIGO] [numeric](30, 0) IDENTITY(1,1) NOT NULL,
    [QUI_CODIGO] [numeric](30, 0) NULL,
    [TAREA4] [decimal](8, 2) NULL,
    [TAREA1] [decimal](8, 2) NULL,
    [ACT_CLASE1] [decimal](8, 2) NULL,
    [TALLER1] [decimal](8, 2) NULL,
    [LECCIONES1] [decimal](8, 2) NULL,
    [PRUEBA_ESTRUCTURADA1] [decimal](8, 2) NULL,
    [PARCIAL1] [decimal](8, 2) NULL,
    [TAREA2] [decimal](8, 2) NULL,
    [ACT_CLASE2] [decimal](8, 2) NULL,
    [TALLER2] [decimal](8, 2) NULL,
    [LECCIONES2] [decimal](8, 2) NULL,
    [PRUEBA_ESTRUCTURADA2] [decimal](8, 2) NULL,
    [PARCIAL2] [decimal](8, 2) NULL,
    [TAREA3] [decimal](8, 2) NULL,
    [ACT_CLASE3] [decimal](8, 2) NULL,
    [TALLER3] [decimal](8, 2) NULL,
    [LECCIONES3] [decimal](8, 2) NULL,
    [PRUEBA_ESTRUCTURADA3] [decimal](8, 2) NULL,
    [PARCIAL3] [decimal](8, 2) NULL,
    [PROMEDIO] [decimal](8, 2) NULL,
    [EXAMEN_QUIMESTRAL] [decimal](8, 2) NULL,
    [PROMEDIO_QUIMESTRAL] [decimal](8, 2) NULL,
    CONSTRAINT [PK_PERIODOS_QUIMESTRE] PRIMARY KEY NONCLUSTERED
(
    [PERQ_CODIGO] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
) ON [PRIMARY]  
GO  
CREATE NONCLUSTERED INDEX [FK_PERIODOS_RELATIONS_QUIMESTR_FK] ON  
[dbo].[PERIODOS_QUIMESTRE]  
(  
    [QUI_CODIGO] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,  
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON  
[PRIMARY]  
GO  
/***** Object: StoredProcedure [dbo].[sp_insertar_notas]      Script  
Date: 03/24/2015 20:23:43 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE proc [dbo].[sp_insertar_notas]  
@codCurso int ,  
@codMetricula int  
as  
begin  
declare  
@codMateria int  
declare Cursor1 Cursor  
for select PEN_CODIGO from dbo.MATERIA_CURSO C WHERE  
C.CUR_CODIGO=@codCurso  
open Cursor1  
fetch Cursor1 into @codMateria  
  
while(@@FETCH_STATUS=0)  
BEGIN  
if(@codMateria is not null)  
    BEGIN  
        INSERT INTO [MATRICULACION].[dbo].[QUIMESTRALES]  
            ([PEN_CODIGO]  
            , [MATR_CODIGO]  
            , [QUI_NOMBRE]  
            , [QUI_NUMERO]  
            , [QUI_NOTA]  
            , [QUI_ESTADO])  
VALUES (@codMateria --codpen  
        , @codMetricula --codMatr  
        , 'PRIMERO'  
        , 1  
        , 0  
        , 'REPROBADO')  
  
        INSERT INTO [MATRICULACION].[dbo].[QUIMESTRALES]  
            ([PEN_CODIGO]  
            , [MATR_CODIGO]  
            , [QUI_NOMBRE]  
            , [QUI_NUMERO]  
            , [QUI_NOTA]  
            , [QUI_ESTADO])  
VALUES (@codMateria --codpen  
        , @codMetricula --codMatr  
        , 'SEGUNDO'
```

[illegible]

```
,0
,0
,0
,0
,0
,0
,0
,0
,0
,0
,0
,0
,0)

fetch Cursor2 into @codQuimetre
end
close Cursor2
deallocate Cursor2
fetch Cursor1 into @codMateria

end
close Cursor1
deallocate Cursor1

end
GO
/***** Object:  StoredProcedure [dbo].[sp_calcularNotas]      Script
Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[sp_calcularNotas]
@codPeriodoQuimestre int
as
begin

UPDATE [MATRICULACION].[dbo].[PERIODOS_QUIMESTRE]
    SET [PARCIAL1] =
    ((TAREA1+ACT_CLASE1+TALLER1+LECCIONES1+PRUEBA_ESTRUCTURADA1)/5)
    , [PARCIAL2] =
    ((TAREA2+ACT_CLASE2+TALLER2+LECCIONES2+PRUEBA_ESTRUCTURADA2)/5)
    , [PARCIAL3] =
    ((TAREA3+ACT_CLASE3+TALLER3+LECCIONES3+PRUEBA_ESTRUCTURADA3)/5)
    WHERE PERQ_CODIGO=@codPeriodoQuimestre

UPDATE [MATRICULACION].[dbo].[PERIODOS_QUIMESTRE]
    SET [PROMEDIO] = ((PARCIAL1+PARCIAL2+PARCIAL3)/3)
    WHERE PERQ_CODIGO=@codPeriodoQuimestre

UPDATE [MATRICULACION].[dbo].[PERIODOS_QUIMESTRE]
    SET [PROMEDIO_QUIMESTRAL] =
    (PROMEDIO*0.8)+(EXAMEN_QUIMESTRAL*0.2)
    WHERE PERQ_CODIGO=@codPeriodoQuimestre

end
GO
/***** Object:  View [dbo].[CALIFICACIONESPORESTUDIANTE]      Script
Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```



```
CREATE VIEW [dbo].[CALIFICACIONESPORESTUDIANTE]
AS
select MAT.PEN_MATERIA,Q.QUI_NOMBRE, P.*, M.EST_CODIGO from
ESTUDIANTE E,MATRICULA M,QUIMESTRALES Q,PERIODOS_QUIMESTRE P,MATERIA
MAT
WHERE M.EST_CODIGO=E.EST_CODIGO AND M.MATR_CODIGO=Q.MATR_CODIGO AND
P.QUI_CODIGO=Q.QUI_CODIGO AND MAT.PEN_CODIGO=Q.PEN_CODIGO
GO
/***** Object:  StoredProcedure [dbo].[sp_procesar_Nota_inicial]
Script Date: 03/24/2015 20:23:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[sp_procesar_Nota_inicial]
as
BEGIN
DECLARE
@codMatriculaReg int,
@codCursoReg int,
@validarRegistroNotas int
SELECT
TOP (1)@codMatriculaReg=M.MATR_CODIGO,@codCursoReg=M.CUR_CODIGO FROM
MATRICULA M ORDER BY M.MATR_CODIGO DESC
PRINT @codMatriculaReg
select @validarRegistroNotas=Q.MATR_CODIGO from QUIMESTRALES Q WHERE
Q.MATR_CODIGO=@codMatriculaReg
IF(@validarRegistroNotas IS not NULL)
BEGIN

PRINT 'REGISTRO EXITOSO'
END
ELSE
BEGIN
exec sp_insertar_notas @codCursoReg,@codMatriculaReg
PRINT 'YA EXISTE EL REGISTRO'
END

end
GO
/***** Object:  ForeignKey [FK_BECA_FK_BECA_R_ESTUDIAN]      Script
Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[BECA] WITH CHECK ADD CONSTRAINT
[FK_BECA_FK_BECA_R_ESTUDIAN] FOREIGN KEY ([CUR_CODIGO])
REFERENCES [dbo].[CURSOS] ([CUR_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BECA] CHECK CONSTRAINT
[FK_BECA_FK_BECA_R_ESTUDIAN]
GO
/***** Object:  ForeignKey [FK_FACTURA_FK_FACTUR_REPRESEN]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[FACTURA] WITH CHECK ADD CONSTRAINT
[FK_FACTURA_FK_FACTUR_REPRESEN] FOREIGN KEY ([REP_CODIGO])
REFERENCES [dbo].[REPRESENTANTE] ([REP_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[FACTURA] CHECK CONSTRAINT
[FK_FACTURA_FK_FACTUR_REPRESEN]
GO
/***** Object:  ForeignKey [FK_MATERIA_FK_MATERI_DOCENTE]      Script
Date: 03/24/2015 20:23:43 *****/
```

```
ALTER TABLE [dbo].[MATERIA] WITH CHECK ADD CONSTRAINT
[FK_MATERIA_FK_MATERI_DOCENTE] FOREIGN KEY([DOCEN_CODIGO])
REFERENCES [dbo].[DOCENTE] ([DOCEN_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATERIA] CHECK CONSTRAINT
[FK_MATERIA_FK_MATERI_DOCENTE]
GO
/***** Object: ForeignKey [FK_ESTUDIAN_FK_ESTUDI_REPRESEN]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[ESTUDIANTE] WITH CHECK ADD CONSTRAINT
[FK_ESTUDIAN_FK_ESTUDI_REPRESEN] FOREIGN KEY([REP_CODIGO])
REFERENCES [dbo].[REPRESENTANTE] ([REP_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[ESTUDIANTE] CHECK CONSTRAINT
[FK_ESTUDIAN_FK_ESTUDI_REPRESEN]
GO
/***** Object: ForeignKey [FK_FACTURA_FK_DETALLE_FACTURA]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[DETALLE_FACTURA] WITH CHECK ADD CONSTRAINT
[FK_FACTURA_FK_DETALLE_FACTURA] FOREIGN KEY([FAC_CODIGO])
REFERENCES [dbo].[FACTURA] ([FAC_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[DETALLE_FACTURA] CHECK CONSTRAINT
[FK_FACTURA_FK_DETALLE_FACTURA]
GO
/***** Object: ForeignKey [FK_DOCUMENT_FK_ESTUDIANTE]      Script
Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[DOCUMENTACION] WITH CHECK ADD CONSTRAINT
[FK_DOCUMENT_FK_ESTUDIANTE] FOREIGN KEY([EST_CODIGO])
REFERENCES [dbo].[ESTUDIANTE] ([EST_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[DOCUMENTACION] CHECK CONSTRAINT
[FK_DOCUMENT_FK_ESTUDIANTE]
GO
/***** Object: ForeignKey [FK_MATRICUL_FK_MATRIC_CURSOS]      Script
Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[MATRICULA] WITH CHECK ADD CONSTRAINT
[FK_MATRICUL_FK_MATRIC_CURSOS] FOREIGN KEY([CUR_CODIGO])
REFERENCES [dbo].[CURSOS] ([CUR_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATRICULA] CHECK CONSTRAINT
[FK_MATRICUL_FK_MATRIC_CURSOS]
GO
/***** Object: ForeignKey [FK_MATRICUL_FK_MATRIC_ESPECIAL]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[MATRICULA] WITH CHECK ADD CONSTRAINT
[FK_MATRICUL_FK_MATRIC_ESPECIAL] FOREIGN KEY([ESP_CODIGO])
REFERENCES [dbo].[ESPECIALIDAD] ([ESP_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATRICULA] CHECK CONSTRAINT
[FK_MATRICUL_FK_MATRIC_ESPECIAL]
GO
/***** Object: ForeignKey [FK_MATRICUL_FK_MATRIC_ESTUDIAN]
Script Date: 03/24/2015 20:23:43 *****/
```

```
ALTER TABLE [dbo].[MATRICULA] WITH CHECK ADD CONSTRAINT
[FK_MATRICUL_FK_MATRIC_ESTUDIAN] FOREIGN KEY([EST_CODIGO])
REFERENCES [dbo].[ESTUDIANTE] ([EST_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATRICULA] CHECK CONSTRAINT
[FK_MATRICUL_FK_MATRIC_ESTUDIAN]
GO
/***** Object: ForeignKey [FK_MATRICUL_FK_MATRIC_JORNADA]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[MATRICULA] WITH CHECK ADD CONSTRAINT
[FK_MATRICUL_FK_MATRIC_JORNADA] FOREIGN KEY([JOR_CODIGO])
REFERENCES [dbo].[JORNADA] ([JOR_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATRICULA] CHECK CONSTRAINT
[FK_MATRICUL_FK_MATRIC_JORNADA]
GO
/***** Object: ForeignKey [FK_MATERIA__FK_MATERI_CURSOS]      Script
Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[MATERIA_CURSO] WITH CHECK ADD CONSTRAINT
[FK_MATERIA__FK_MATERI_CURSOS] FOREIGN KEY([CUR_CODIGO])
REFERENCES [dbo].[CURSOS] ([CUR_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATERIA_CURSO] CHECK CONSTRAINT
[FK_MATERIA__FK_MATERI_CURSOS]
GO
/***** Object: ForeignKey [FK_MATERIA__FK_MATERI_MATERIA]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[MATERIA_CURSO] WITH CHECK ADD CONSTRAINT
[FK_MATERIA__FK_MATERI_MATERIA] FOREIGN KEY([PEN_CODIGO])
REFERENCES [dbo].[MATERIA] ([PEN_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MATERIA_CURSO] CHECK CONSTRAINT
[FK_MATERIA__FK_MATERI_MATERIA]
GO
/***** Object: ForeignKey [FK_ASISTENC_FK_ASISTE_ESTUDIAN]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[ASISTENCIA] WITH CHECK ADD CONSTRAINT
[FK_ASISTENC_FK_ASISTE_ESTUDIAN] FOREIGN KEY([EST_CODIGO])
REFERENCES [dbo].[ESTUDIANTE] ([EST_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[ASISTENCIA] CHECK CONSTRAINT
[FK_ASISTENC_FK_ASISTE_ESTUDIAN]
GO
/***** Object: ForeignKey [FK_QUIMESTR_RELATIONS_MATERIA]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[QUIMESTRALES] WITH CHECK ADD CONSTRAINT
[FK_QUIMESTR_RELATIONS_MATERIA] FOREIGN KEY([PEN_CODIGO])
REFERENCES [dbo].[MATERIA] ([PEN_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[QUIMESTRALES] CHECK CONSTRAINT
[FK_QUIMESTR_RELATIONS_MATERIA]
GO
/***** Object: ForeignKey [FK_QUIMESTR_RELATIONS_MATRICUL]
Script Date: 03/24/2015 20:23:43 *****/
```

```
ALTER TABLE [dbo].[QUIMESTRALES] WITH CHECK ADD CONSTRAINT
[FK_QUIMESTR_RELATIONS_MATRICUL] FOREIGN KEY ([MATR_CODIGO])
REFERENCES [dbo].[MATRÍCULA] ([MATR_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[QUIMESTRALES] CHECK CONSTRAINT
[FK_QUIMESTR_RELATIONS_MATRICUL]
GO
/***** Object: ForeignKey [FK_PERIODOS_FK_PERIOD_QUIMESTR]
Script Date: 03/24/2015 20:23:43 *****/
ALTER TABLE [dbo].[PERIODOS_QUIMESTRE] WITH CHECK ADD CONSTRAINT
[FK_PERIODOS_FK_PERIOD_QUIMESTR] FOREIGN KEY ([QUI_CODIGO])
REFERENCES [dbo].[QUIMESTRALES] ([QUI_CODIGO])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[PERIODOS_QUIMESTRE] CHECK CONSTRAINT
[FK_PERIODOS_FK_PERIOD_QUIMESTR]
GO
```

## **A.08 Manual de Instalación.**

### **ÍNDICE GENERAL**

#### **1. Introducción**

#### **2. Requerimiento**

2.01. Requerimientos Mínimos

2.02. Requerimientos Recomendados

#### **3. Instalación de Aplicación**

3.01. Instalación de Sql Server 2008

3.02. Instalación de Visual Studio 2010

#### **4. Iniciar Sesión**

#### **5. Pantalla principal**

5.01. Menú Principal

5.02. Inicio de Sesión

#### **6. Administración Avanzada del sistema**

6.01. Menú de Administración Avanzada

6.02. Mantenimiento del sistema

6.03. Como gestiona una tabla de la Base de Datos

6.04. Agregar registros

6.05. Editar registros

6.06. Eliminar registros

#### **7. Administración básica**

7.01. Ingresar Clientes

7.02. Ingresar Equipos

## 1. Introducción

El presente manual muestra los pasos a seguir para dar uso y administrar de forma correcta a nivel de administrador y usuario, el sistema de mejoramiento de procesos académicos.

## 2. Requerimientos

Los requerimientos descritos a continuación están divididos en mínimos y recomendados, de tal manera que el uno o el otro determinaran el buen funcionamiento del sistema.

### 2.01 Requerimientos mínimos

Los requerimientos mínimos listados a continuación determinaran un funcionamiento normal del sistema pero se no se garantiza que realice los procesos a un velocidad excelente, por lo cual el sistema puede llegar a fallar en ciertos momentos.

#### Hardware

- Procesador: Intel Core 2 Dúo de 2GHz.
- Memoria RAM: 2 Gb.
- Disco Duro: 50 Gb de memoria libre son suficientes para alojar archivos temporales y reportes, además para instalación de navegadores Web.
- Pantalla de 17 pulgadas para una buena visualización

#### Software

- Sistema operativo: Windows vista o superior.
- Microsoft Office: 2010 o superior.

- Navegador Web: Firefox o Google Chrome

## **2.02 Requerimientos recomendados**

Los requerimientos recomendados listados a continuación determinarán un buen funcionamiento del sistema, en la ejecución del mismo y en los procesos que realiza.

### **Hardware**

- Procesador: Intel Core i3 de 2GHz.
- Memoria RAM: 4 Gb.
- Disco Duro: 50 Gb de memoria libre son suficientes para alojar archivos temporales y reportes, además para instalación de navegadores Web.

- Pantalla de 17 pulgadas para una buena visualización.

### **Software**

- Sistema operativo: Windows 7 o superior.
- Microsoft Office: 2010 o superior.
- Navegador Web: Firefox o Google Chrome.

### 3. Instalación de Aplicaciones

#### 3.01 Instalación de Sql Server 2008

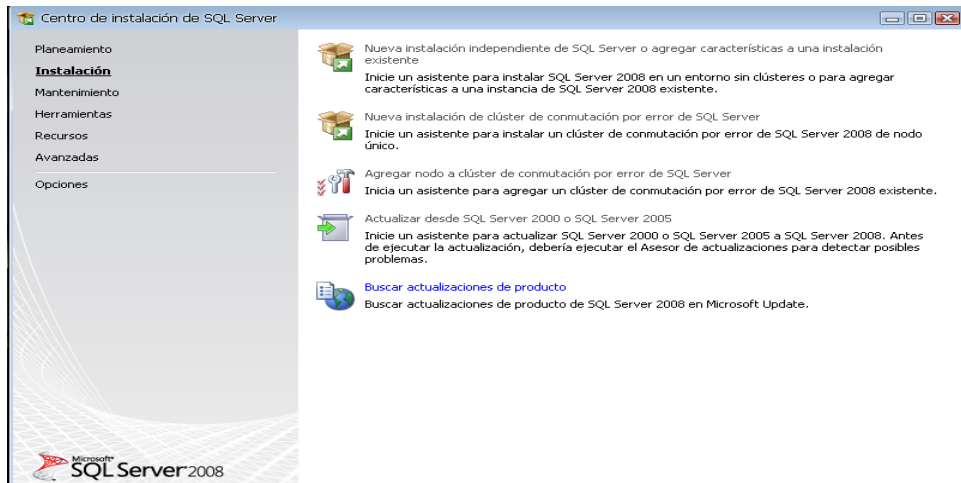


Figura 1. Ventana de instalación

Primero se ejecuta la instalación de SQL SERVER 2008 desde el CD, luego nos sale un ventana que es el asistente de la Instalación, hacemos clic en la Pestaña “Instalación” y luego en la opción “Nueva Instalación independiente de SQL Server o agregar características a una instalación existente”.

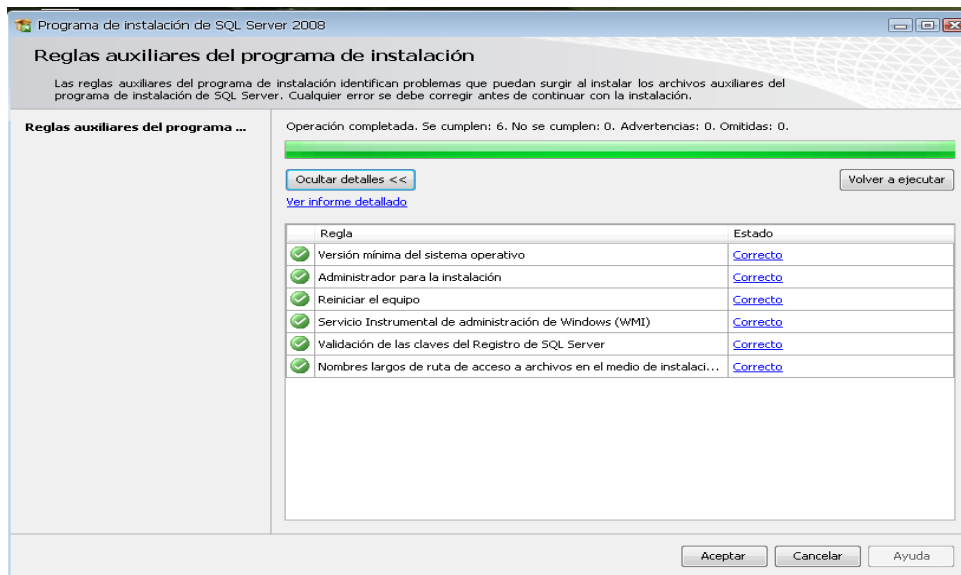


Figura 2. Ventana de reglas auxiliares

En la siguiente ventana comienza a cargar algunas Reglas auxiliares que son necesarias para la instalación de SQL Server 2008, si todos están en Estado

**Correcto** hacer clic en el botón siguiente de lo contrario si no están algunos en             
AUTOMATIZACIÓN DE LOS PROCESOS ACADÉMICOS DE INSTITUCIONES EDUCATIVAS DE NIVEL MEDIO-  
BACHILLERATO MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN EDUCATIVA: MÓDULO DE GESTIÓN  
ACADEMICA



Estado **Correcto** no se puede seguir o continuar con la instalación (para ver las reglas clic en el botón mostrar detalles).

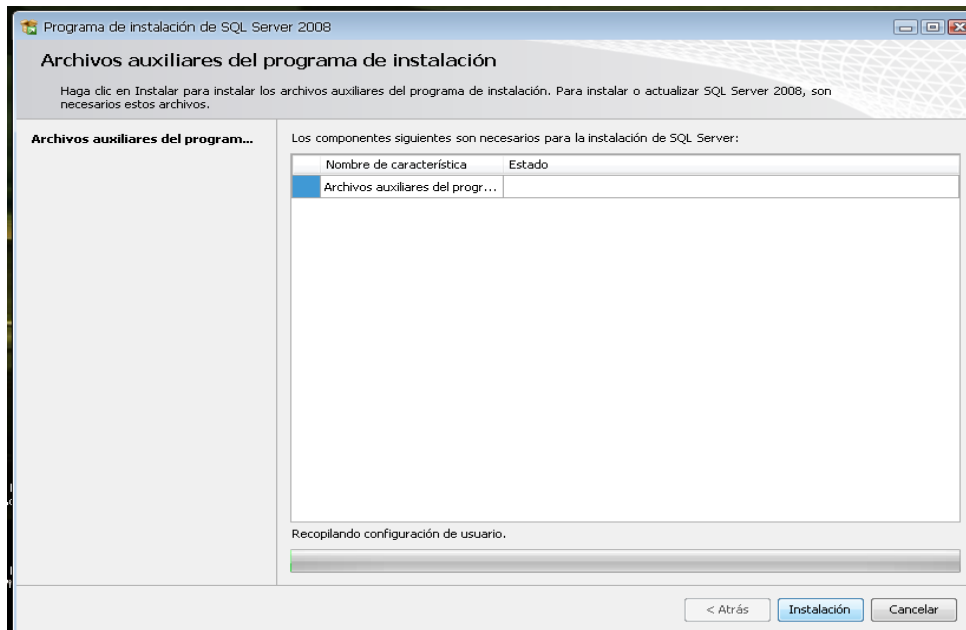


Figura 3. Ventana de archivos auxiliares

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

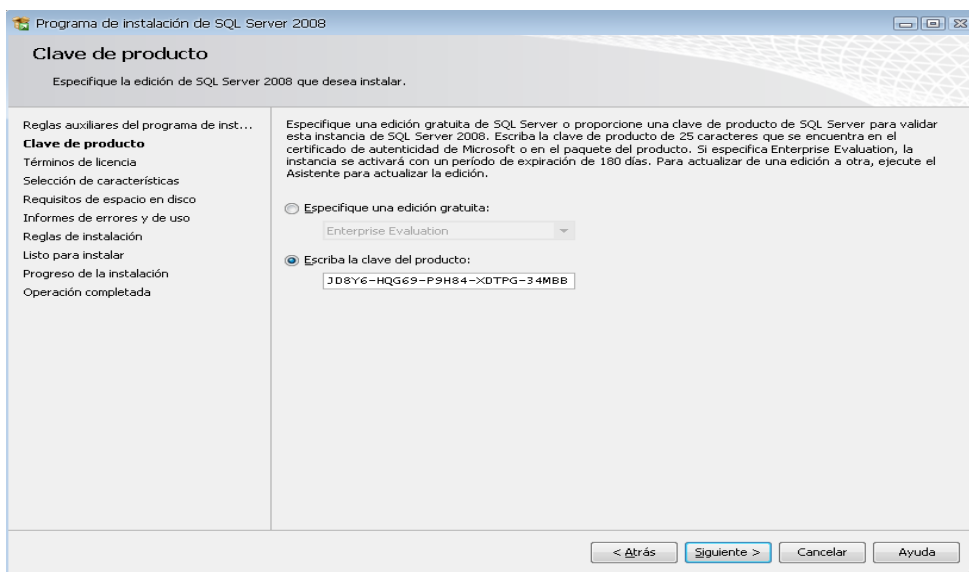


Figura 4. Ventana Clave del producto.

Luego de haber instalado los archivos auxiliares que son necesarios para la

Instalación de SQL Server 2008, la siguiente ventana que nos sale es la clave del

producto y nos dan dos opciones: “Especifique una edición gratuita” y “Escriba la clave del Producto”, por defecto sale seleccionado la segunda opción y la clave, así que solo damos clic en el botón siguiente.



Figura 5. Ventana términos de licencia.

En esta ventana es si sobre los términos de la licencia del producto, hacemos clic en la opción “Acepto los términos de licencia” y luego clic en Siguiente

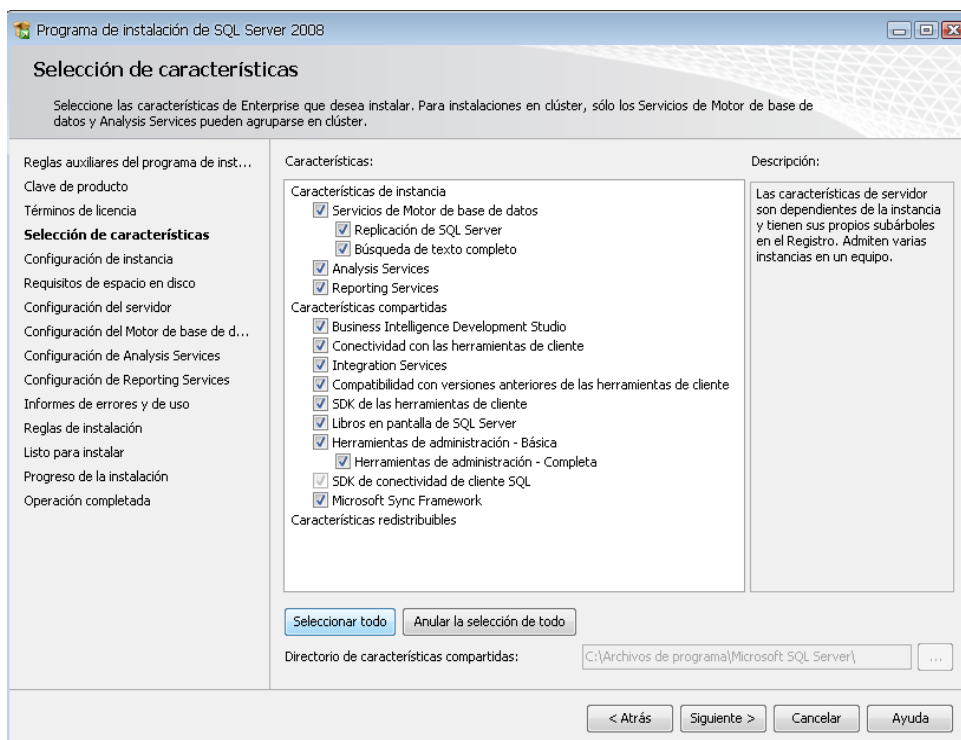


Figura 6. Ventana selección de características.

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

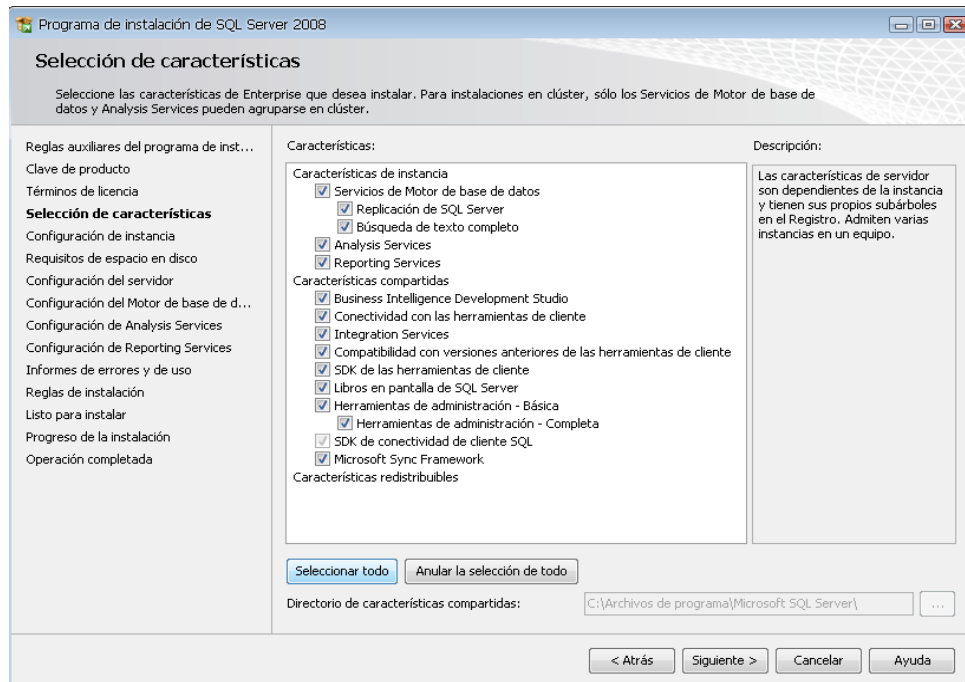


Figura 7. Ventana selección de características.

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

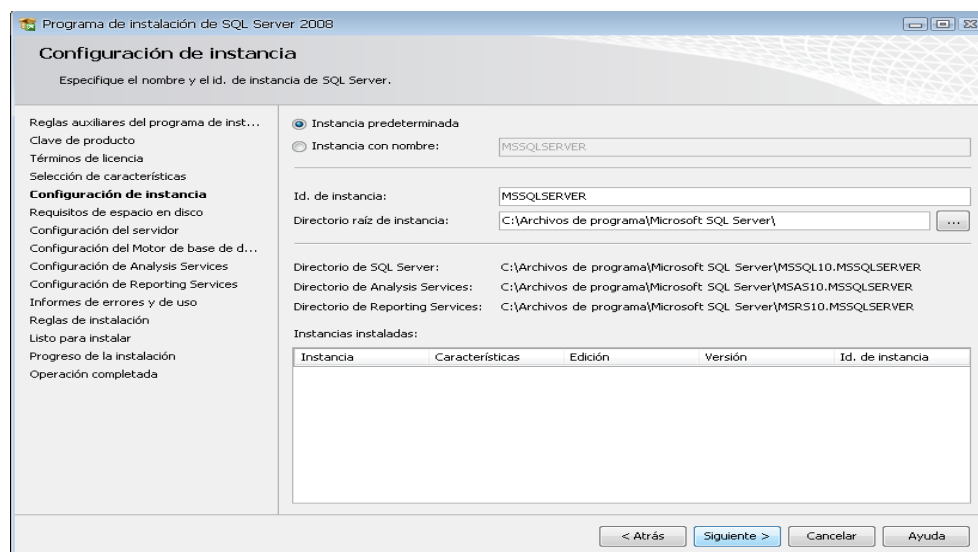


Figura 7. Ventana configuración de instancia.

Luego de haber agregado las características, la siguiente ventana a mostrar es sobre la configuración de la instancia de SQL Server 2008; es decir el nombre de la instalación, su ID y su ubicación en donde se instalará el producto. Nos dan 2

opciones: "Instancia predeterminada" y "Instancia con nombre", seleccionar la primera opción y luego clic en siguiente. En la siguiente ventana también damos click en siguiente.

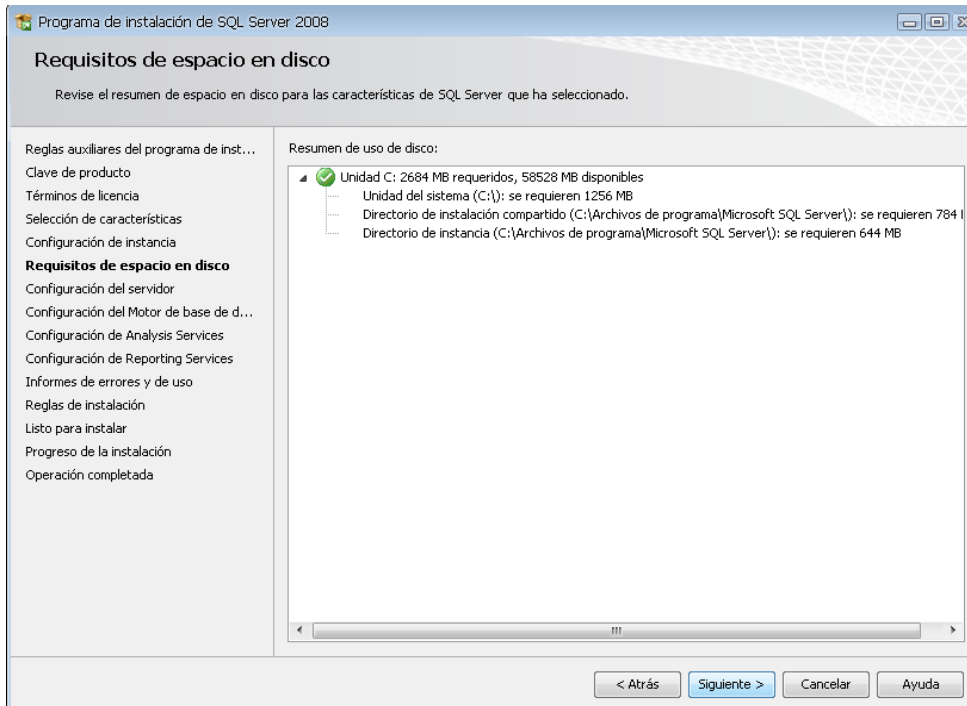


Figura 8. Ventana requisitos de espacio.

En esta ventana le damos siguiente ya que solo nos indica el espacio requerido en el disco.

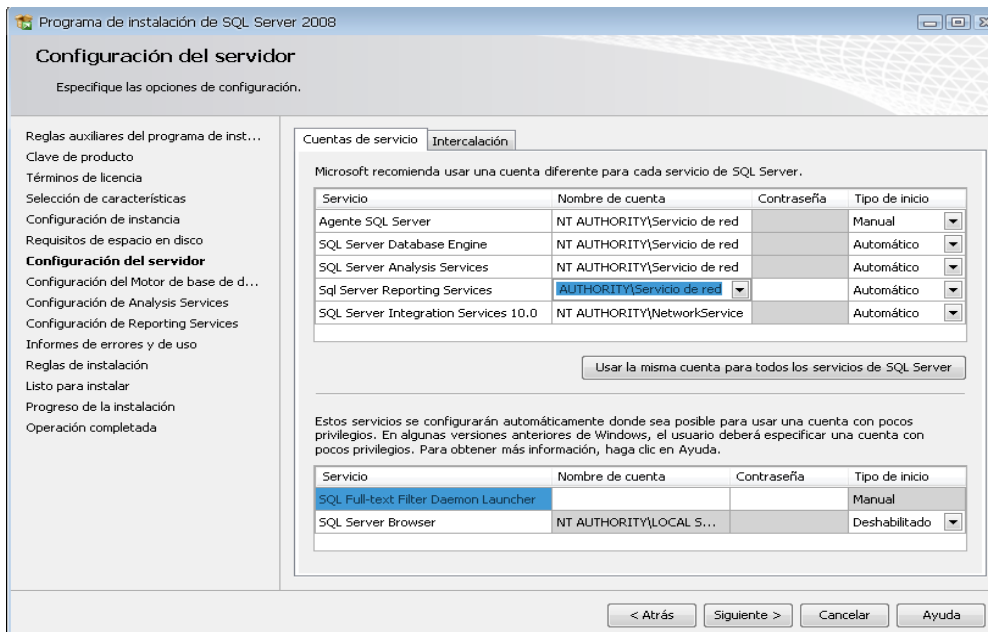


Figura 9. Ventana configuración del servidor.

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

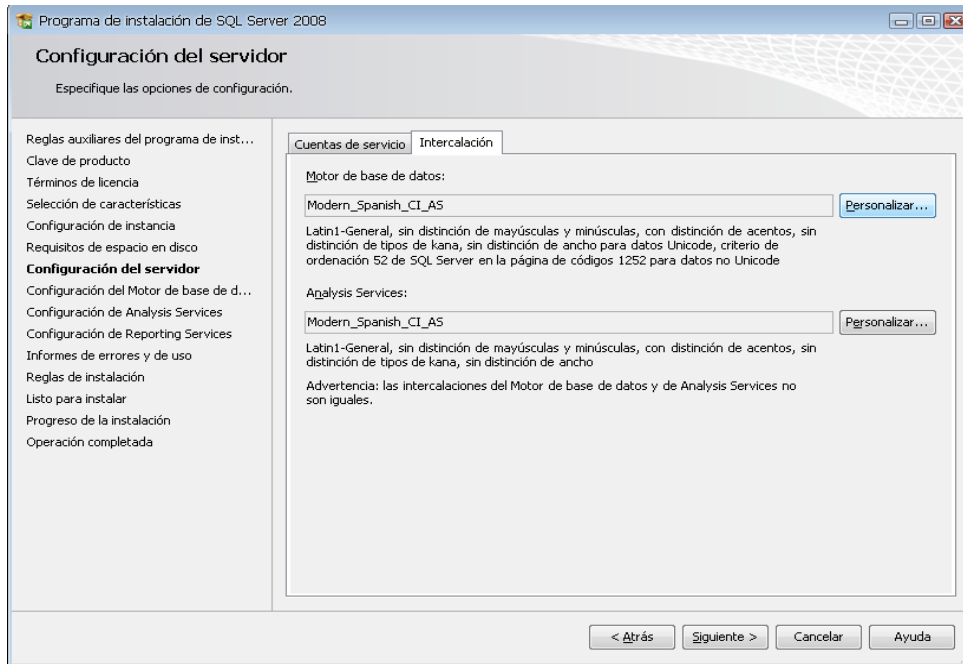


Figura 10. Ventana configuración del servidor 1.

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

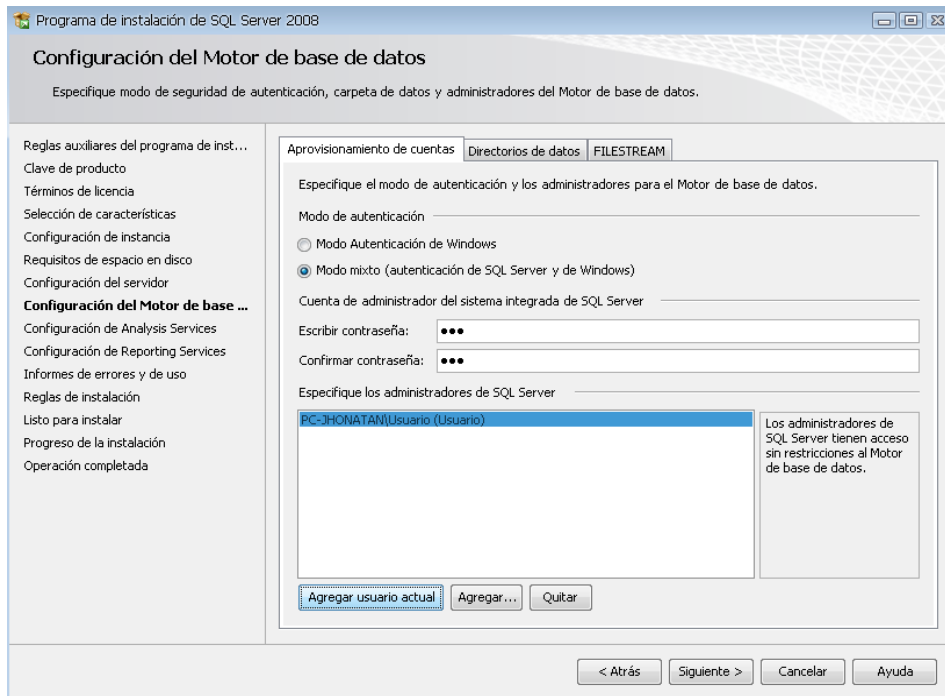


Figura 11. Ventana configuración del motor de base de datos.

Luego la siguiente ventana que nos muestra es la configuración del MOTOR DE BASE DE DATOS, nos dan 2 opciones MODO AUTENTICACION DE WINDOWS Y MODO MIXTO (autenticación de SQL Server y de Windows). Lo recomendable es seleccionar el MODO MIXTO, al seleccionar esta opción se activan las cajas de contraseña para el inicio de sesión de autenticación de SQL Server (inicio de sesión de sa). **No olvidar la contraseña porque va a ser necesario más adelante.**

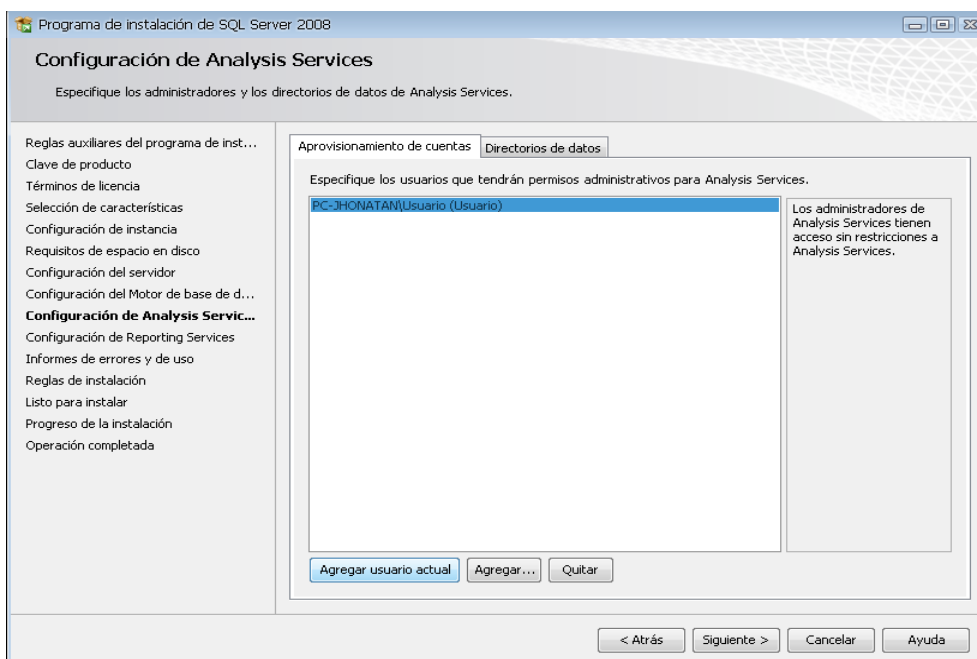


Figura 12. Ventana configuración de analysis services.

Luego nos pide que agreguemos los administradores de SQL Server para lo cual agregamos el usuario actual; luego cli en siguiente.

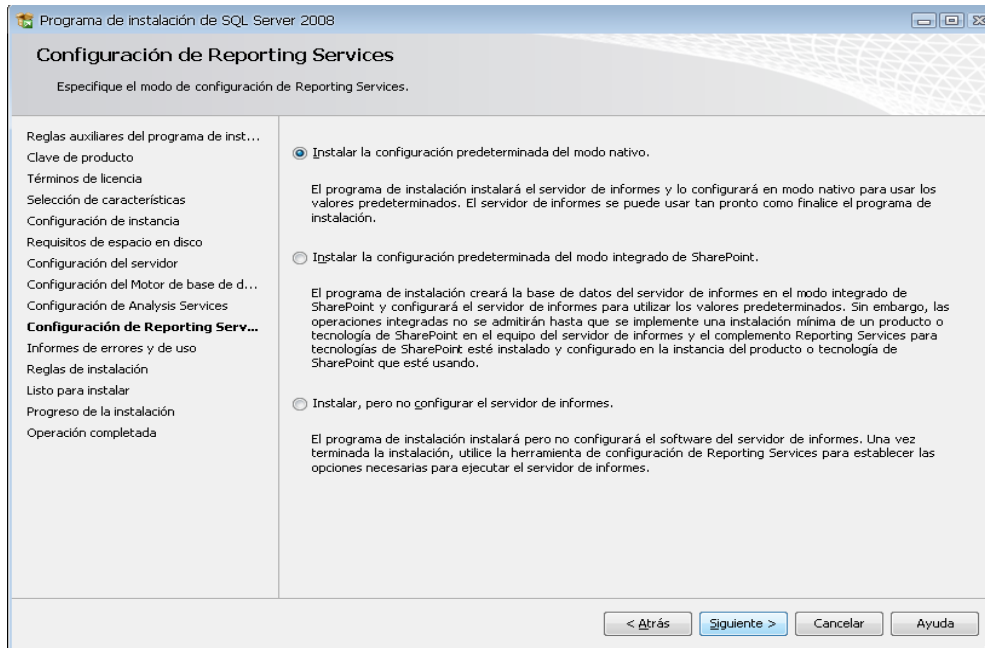


Figura 13. Ventana configuración de reporting services.

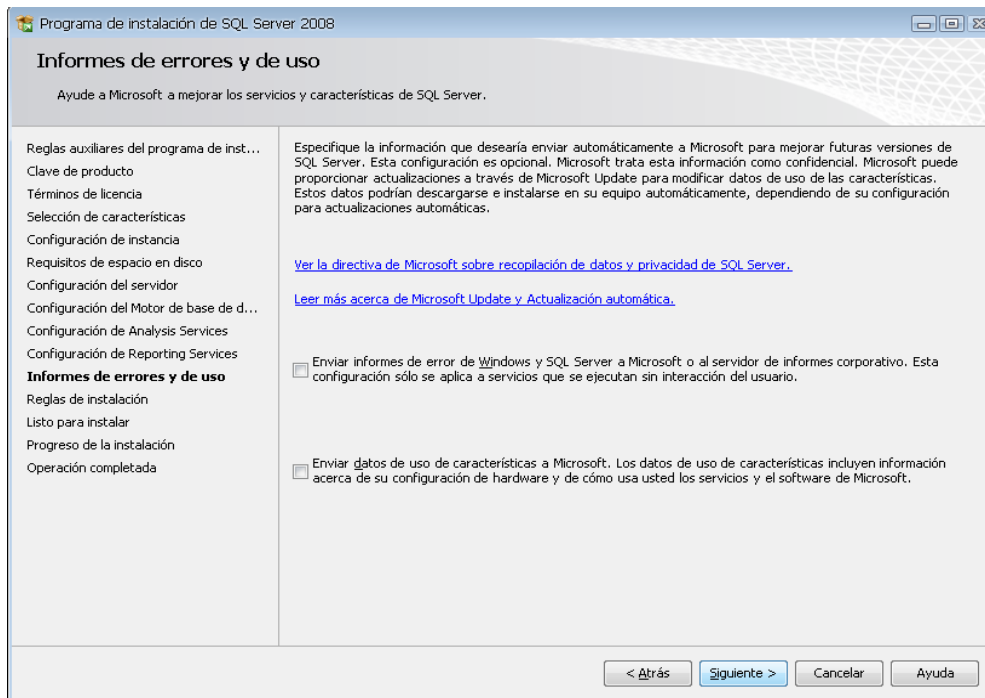


Figura 14. Ventana informes de errores.

En esta ventana de Configuración de Reporting Services también nos dan 3 opciones, por defecto sale seleccionada la primera opción, así que solo damos clic en siguiente (2 veces).

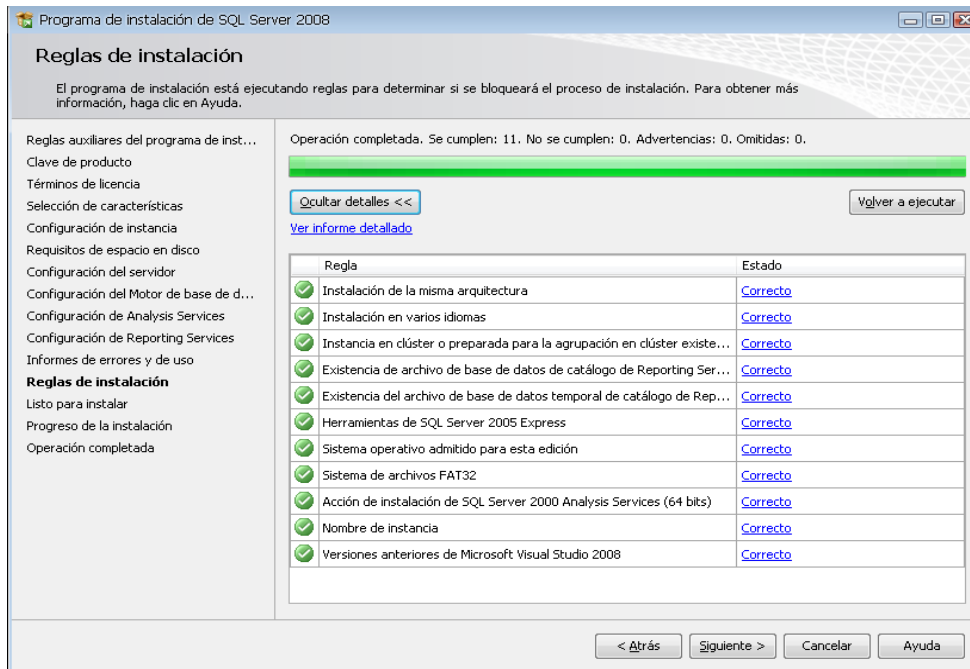


Figura 15. Ventana reglas de instalacion.

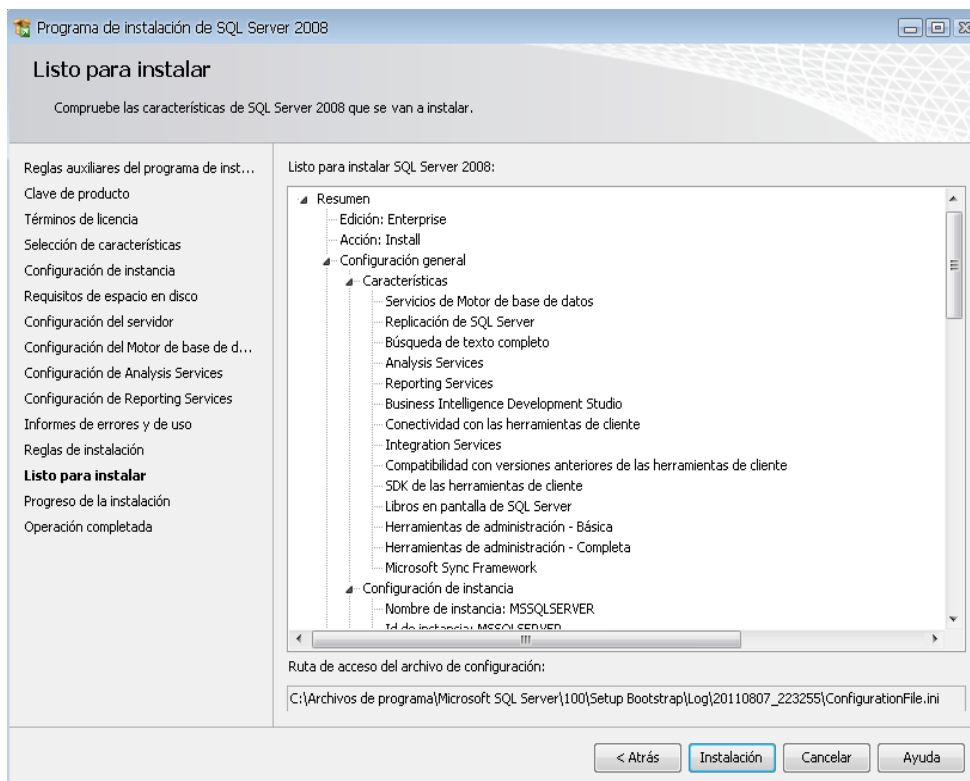


Figura 16. Ventana listo para instalar.

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

**NOTA:** Si todas las reglas no están en estado **Correcto** no se podrá seguir con la instalación (para ver las reglas clic en el botón mostrar detalles).



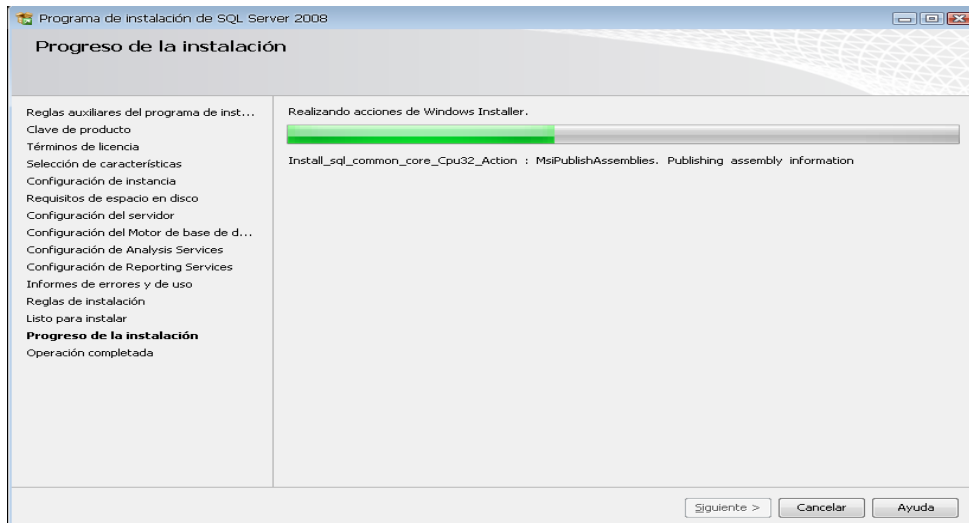


Figura 17. Ventana progreso de instalación.  
Esperamos que termine la instalación, luego clic en siguiente y finalmente en cerrar.

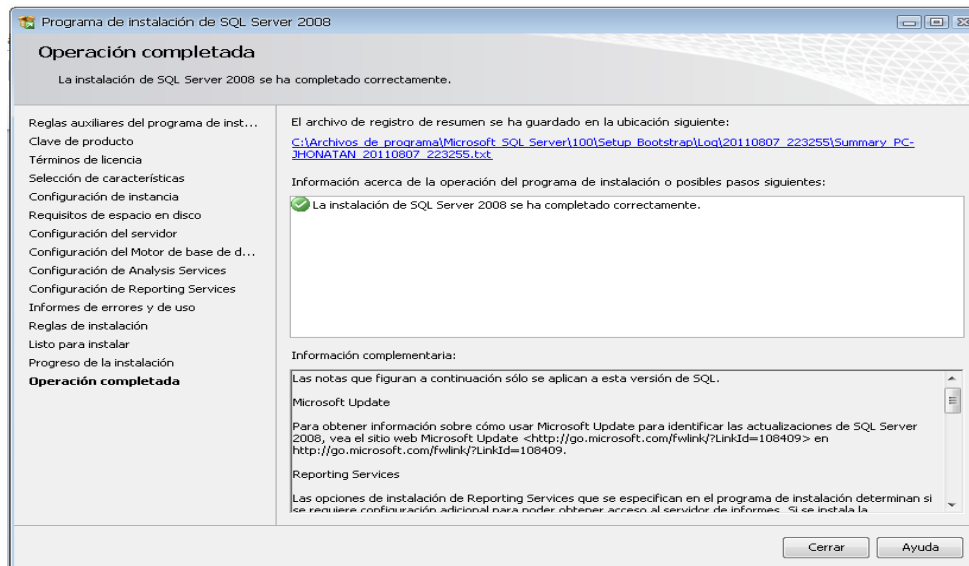


Figura 18. Ventana operación clompletada

### 3.02 Instalación de Visual Estudio 2010



Figura 19. Ventana instalación



Figura 20. Ventana Asistente de instalación.

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"

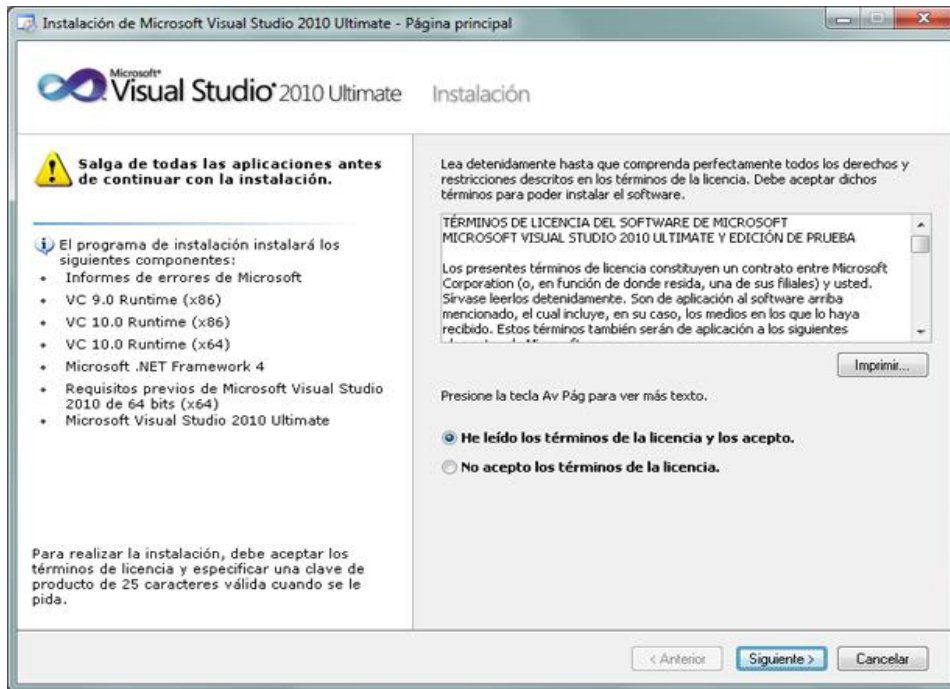


Figura 21. Ventana de términos.

Leeremos los términos de licencia del software de Microsoft Visual Studio 2010 Ultimate (edición de prueba). Si estamos de acuerdo marcaremos "He leído los términos de la licencia y los acepto". Pulsaremos "Siguiente" para continuar:

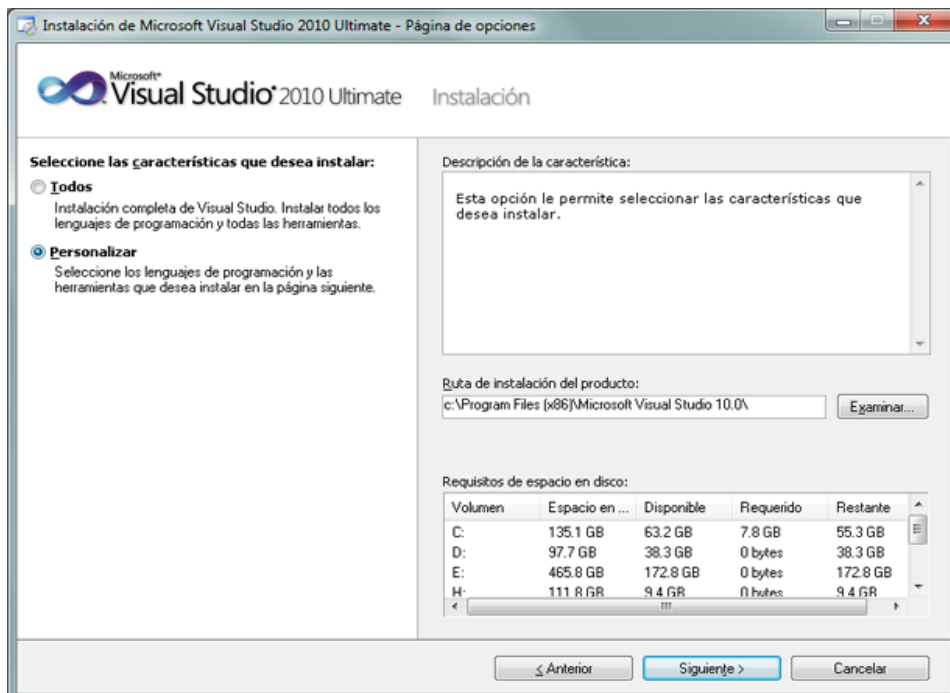


Figura 21. Ventana de personalización

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:

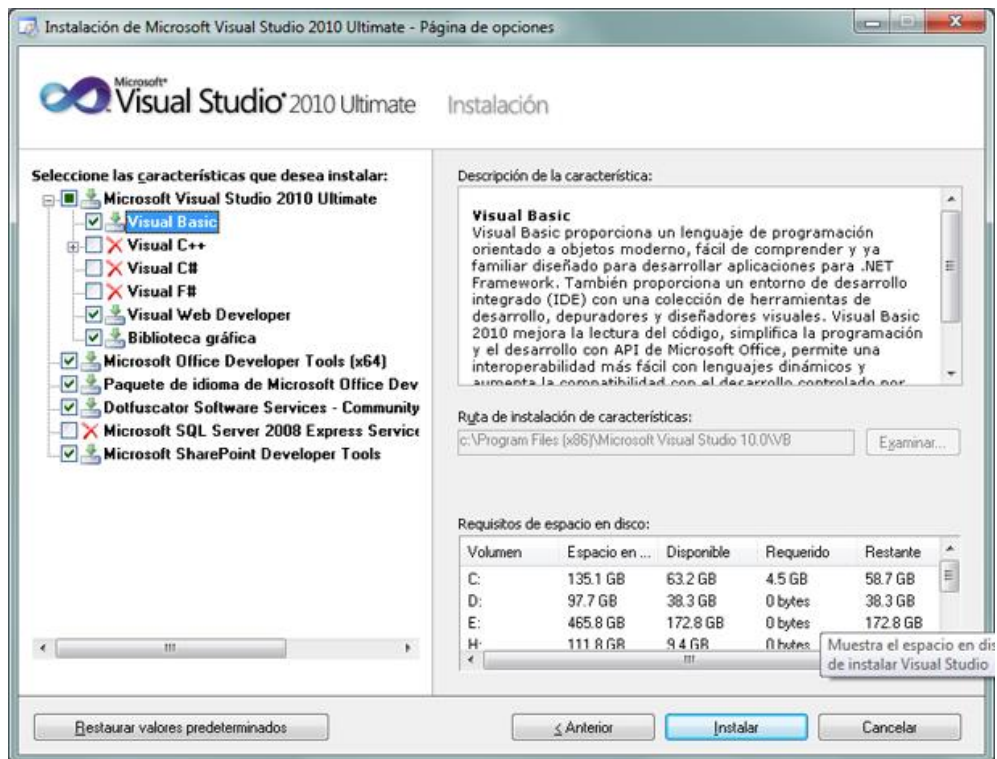


Figura 22. Ventana selección de características.

Seleccionaremos los lenguajes a instalar:

- Visual Basic.
- Visual C++.
- Visual C#.
- Visual F#.

Seleccionaremos también las características a instalar:

- Microsoft Office Developer Tools.
- Dotfuscator Software Services.
- Microsoft SQL Server 2008 Express.
- Microsoft SharePoint Developer Tools.

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

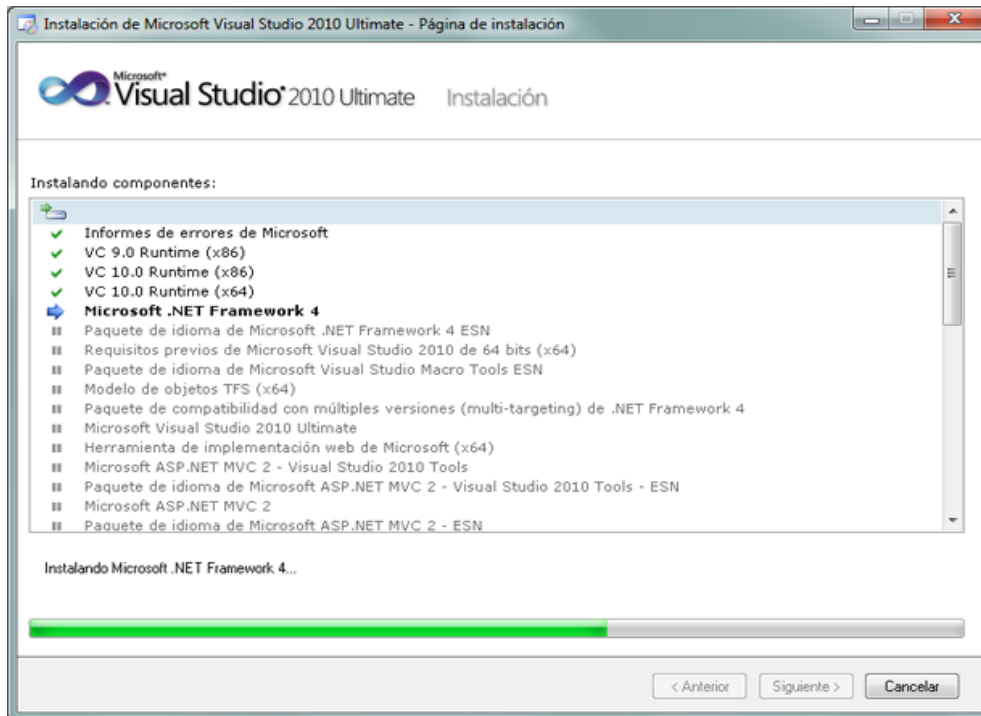


Figura 23. Ventana de instalación de componentes.

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.:

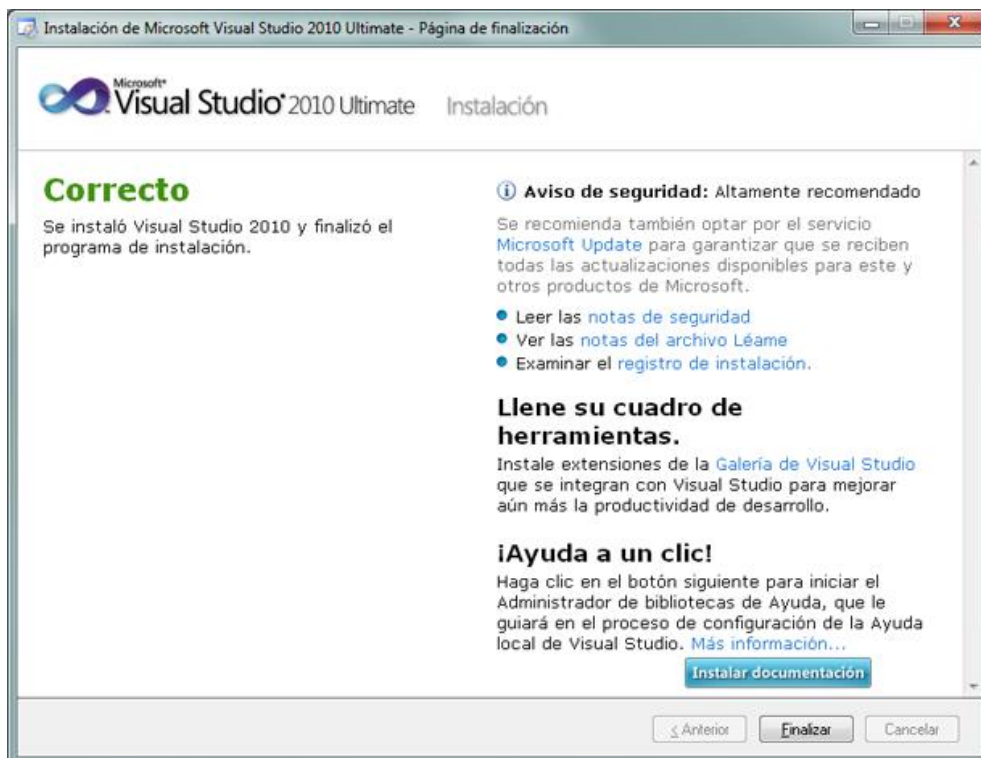


Figura 24. Ventana de instalación Correcta

Pulsaremos "Finalizar":

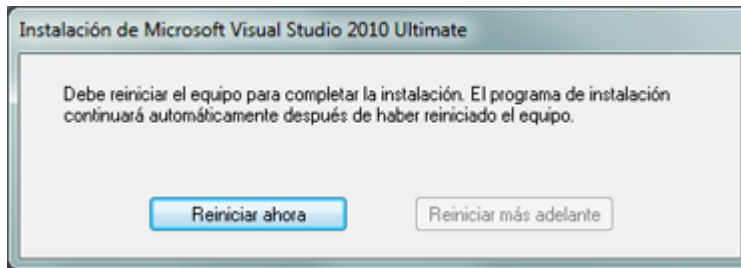


Figura 25. Ventana de reinicio de equipo

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

## Pantalla Principal



Figura 26. Ventana pantalla principal.

## Inicio de Sesión

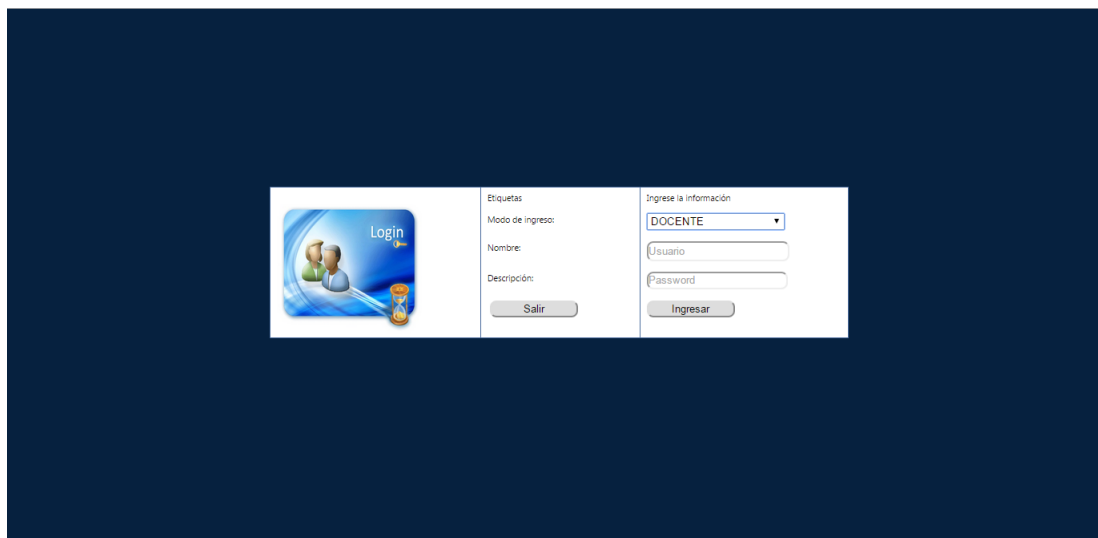


Figura 27. Ventana inicio de sesión



## Ingreso de notas

**SISTEMA DE MATRICULACIÓN MODULO DOCENTE**

LISTAR CALIFICACIONES

Seleccione un Estudiante: PEREZ 2 LUIS Q
 Seleccione un Quimestre: PRIMERO
Buscar

	COD	M	T_1	AC_1	TA_1	L_1	PE_1	P_1	T_2	AC_2	TA_2	L_2	PE_2	P_2	T_3	AC_3	TA_3	L_3	PE_3	P_3	IPR	EQ	PI
Calificar	443	Física	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Calificar	445	Literatura	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Calificar	44L	Matemáticas	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

**INGRESAR CALIFICACIONES**

T\_1

AC\_1

TA\_1

L\_1

PE\_1

T\_2

AC\_2

TA\_2

L\_2

PE\_2

T\_3

AC\_3

TA\_3

L\_3

PE\_3

Guardar

Figura 28. Ventana ingreso de notas.

## A. 09 Manual de Usuario

### Pantalla Principal



Figura 1. Pantalla principal.

En esta interfaz se encuentra la misión, visión, contactos de la institución y el ingreso al sistema.

### Inicio de Sesión

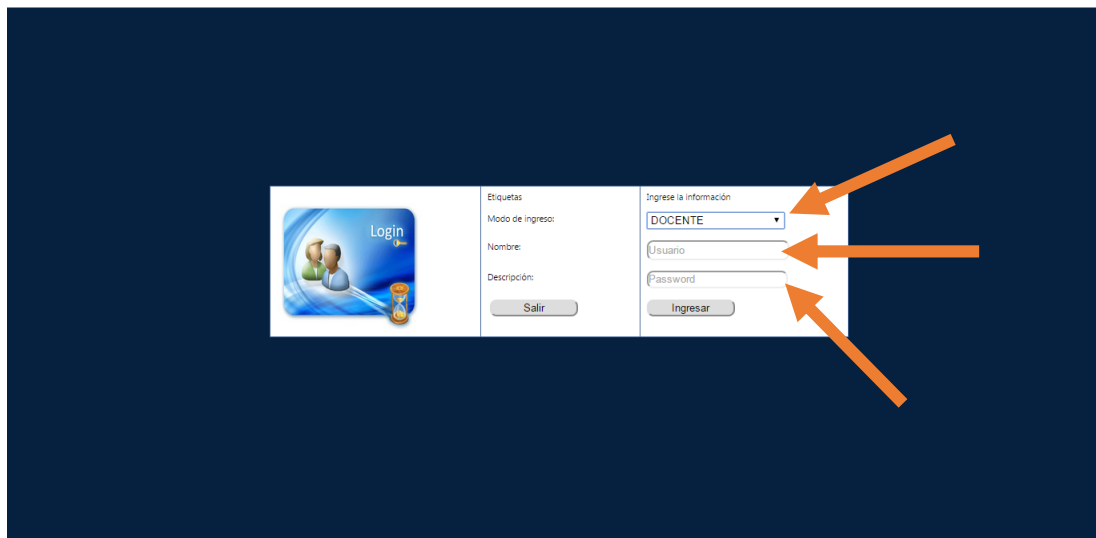


Figura 2. Ventana Inicio de sesión.

Aquí podemos seleccionar el tipo de usuario que sea la persona que ingresa, después colocamos el usuario y contraseña respectiva



## Principal docente



Figura 3. Ventana principal docentes.

Aquí se encuentra la interfaz que tiene el docente y las opciones a escoger

## Ingreso de notas

**SISTEMA DE MATRICULACIÓN MODULO DOCENTE**

LISTAR CALIFICACIONES

Seleccione un Estudiante: PEREZ 2 LUIS Q      Seleccione un Quimestre: PRIMERO      Buscar

	COD	M	T_1	AC_1	TA_1	L_1	PE_1	P_1	T_2	AC_2	TA_2	L_2	PE_2	P_2	AC_3	TA_3	L_3	PE_3	P_3	IPR	EQ	PI
Calificar	443	Física	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Calificar	445	Literatura	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Calificar	441	Matemáticas	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

INGRESAR CALIFICACIONES

T_1	AC_1	TA_1	L_1	PE_1
0,01	0,00	0,00	0,00	0,00
T_2	AC_2	TA_2	L_2	PE_2
0,00	0,00	0,00	0,00	0,00
T_3	AC_3	TA_3	L_3	PE_3
0,00	0,00	0,00	0,00	0,00

Guardar

Figura 4. Ventana ingreso de notas.

Aquí el docente selecciona al alumno y el curso.  
También seleccionara las actividades a calificar.  
Y procederá a llenar los campos requeridos.

## Asistencia alumno

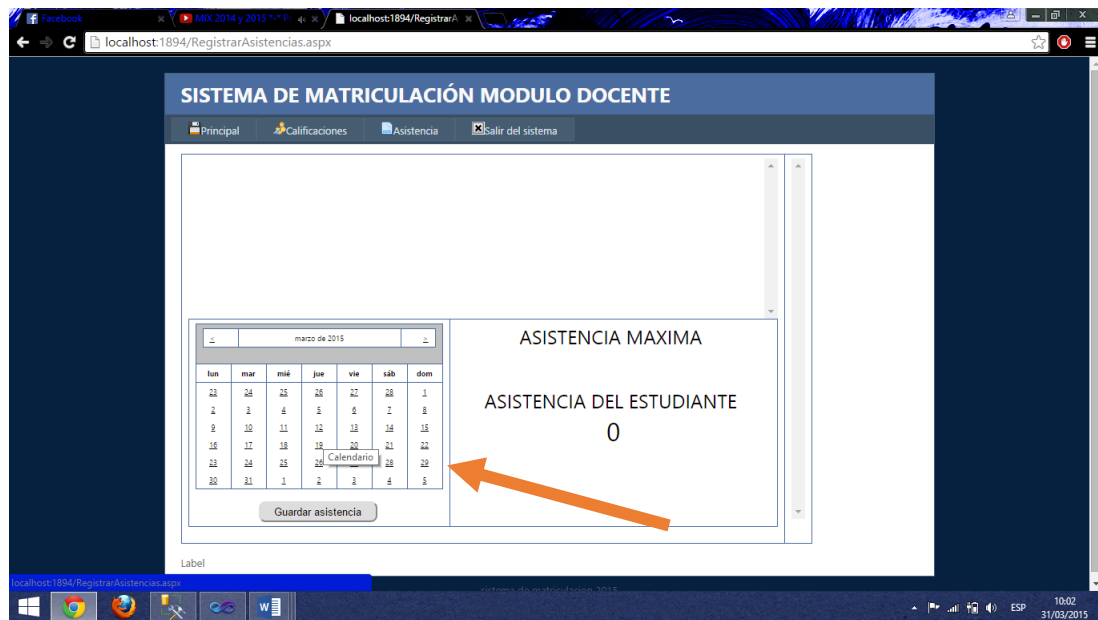


Figura 5. Ventana ingreso de asistencia

El docente deberá escoger el día de asistencia.

## A. 10 Manual Técnico

### Diccionario de Datos

El diccionario se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos de sistemas.

CARLOS-PC.MATRI... dbo.ASISTENCIA			
	Nombre de columna	Tipo de datos	Permitir v...
▶	ASI_CODIGO	numeric(30, 0)	<input type="checkbox"/>
	EST_CODIGO	numeric(30, 0)	<input checked="" type="checkbox"/>
	ASI_FECHA	datetime	<input checked="" type="checkbox"/>
	ASI_ESTADO	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 1. Tabla de Asistencia

CARLOS-PC.MATRI...ON - dbo.CURSOS			
	Nombre de columna	Tipo de datos	Permitir v...
▶	CUR_CODIGO	numeric(30, 0)	<input type="checkbox"/>
	CUR_NOMBRE	varchar(150)	<input checked="" type="checkbox"/>
	CUR_NIVEL	varchar(150)	<input checked="" type="checkbox"/>
	CUR_NUM_ESTUDIAN...	varchar(150)	<input checked="" type="checkbox"/>
	CUR_PARALELO	varchar(150)	<input checked="" type="checkbox"/>
	CUR_ESTADO	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 2. Tabla de Cursos

CARLOS-PC.MATR...N - dbo.DOCENTE			
	Nombre de columna	Tipo de datos	Permitir v...
►?	DOCEN_CODIGO	numeric(30, 0)	<input type="checkbox"/>
	DOCEN_NOMBRE	varchar(150)	<input checked="" type="checkbox"/>
	DOCEN_APELLIDO	varchar(150)	<input checked="" type="checkbox"/>
	DOCEN_DIRECCION	varchar(150)	<input checked="" type="checkbox"/>
	DOCEN_TELEFONO	varchar(20)	<input checked="" type="checkbox"/>
	DOCEN_CORREO	varchar(150)	<input checked="" type="checkbox"/>
	DOCEN_CEDULA	varchar(13)	<input checked="" type="checkbox"/>
	DOCEN_CONTRASENA	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 3. Tabla de Docente.

CARLOS-PC.MAT...DOS_QUIMESTRE			
	Nombre de columna	Tipo de datos	Permitir v...
►?	PERQ_CODIGO	numeric(30, 0)	<input type="checkbox"/>
	QUI_CODIGO	numeric(30, 0)	<input checked="" type="checkbox"/>
	TAREA4	decimal(8, 2)	<input checked="" type="checkbox"/>
	TAREA1	decimal(8, 2)	<input checked="" type="checkbox"/>
	ACT_CLASE1	decimal(8, 2)	<input checked="" type="checkbox"/>
	TALLER1	decimal(8, 2)	<input checked="" type="checkbox"/>
	LECCIONES1	decimal(8, 2)	<input checked="" type="checkbox"/>
	PRUEBA_ESTRUCTUR...	decimal(8, 2)	<input checked="" type="checkbox"/>
	PARCIAL1	decimal(8, 2)	<input checked="" type="checkbox"/>
	TAREA2	decimal(8, 2)	<input checked="" type="checkbox"/>
	ACT_CLASE2	decimal(8, 2)	<input checked="" type="checkbox"/>
	TALLER2	decimal(8, 2)	<input checked="" type="checkbox"/>
	LECCIONES2	decimal(8, 2)	<input checked="" type="checkbox"/>
	PRUEBA_ESTRUCTUR...	decimal(8, 2)	<input checked="" type="checkbox"/>
	PARCIAL2	decimal(8, 2)	<input checked="" type="checkbox"/>
	TAREA3	decimal(8, 2)	<input checked="" type="checkbox"/>
	ACT_CLASE3	decimal(8, 2)	<input checked="" type="checkbox"/>
	TALLER3	decimal(8, 2)	<input checked="" type="checkbox"/>
	LECCIONES3	decimal(8, 2)	<input checked="" type="checkbox"/>
	PRUEBA_ESTRUCTUR...	decimal(8, 2)	<input checked="" type="checkbox"/>
	PARCIAL3	decimal(8, 2)	<input checked="" type="checkbox"/>
	PROMEDIO	decimal(8, 2)	<input checked="" type="checkbox"/>
	EXAMEN_QUIMESTRAL	decimal(8, 2)	<input checked="" type="checkbox"/>
	PROMEDIO_QUIMEST...	decimal(8, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 4. Tabla de Dos\_Quimestres.

CARLOS-PC.MATR...o.QUIMESTRALES			
	Nombre de columna	Tipo de datos	Permitir v...
►?	QUI_CODIGO	numeric(30, 0)	<input type="checkbox"/>
	PEN_CODIGO	numeric(30, 0)	<input checked="" type="checkbox"/>
	MATR_CODIGO	numeric(30, 0)	<input checked="" type="checkbox"/>
	QUI_NOMBRE	varchar(50)	<input checked="" type="checkbox"/>
	QUI_NUMERO	int	<input checked="" type="checkbox"/>
	QUI_NOTA	decimal(8, 2)	<input checked="" type="checkbox"/>
	QUI_ESTADO	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 5. Tabla de Quimestres

CARLOS-PC.MATR...o.PARAMETRIZAR			
	Nombre de columna	Tipo de datos	Permitir v...
▶	CODIGO	numeric(30, 0)	<input type="checkbox"/>
	NOMBRE	varchar(150)	<input checked="" type="checkbox"/>
	DESCRIPCION	varchar(150)	<input checked="" type="checkbox"/>
	ESP_ESTADO	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 6. Tabla de Parametrizar

El desarrollo del sistema, las interfaces se realizaron en Microsoft Visual Studio 2010.

## Inicio de Sesión

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using com.ec.ClsLogicaNegocio;
using com.ec.ClsLogicaNegocio.Entidades;
using com.ec.ClsLogicaNegocio.Vistas;

namespace com.ec.WebTesis
{
    public partial class InicioSesion : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnSalir_Click(object sender, EventArgs e)
        {
            Response.Redirect("Principal.aspx");
        }
        protected void btnRegistrar_Click(object sender, EventArgs e)
        {
            if (!txtContrasena.Text.Equals("") &&
!txtUsuario.Text.Equals(""))
            {
                if (cmbTipoUsuario.SelectedValue.Equals("1"))
                {

```

```
        Usuario usu = new
ClsLogicaNegocio.ClsLogicaNegocio().consultarUsuario(txtUsuario.Text);
        if (usu.usuario != null)
        {
            if
            (usu.contrasena.Trim().Equals(txtContrasena.Text) &&
            usu.usuario.Trim().Equals(txtUsuario.Text))
            {
                Response.Redirect("Default.aspx");
            }
            else
            {
                //Response.Write("<script
language=javascript>alert('Usuario o contraseña incorrecta');</script>");
                lblValidar.Text = "Usuario o contraseña
incorrecta";
            }
        }
        else
        {
            // Response.Write("<script
language=javascript>alert('No existe el usuario');</script>");
            lblValidar.Text = "No existe el usuario";
        }
    }

    if (cmbTipoUsuario.SelectedValue.Equals("2"))
    {
        Docentes doce = new Docentes();
        List<Docentes> docente = new
ClsLogicaNegocio.ClsLogicaNegocio().listarDocente(txtUsuario.Text);
        foreach (var item in docente)
        {
            doce.cedula = item.cedula;
            doce.contrasena = item.contrasena;
        }
        if (doce.cedula != null)
        {
            if (txtUsuario.Text.Trim().Equals(doce.cedula) &&
            txtContrasena.Text.Trim().Equals(doce.contrasena))
            {
                Utilitario.ClsUtilitarios.pasarDatos.cedulaDocente = doce.cedula;
                Response.Redirect("PrincipalDocente.aspx");
            }
            else
            {
                //Response.Write("<script
language=javascript>alert(Verifique el usuario y la
contraseña');</script>");
                lblValidar.Text = "Verifique el usuario y la
contraseña";
            }
        }
        else
        {

```

```

// Response.Write("<script
language=javascript>alert('No existe el usuario');</script>");
        lblValidar.Text = "No existe el usuario";
    }
}

else
{
    // Response.Write("<script
language=javascript>alert('Verifique la informacion');</script>");
    lblValidar.Text = "Verifique la informacion";
}
}
}
}

```

## Calificar Estudiante

```
#region CLASIFICACIONES
public PeriodoQuimestre listaNotasPorCodigo(string codPeriodoQuim)
{
    PeriodoQuimestre repre = new PeriodoQuimestre();

    DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("select
* from PERIODOS_QUIMESTRE where PERQ_CODIGO='" + codPeriodoQuim + "'");
    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
    {
        PeriodoQuimestre aux = new PeriodoQuimestre();
        //aux.MATERIA = ds.Tables[0].Rows[i][0].ToString();
        //aux.NOMBRE_QUIMESTRE = ds.Tables[0].Rows[i][1].ToString();
        aux.CODIGO_PERIODO = ds.Tables[0].Rows[i][0].ToString();
        aux.COD_QUIMESTRE = ds.Tables[0].Rows[i][1].ToString();
        aux.TAREA1 = ds.Tables[0].Rows[i][3].ToString();
        aux.ACT_CLASE1 = ds.Tables[0].Rows[i][4].ToString();
        aux.TALLE1 = ds.Tables[0].Rows[i][5].ToString();
        aux.LECCION1 = ds.Tables[0].Rows[i][6].ToString();
        aux.PRUEBA_EST1 = ds.Tables[0].Rows[i][7].ToString();
        aux.PARCIAL1 = ds.Tables[0].Rows[i][8].ToString();
        aux.TAREA2 = ds.Tables[0].Rows[i][9].ToString();
        aux.ACT_CLASE2 = ds.Tables[0].Rows[i][10].ToString();
        aux.TALLE2 = ds.Tables[0].Rows[i][11].ToString();
        aux.LECCION2 = ds.Tables[0].Rows[i][12].ToString();
        aux.PRUEBA_EST2 = ds.Tables[0].Rows[i][13].ToString();
        aux.PARCIAL2 = ds.Tables[0].Rows[i][14].ToString();
        aux.TAREA3 = ds.Tables[0].Rows[i][15].ToString();
        aux.ACT_CLASE3 = ds.Tables[0].Rows[i][16].ToString();
        aux.TALLE3 = ds.Tables[0].Rows[i][17].ToString();
        aux.LECCION3 = ds.Tables[0].Rows[i][18].ToString();
        aux.PRUEBA_EST3 = ds.Tables[0].Rows[i][19].ToString();
        aux.PARCIAL3 = ds.Tables[0].Rows[i][20].ToString();
        aux.PROMEDIO = ds.Tables[0].Rows[i][21].ToString();
        aux.EXAMEN = ds.Tables[0].Rows[i][22].ToString();
        aux.PROMEDIO_QUIMESTRE = ds.Tables[i].Rows[i][23].ToString();
        repre = aux;
    }
    return repre;
}
//SELECT * FROM CALIFICACIONESPORESTUDIANTE WHERE PERQ CODIGO='435'
```

```

        public List<PeriodoQuimestreView>
listaPeriodoQuimestrePorCodigoView(string codPeriodoQuimestre)
    {
        List<PeriodoQuimestreView> repre = new
List<PeriodoQuimestreView>();

        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("SELECT
* FROM CALIFICACIONESPORESTUDIANTE WHERE PERQ_CODIGO ='" +
codPeriodoQuimestre + "'");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            PeriodoQuimestreView aux = new PeriodoQuimestreView();
            aux.MATERIA = ds.Tables[0].Rows[i][0].ToString();
            aux.CODIGO = ds.Tables[0].Rows[i][2].ToString();
            aux.T_1 = ds.Tables[0].Rows[i][5].ToString();
            aux.AC_1 = ds.Tables[0].Rows[i][6].ToString();
            aux.TA_1 = ds.Tables[0].Rows[i][7].ToString();
            aux.L_1 = ds.Tables[0].Rows[i][8].ToString();
            aux.PE_1 = ds.Tables[0].Rows[i][9].ToString();
            aux.P_1 = ds.Tables[0].Rows[i][10].ToString();
            aux.T_2 = ds.Tables[0].Rows[i][11].ToString();
            aux.AC_2 = ds.Tables[0].Rows[i][12].ToString();
            aux.TA_2 = ds.Tables[0].Rows[i][13].ToString();
            aux.L_2 = ds.Tables[0].Rows[i][14].ToString();
            aux.PE_2 = ds.Tables[0].Rows[i][15].ToString();
            aux.P_2 = ds.Tables[0].Rows[i][16].ToString();
            aux.T_3 = ds.Tables[0].Rows[i][17].ToString();
            aux.AC_3 = ds.Tables[0].Rows[i][18].ToString();
            aux.TA_3 = ds.Tables[0].Rows[i][19].ToString();
            aux.L_3 = ds.Tables[0].Rows[i][20].ToString();
            aux.PE_3 = ds.Tables[0].Rows[i][21].ToString();
            aux.P_3 = ds.Tables[0].Rows[i][22].ToString();
            aux.PR = ds.Tables[0].Rows[i][23].ToString();
            aux.EQ = ds.Tables[0].Rows[i][24].ToString();
            aux.PRQ = ds.Tables[0].Rows[i][25].ToString();
            aux.JORNADA = ds.Tables[0].Rows[i][27].ToString();
            aux.NIVEL = ds.Tables[0].Rows[i][28].ToString();
            aux.PARALELO = ds.Tables[0].Rows[i][29].ToString();

            repre.Add(aux);
        }
        return repre;
    }
    public List<PeriodoQuimestreView> listaPeriodoQuimestreView(string
codEstudiante, string nombreQuimestre)
    {
        List<PeriodoQuimestreView> repre = new
List<PeriodoQuimestreView>();
        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("SELECT
* FROM CALIFICACIONESPORESTUDIANTE WHERE QUI_NOMBRE='" + nombreQuimestre + "'
AND EST_CODIGO='" + codEstudiante + "' ORDER BY PEN_MATERIA ASC");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            PeriodoQuimestreView aux = new PeriodoQuimestreView();
            aux.MATERIA = ds.Tables[0].Rows[i][0].ToString();
            aux.CODIGO = ds.Tables[0].Rows[i][2].ToString();
            aux.SEG = ds.Tables[0].Rows[i][3].ToString();
            aux.T_1 = ds.Tables[0].Rows[i][5].ToString();
            aux.AC_1 = ds.Tables[0].Rows[i][6].ToString();
            aux.TA_1 = ds.Tables[0].Rows[i][7].ToString();
            aux.L_1 = ds.Tables[0].Rows[i][8].ToString();
            aux.PE_1 = ds.Tables[0].Rows[i][9].ToString();

```



```

        aux.P_1 = ds.Tables[0].Rows[i][10].ToString();
        aux.T_2 = ds.Tables[0].Rows[i][11].ToString();
        aux.AC_2 = ds.Tables[0].Rows[i][12].ToString();
        aux.TA_2 = ds.Tables[0].Rows[i][13].ToString();
        aux.L_2 = ds.Tables[0].Rows[i][14].ToString();
        aux.PE_2 = ds.Tables[0].Rows[i][15].ToString();
        aux.P_2 = ds.Tables[0].Rows[i][16].ToString();
        aux.T_3 = ds.Tables[0].Rows[i][17].ToString();
        aux.AC_3 = ds.Tables[0].Rows[i][18].ToString();
        aux.TA_3 = ds.Tables[0].Rows[i][19].ToString();
        aux.L_3 = ds.Tables[0].Rows[i][20].ToString();
        aux.PE_3 = ds.Tables[0].Rows[i][21].ToString();
        aux.P_3 = ds.Tables[0].Rows[i][22].ToString();
        aux.PR = ds.Tables[0].Rows[i][23].ToString();
        aux.EQ = ds.Tables[0].Rows[i][24].ToString();
        aux.PRQ = ds.Tables[0].Rows[i][25].ToString();
        aux.JORNADA = ds.Tables[0].Rows[i][27].ToString();
        aux.NIVEL = ds.Tables[0].Rows[i][28].ToString();
        aux.PARALELO = ds.Tables[0].Rows[i][29].ToString();
        repre.Add(aux);
    }
    return repre;
}
public void actualizarNotas(PeriodoQuimestre per)
{
    string SQL = "UPDATE [MATRICULACION].[dbo].[PERIODOS_QUIMESTRE] "
+
    " SET [TAREA1] = '" + per.TAREA1 + "' " +
    " , [ACT_CLASE1] = '" + per.ACT_CLASE1 + "' " +
    " , [TALLER1] = '" + per.TALLE1 + "' " +
    " , [LECCIONES1] = '" + per.LECCION1 + "' " +
    " , [PRUEBA_ESTRUCTURADA1] = '" + per.PRUEBA_EST1 + "' " +
    " , [TAREA2] = '" + per.TAREA2 + "' " +
    " , [ACT_CLASE2] = '" + per.ACT_CLASE2 + "' " +
    " , [TALLER2] = '" + per.TALLE2 + "' " +
    " , [LECCIONES2] = '" + per.LECCION2 + "' " +
    " , [PRUEBA_ESTRUCTURADA2] = '" + per.PRUEBA_EST2 + "' " +
    " , [TAREA3] = '" + per.TAREA3 + "' " +
    " , [ACT_CLASE3] = '" + per.ACT_CLASE2 + "' " +
    " , [TALLER3] = '" + per.TALLE3 + "' " +
    " , [LECCIONES3] = '" + per.LECCION3 + "' " +
    " , [PRUEBA_ESTRUCTURADA3] = '" + per.PRUEBA_EST3 + "' " +
    "WHERE PERQ_CODIGO='" + per.CODIGO_PERIODO + "'";
    new ClsConexion().ExecuteNonQuery(SQL);
    SQL = "exec sp_calcularNotas '" + per.CODIGO_PERIODO + "'";
    new ClsConexion().ExecuteNonQuery(SQL);
}
public void actualizarQuimestre(string nota, string codQuimestre)
{
    string SQL = "UPDATE [MATRICULACION].[dbo].[QUIMESTRALES] SET
[QUI_NOTA] = '" + nota + "' WHERE QUI_CODIGO='" + codQuimestre + "'";
    new ClsConexion().ExecuteNonQuery(SQL);
}
#endregion

```

## Registrar Notas

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using com.ec.ClsLogicaNegocio;
using com.ec.ClsLogicaNegocio.Entidades;
using com.ec.ClsLogicaNegocio.Vistas;

namespace com.ec.WebTesis
{
    public partial class RegistrarNotas : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            mostrarNotas();
        }
        private void mostrarNotas()
        {
            try
            {
                List<PeriodoQuimestreView> listaPe = new
                ClsLogicaNegocio.ClsLogicaNegocio().listaPeriodoQuimestrePorCodigoView(Utilita
                rio.ClsUtilitarios.pasarDatos.codPeriodoQuimestre);
                dgvNotas.DataSource=listaPe;
                dgvNotas.DataBind();
            }
            catch (Exception)
            {
                throw;
            }
        }
        protected void btnRegistrar_Click(object sender, EventArgs e)
        {
            try
            {
                PeriodoQuimestre per = new PeriodoQuimestre();
                per.TAREA1 = txtT_1.Text;
                per.ACT_CLASE1 = txtAC_1.Text;
                per.TALLE1 = txtTA_1.Text;
                per.LECCION1 = txtL_1.Text;
                per.PRUEBA_EST1 = txtPE_1.Text;
                per.TAREA2 = txtT_2.Text;
                per.ACT_CLASE2 = txtAC_2.Text;
                per.TALLE2 = txtTA_2.Text;
                per.LECCION2 = txtL_2.Text;
                per.PRUEBA_EST2 = txtPE_2.Text;
                per.TAREA3 = txtT_3.Text;
                per.ACT_CLASE3 = txtAC_3.Text;
                per.TALLE3 = txtTA_3.Text;
                per.LECCION3 = txtL_3.Text;
                per.PRUEBA_EST3 = txtPE_3.Text;
                per.CODIGO_PERIODO =
                Utilitario.ClsUtilitarios.pasarDatos.codPeriodoQuimestre;
                new ClsLogicaNegocio.ClsLogicaNegocio().actualizarNotas(per);
            }
            catch (Exception)
            {
            }
        }
    }
}
```

```

        throw;
    }
}
protected void txtT_1_TextChanged(object sender, EventArgs e)
{
}
}
}

```

## Registrar Asistencia

```

#region ASISTENCIA
    public void insertarAsistencia(Asistencia asisten)
    {
        string SQL = "INSERT INTO [MATRICULACION].[dbo].[ASISTENCIA] " +
            " ([EST_CODIGO] " +
            " ,[ASI_FECHA] " +
            " ,[ASI_ESTADO]) " +
            " VALUES " +
            " ('" + asisten.codEstudiante + "' " +
            " ," + asisten.fecha + "' " +
            " ,'Registrado')";
        new ClsConexion().ExecuteNonQuery(SQL);
    }
    public List<AsistenciaView> listaAsistenciaPorEstudiante(string
codEstudiante)
    {
        List<AsistenciaView> asist = new List<AsistenciaView>();

        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("SELECT
*FROM ASISTENCIA A WHERE A.EST_CODIGO='" + codEstudiante + "' ORDER BY
A.ASI_FECHA DESC");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            AsistenciaView aux = new AsistenciaView();
            aux.FECHA = ds.Tables[0].Rows[i][2].ToString();
            aux.ESTADO = ds.Tables[0].Rows[i][3].ToString();
            asist.Add(aux);
        }
        return asist;
    }

    public AsistenciaView listaAsistenciaPorFecha(string fecha, string
idEstudiante)
    {
        AsistenciaView aux = new AsistenciaView();
        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("SELECT
*FROM ASISTENCIA A WHERE A.ASI_FECHA='" + fecha + "' AND A.EST_CODIGO='" +
idEstudiante + "'");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            aux.FECHA = ds.Tables[0].Rows[i][2].ToString();
            aux.ESTADO = ds.Tables[0].Rows[i][3].ToString();
            ;
        }
        return aux;
    }
    public string asistenciaMaximaPorCurso()
    {
        String asistenciaMaxima = "";
    }
}

```

```
        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("select
* from maximaAsistencia");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            asistenciaMaxima = ds.Tables[0].Rows[i][1].ToString();
        }
        return asistenciaMaxima;
    }

    public string asistenciaPorEstudiante(String idEstu)
    {
        String asistenciaMaxima = "";
        DataSet ds = new ClsComunBase.ClsConexion().ExecuteDataSet("SELECT
COUNT (A.ASI_FECHA) FROM ASISTENCIA A WHERE A.EST_CODIGO='" + idEstu + "'");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            asistenciaMaxima = ds.Tables[0].Rows[i][0].ToString();
        }
        return asistenciaMaxima;
    }
}
#endregion
```

## WEBGRAFIA

Educacion, M. d. (2012). Marco legal educativo 2012. En m. l. educativo, *Marco Legal Educativo* (págs. 1-276). Quito: ISBN.

<http://educacion.gob.ec/documentos-legales-y-normativos>

<http://educacion.gob.ec/documentos-pedagogicos>

<http://educacion.gob.ec/calendario-escolar/>

[https://docs.moodle.org/all/es/Calificaciones con Letras#Ecuador](https://docs.moodle.org/all/es/Calificaciones_con_Letras#Ecuador)

[http://www.tomasmoro.ec/portal/boletines/cronograma escolar 2014-2015 - ciclo sierra.pdf](http://www.tomasmoro.ec/portal/boletines/cronograma_escolar_2014-2015_-_ciclo_sierra.pdf)

<http://es.slideshare.net/fiorellaabarca/sistema-de-calificacin-y-significado-de-notas>

<http://es.slideshare.net/clemente2013/educacion-quimestressierra-ecmfil201302190003>