



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

ESCUELA DE SISTEMAS

**Proyecto de Grado, previa obtención del título de:
Tecnólogo Analista de Sistemas**

TEMA:

**DESARROLLO DE UNA PLATAFORMA VIRTUAL TRES CAPAS
PARA LA GESTIÓN DE PROYECTOS DEL INSTITUTO
TECNOLÓGICO SUPERIOR "CORDILLERA"**

AUTOR:

VICTOR ALFONSO CHIZA MONARCO

TUTOR:

ING. JAIME BASANTES

2010 - 2011

QUITO - ECUADOR



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

DECLARACIÓN DE AUTENTICIDAD

Los abajo firmantes, declaramos que los contenidos y los resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del Título de Tecnólogo Analista de Sistemas, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica de los autores.

Víctor Alfonso Chiza Monarco
1723687149

Ing. Jaime Basantes



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

AGRADECIMIENTO

A los Profesores de la institución que con sus conocimientos impartidos día a día me ayudaron a llegar a este término y a mis padres que me dieron todo su apoyo lo largo de toda mi Carrera
Gracias.



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

DEDICATORIA

Este proyecto se lo dedico a mi familia que con su apoyo económico y moral me ayudaron en el progreso y conclusión de mi profesión y me formaron como profesional.



ÍNDICE

CONTENIDO	N° PÁGINA
Índice.....	viii
Índice de Tablas.....	xi
Índice de Imágenes.....	xiii
CAPITULO I.....	1
El Problema.....	1
1.1 Planteamiento del problema.....	1
1.2 Formulación Del Problema.....	3
1.3 Delimitación del Problema.....	3
1.4 Objetivos.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
1.5 Justificación E Importancia.....	4
1.6 Alcances.....	5
CAPITULO II.....	8
Marco Teórico.....	8



2.1 Antecedentes.....	8
2.2 Reseña Histórica.....	8
2.3 Marco Referencial.....	9
2.4 Marco Legal.....	27
CAPITULO III.....	31
Investigación Científica.....	31
3.1 Tipos de Investigación.....	31
3.1.1 Investigación Cuantitativa.....	31
3.1.2 Investigación de Campo.....	32
3.2 Métodos De Investigación.....	34
3.2.1 Métodos Teóricos.....	34
3.3 Herramientas de Recolección de Información.....	37
CAPITULO IV.....	44
Desarrollo De La Propuesta.....	44
4.1 Diagnostico Situacional.....	44
4.2 Estructura Organizacional Jerárquica.....	46
4.3 Infraestructura Informática.....	47
4.3.1 Hardware.....	47
4.3.2 Software.....	47
4.3.3 Comunicaciones.....	47
4.3.4 Recurso Humano Técnico.....	47
4.4 Descripción De Alternativas.....	47
4.4.1 Nombre Alternativa 1.....	48



4.4.2 Nombre Alternativa 2.....	50
4.4.3 Nombre Alternativa 3.....	51
4.5 Evaluación Y Selección De Alternativas.....	52
4.6 Factibilidad Técnica.....	55
4.7 Descripción De Procesos.....	56
4.8 Descripción De Metodología Desarrollo.....	59
4.9 Modelo Lógico.....	62
4.10 Modelo Físico.....	63
4.11 Diccionario De Datos.....	66
4.12 Estándares De Programación.....	70
4.13 Pantallas y Reportes (Prototipo).....	76
4.14 Pruebas Y Depuración.....	81
4.15 Recopilación Y Carga De Datos.....	84
4.16 Pruebas Y Depuración Final En Funcionamiento.....	84
4.17 Puesta En Marcha Del Sistema.....	84
4.18 Capacitación Al Usuario Final.....	85
4.19 Capacitación Al Personal Técnico.....	87
CAPITULO V.....	89
Principales Impactos.....	89
5.1 Científico.....	89
5.2 Educativo.....	89
5.3 Técnico.....	89



5.4	Tecnológico.....	90
5.5	Empresarial.....	90
5.6	Social.....	90
5.7	Económico.....	90
5.8	Conclusiones.....	90
5.9	Recomendaciones.....	92

CAPITULO VI.....96

Bibliografía.....	96
Net Grafía.....	96
Glosario de Términos Técnicos.....	97
Anexos.....	101
Manual De Usuario.....	102
Manual Técnico.....	109

ÌNDICE DE TABLAS

CONTENIDO	N° PÁGINA
Tabla No1 Hardware.....	47
Tabla No2 Software.....	47
Tabla No3 Comunicaciones.....	47
Tabla No4 Recurso Humano Técnico.....	47
Tabla No5 Descripción de Alternativas.....	49
Tabla No6 Descripción de Alternativas 1.....	50



Tabla No7 Descripción de Alternativas 2.....	51
Tabla No8 Descripción de Alternativas 3.....	52
Tabla No9 Evaluación y Selección de Alternativas.....	54
Tabla No10 Tabla de Costos.....	54
Tabla No11 Tabla de Garantías.....	54
Tabla No12 Soporte Técnico.....	55
Tabla No13 Lista de Entidades.....	67
Tabla No14 Lista de Atributos de Entidades.....	70
Tabla No15 Tipo de Datos.....	71
Tabla No16 Estándares de Tablas.....	73
Tabla No17 Estándares de Campos Primarios.....	74
Tabla No18 Estándares de Campos.....	76
Tabla No19 Capacitación de Usuario Final.....	85
Tabla No20 Capacitación de Personal Técnico.....	87
Tabla No21 Recurso Económico.....	128
Tabla No22 Recurso Humano	128



ÌNDICE DE IMÁGENES

CONTENIDO	N° PÁGINA
Gráfico N° 1 (Instituto Tecnológico Superior Cordillera).....	4
Gráfico N° 2 (Fases de RUP 10).....	23
Gráfico N°3(Diagrama Organizacional del Instituto Cordillera).....	47
Gráfico N°4(Asignación de Tutores).....	56
Gráfico N°5(Procesos Estadísticos).....	56
Gráfico N°6(Procesos de Tutores).....	57
Gráfico N°7(Proceso de Lector).....	58
Gráfico N°8(Proceso de Directivos).....	58
Gráfico N°9(Proceso de Administrador).....	59
Gráfico N°10(Proceso de Reportes).....	59
Gráfico N°11(Modelo Lógico).....	63
Gráfico N°12(Modelo Físico).....	65
Gráfico N°13 (Pantalla Login).....	76
Gráfico N°14 (Formularios Búsqueda).....	77
Gráfico N°15 (Formularios de Ingreso de Datos).....	77
Gráfico N°16 (Formularios de Eliminación de Datos).....	77
Gráfico N°17 (Formularios de Modificación).....	78
Gráfico N°18 (Formulario Principal).....	78
Gráfico N°19 (Formulario para Crear Cuentas).....	79
Gráfico N°20 (Formularios de Reportes).....	80



Gráfico N°21 (Formularios de Avances).....	80
Gráfico N°22 (Formularios de Ingreso de Alumnos).....	81
Gráfico N°23 (Formularios para Modificar Datos).....	81
Gráfico N°24 (Ingreso al Sistema).....	102
Gráfico N°25 (Pantalla Principal).....	103
Gráfico N°26 (Pantalla de Inicio de Sesión).....	104
Gráfico N°27 (Mensajes).....	104
Gráfico N°28 (Formularios de Usuarios).....	105
Gráfico N°29 (Formulario de Administrador).....	105
Gráfico N°30 (Formularios de Mantenimientos).....	106
Gráfico N°31 (Formularios para Crear Cuentas).....	106
Gráfico N°32 (Formulario de Procesos).....	107
Gráfico N°33 (Formularios de Búsqueda).....	108



RESUMEN EJECUTIVO

CAPITULO I

El capítulo uno, hace referencia al principal problema de la empresa para saber el objeto de estudio, planteando objetivos generales y objetivos específicos que se llevarán a cabo, según eso analizamos el alcance del proyecto.

CAPITULO II

El capítulo dos, describe en su totalidad a la empresa, como se encuentra actualmente constituida, examina los antecedentes del Instituto Tecnológico Superior Cordillera, para el proceso de toda la información, siendo esta técnica u operativa, realizando diagramas de toda las actividades y un marco conceptual del vocabulario utilizado en la tesis.

CAPITULO III

El capítulo tres, describe las principales metodologías de investigación, así como también las técnicas de recolección de información.

CAPITULO IV

El capítulo cuatro, está enfocado al diagnóstico situacional de la empresa, su infraestructura de software y hardware, se plantea una alternativa conveniente para la empresa enfocándonos en la problemática de la empresa.

CAPITULO V

El capítulo cinco describe los principales impactos que ha tenido el desarrollo de proyecto, las conclusiones y recomendaciones para el mejor funcionamiento de la empresa, así como también el recurso humano.



CAPITULO I EL PROBLEMA

1.1 Planteamiento del problema

El Instituto Tecnológico Superior Cordillera, ha venido experimentando un moderado pero sostenido crecimiento y aumento de su participación en la preparación académica de Tecnólogos Analistas de Sistemas, a causa de la cogida ciudadana por lo que hay en sus tres modalidades de estudio que supera los 4000 estudiantes.

Otras de las causas que contribuyen, es el número de graduados que hay semestralmente que es mayor a los 280 graduados por todas las escuelas. Razón por la cual es muy necesario implementar esta plataforma virtual para controlar los proyectos.

El control de los proyectos es muy fundamental en nuestro instituto, ya que cada alumno deberá escoger un tema para poder desarrollarlos, vemos inconvenientes en controlar estos temas ya que hay una gran cantidad de alumnos, es por ello la necesidad de implementar nuestro sistema.

Razón por la cual todo este proceso de control no está totalmente automatizado, lo cual queremos que se realice de una manera mucho más fácil, sencilla, ordenada y rápida el control de los proyectos que realizan los estudiantes para graduarse.

La base de todo nuestro proyecto tiene como finalidad desarrollar un sistema para el Instituto Tecnológico Superior Cordillera, el cual encierra un conjunto de actividades y procesos, que desarrolladas secuencialmente de una forma eficiente y planificadas correctamente con autoridad, busca, como principal objetivo, suministrar una base de datos



para construir un software de alta calidad en una forma eficiente, y así facilitar el proceso de control de proyectos que satisfaga las necesidades de todo el Instituto Tecnológico Superior Cordillera.

Otro de los problemas que tiene nuestra Institución es el control de avances de proyectos, ya que cada estudiante debe presentar el avance en una fecha asignada por el profesor de proyectos, y mientras no se entregue dicho avance no podrá presentar los siguientes, por medio de esto se lograra controlar de una manera más ordenada, todos los avances de proyectos por alumnos.

Las tutorías que realizan los directores del instituto son deficientes, por la razón de que no tienen un control adecuado de las tutorías que está manejando cada director, por ello en nuestro sistema se lograra controlar este problema de una manera más ordenada.

También de acuerdo a nuestro análisis e investigación de este problema, pudimos observar que no pueden controlar la cantidad de alumnos que están presentando los proyectos, y también por escuelas.

Otra de las deficiencias y dificultades que tiene nuestro instituto es la de controlar los temas repetidos que en cada escuela se presentan, para ello con nuestro sistema se lograra controlar toda esta redundancia de información, que anteriormente era una gran dificultad saber cuántos alumnos estaban presentando los mismos temas.

Con todo este proceso ordenado en nuestro sistema no solo lograremos controlar los temas que están repetidos, sino también el evaluar la cantidad de temas que existen y ver si están o no aprobados, tanto por los directores, como también por el jefe de proyectos.

Otra causa que se puede destacar en el Instituto es también que muchos temas son enfocados en solo ámbito, es decir no existe diversidad de temas por lo que también es muy importante controlar.

Al ver todos estos problemas en el Instituto Tecnológico Superior Cordillera, hemos visto la necesidad de implementar, un sistema completo que satisfaga todas estas necesidades de nuestro instituto, para llevar así todos estos procesos de una manera más ordenada, rápida y



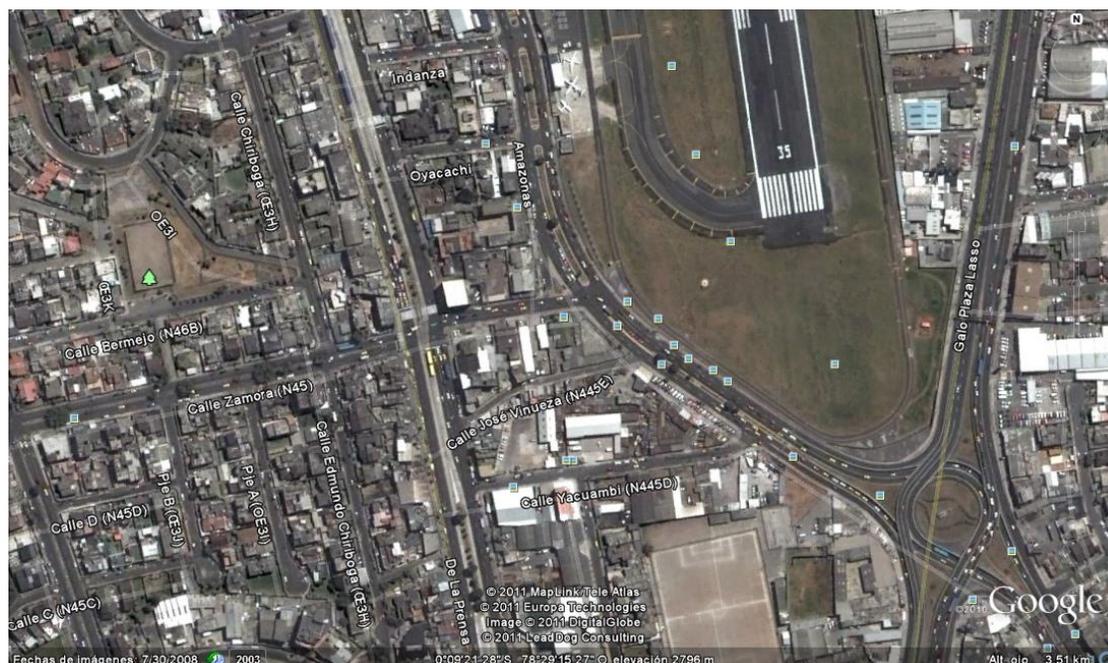
sin redundancia de información, manteniendo de esta manera un eficaz desempeño en todo el Instituto para la revisión de proyectos en todas las escuelas.

1.2 Formulación Del Problema

¿Con la implementación de la Plataforma Virtual tres capas para la gestión de proyectos, se lograra automatizar y mejorar el control de los proyectos en el ITSCO?

1.3 Delimitación del Problema

La implementación de este sistema se llevara a cabo realizando primeramente la delimitación y ubicación de nuestro Instituto Tecnológico Superior Cordillera, que tiene lugar en la provincia de Pichincha, ciudad de Quito, sector Aeropuerto, calle principal: La Prensa y Logroño, que consta de 4000 estudiantes.



Nombre: Instituto Tecnológico Superior Cordillera.

Gráfico: N.- 01.

Fuente: Internet.

1.4 Objetivos



1.4.1 Objetivo General

Desarrollar e implementar la Plataforma Virtual tres capas para el Instituto Tecnológico Superior Cordillera, a fin de mejorar los procesos.

1.4.2 Objetivos Específicos

- Analizar los procesos y reglas del negocio.
- Modelar los procesos y reglas del negocio.
- Programar el código fuente del software aplicativo.
- Realizar pruebas de banco y de campo respecto a nuestro software aplicativo.
- Implementar y capacitar el manejo del software aplicativo.

1.5 Justificación E Importancia

El ITSCO, ha venido experimentando un moderado pero sostenido crecimiento y aumento de su participación en el mercado académico estudiantil, por ello podemos justificar que es muy importante implementar este sistema para automatizar todos los procesos que venían llevando para el control de los proyectos, ya que de esta manera se facilitara este control de una manera global en toda la Institución.

Este sistema de control es muy necesario ya que lograra satisfacer en un cien por ciento los procesos de una manera rápida, ordenada y eficaz los avances de los proyectos para todas las escuelas, llevando así una mejora total el control de cada estudiante.

Razón por la cual todo este proceso de control no está totalmente automatizado, lo cual queremos que se realice de una manera mucho más fácil, sencilla, ordenada y rápida el control de los proyectos que realizan los estudiantes para graduarse.



La base de todo nuestro proyecto tiene como finalidad desarrollar un sistema para el Instituto Tecnológico Superior Cordillera, el cual encierra un conjunto de actividades y procesos, que desarrolladas secuencialmente de una forma eficiente y planificadas correctamente con autoridad, busca, como principal objetivo, suministrar una base de datos para construir un software de alta calidad en una forma eficiente, y así facilitar el proceso de control de proyectos que satisfaga las necesidades de todo el Instituto Tecnológico Superior Cordillera.

De acuerdo con lo anteriormente expuesto se puede deducir claramente la imperiosa necesidad de implementar este sistema, lo que implica a la total atención al manejo de las reglas y procesos del Instituto Tecnológico Superior Cordillera.

La agilidad control y pertinencia de la información se verá reflejada en la velocidad de respuesta que se obtenga a través de la interface de salida de nuestro sistema.

Cabe recalcar que todo este proceso de automatización ayudarán a que toda la operatividad de la entidad se halle enmarcada dentro de los estándares, rapidez, seguridad, solides de manejo de información que sobre todas las cosas debemos dar soluciones y mantener la satisfacción de nuestro Instituto Tecnológico Superior Cordillera, que es la razón para implementar este sistema de control de proyectos.

1.6 Alcances

Dadas las especificaciones y conocimientos generales de nuestro Instituto Tecnológico Superior Cordillera en investigación, se ha deducido que el sistema a implementar según los requerimientos, es un sistema de control de proyectos, mediante este sistemas se podrá generar reportes de forma individual, por escuela y global de los avances de cada proyecto que realizan los estudiantes, para ello toda esta información será almacenada en una base de datos.



- Dada la necesidad de implementar este sistema es importante tener en cuenta los siguientes alcances:
- **Administración de Usuarios:** Por ser una Institución que cuenta con un nivel académico alto, el ITSCO cuenta con una gran variedad de Profesores en cada escuela que ahora administraran los avances de los proyecto como tutores de los alumnos que desarrollan los proyectos.
- **Administración del Control de Proyectos por escuelas:** Dado que el ITSCO cuenta con varias escuelas, se tiene la necesidad de llevar un control ordenado, rápido y eficaz de los proyectos, es necesario administrar por escuelas el avance de los proyectos.
- **Módulo de mantenimiento de tablas:** Ya que por ser un sistema de Control de Proyectos se maneja frecuentemente el movimiento de tablas como son las siguientes:
 - Creación de nuevas tablas si el sistema lo necesita de acuerdo con el desarrollo de los distintos proyectos que pueden surgir en el futuro en nuestra Institución.
 - Eliminación de tablas, si el sistema lo requiere dado el caso de redundancia de información.
 - Modificación de tablas o de campos si los requiera el sistema.
- **Módulo de Seguridad:** El sistema se controlara por medio de claves (las cuales estarán encriptados dentro de la base de datos) el acceso al sistema y a cada una de las secciones del sistema de control, ya que todo va a estar validado, para que solo el personal, autorizado pueda acceder al sistema, tanto como usuario o como administrador cada uno llevar su debido privilegio para ello.
- **Consistencia de la Información:** Este proyecto garantiza la seguridad y consistencia de la información ingresada con el objetivo de mostrar un verdadero análisis. El sistema permitirá administrar y controlar los siguientes procesos:



- Supervisar el control de los proyectos por medio de un Administrador o supe usuario.
- Supervisar el control de los proyectos por medio de un usuario.
- Realizar cambio de equipo.
- Revisar el sistema por medio de la web en cualquier parte del mundo.
- Iniciar Sesión.
- Cerrar Sesión.
- Modificar Contraseñas.
- Administrar perfiles de usuarios.

CAPITULO II MARCO TEÓRICO

2.1 Antecedentes

El ITSCO, quiere facilitar a los aspirantes a nuestra institución una visión tanto global como detallada, de quienes somos y lo que ofrecemos en las distintas carreras que ofertamos. El ITSCO, ha desarrollado su posicionamiento a pasos continuos hasta convertirse en un Instituto competitivo, ya que sabe combinarla calidad de la excelencia académica con la holística. La búsqueda de este equilibrio es nuestra política permanente y nuestra preocupación por la mejora en toda su actividad, ha hecho del ITSCO un centro de estudios superiores alternativo.

También podemos destacar que ya existe un sistema, para el control de los proyectos de grado, pero surge un inconveniente, al ser un sistema dos capas los directores deben estar en



el instituto para hacer el control de los proyectos de grado, es por ello la necesidad de implementar un nuevo sistema, tres capas que se permita utilizarlo mediante la web, facilitando así, la utilización del sistema en cualquier parte del mundo mediante el internet.

2.2 Reseña Histórica

Una de las dimensiones que es necesario tener en cuenta para la renovación de los Diseños Curriculares de la Escuela de Sistemas está vinculada con sus orígenes y la historia de la Institución, de sus estructuras, de sus prácticas y de sus puntos de partida teóricos son las bases sobre las que se sustentan las modificaciones de los planes de estudio de todas sus carreras de grado. Carreras que, en la actualidad, son Administración Bancaria Y Financiera, Administración De Boticas Y Farmacias, Administración De Recursos Humanos-Personal, Administración Industrial Y De La Producción, Administración Turística Y Hotelera, Administración Y Desarrollo Del Patrimonio Cultural, Análisis De Sistemas, Auditoria Informática, Desarrollo Del Talento Infantil, Diseño Gráfico, Marketing Interno Y Externo, Optometría dentro del campo tecnológico y Diseño Gráfico, Mantenimiento Y Reparación De Equipos De Cómputo, Programación De Sistemas, Administración Del Patrimonio Cultural, Administración De Empresas, en el nivel de Técnico Superior y se hallan organizadas en las siguientes carreras: Sistemas, Administración de Empresas, Salud, Educación, Diseño Gráfico, Administración Hotelera.

La dimensión histórica del Instituto Tecnológico Cordillera se sustenta en su riqueza académica, el nivel en la práctica de valores y el aporte permanente de sus profesores; así, es importante destacar que en su "filosofía" da respuesta no sólo a necesidades que tenían que ver con una problemática educativa, la formación de profesionales, sino también a las necesidades sociales que ese momento histórico presentaba y sigue presentando.

Las finalidades, funciones y organización sobre las que se fundó este Instituto proporcionaron al poco tiempo una cantidad de egresados que, a través de su inserción en la sociedad y sus aportes a la cultura, fueron capaces de brindar a las empresas profesionales



competentes y con sólida formación en valores. Formación que en aquel momento se veía amenazada y que supuso, para estos egresados, el cumplimiento profesional de una responsabilidad social de las funciones encomendadas.

Justamente, en el decreto de Fundación del Instituto Tecnológico Superior Cordillera, mediante Acuerdo No. 1999 de Ministerio de Educación y Registro CONESUP No. 17030 del 8 de octubre del 2001; encontramos algunas de estas consideraciones sobre el perfil profesional, que constituyen las huellas precisas del mandato histórico que hemos heredado. Esta perspectiva histórica de la formación de Tecnólogos Analistas de Sistemas se ve influenciada por la situación actual de crisis política, económica, social, de valores y, por lo tanto, educativa. Esto nos compromete a cuestionarnos profundamente acerca de cómo abordar los cambios en la formación de profesionistas en esta carrera, reconociendo que la misma está atravesada por este marco en el cual se hace difícil, pero no imposible, encarar los nuevos caminos que la misma realidad institucional nos plantea.

Según nuestra concepción, esta expectativa la podrá lograr aquel trabajador intelectual, grado que adquiere el estudiante que pasa por las aulas de Instituto Tecnológico Cordillera que en su formación desarrolla:

- Capacidad para reflexionar sobre su propia práctica e incorporar en ella los condicionamientos externos que la atraviesan.
- Capacidad para cuestionar y cuestionarse, para dudar, conmocionarse, para enseñar y, simultáneamente, aprender de los otros.
- Capacidad para fundamentar su práctica y sostener su pensamiento crítico tanto en lo macro (su inserción en la sociedad); como en lo micro (su práctica en el aula y en la institución).
- Capacidad para construirse como un profesional autónomo, es decir, un aquel que tome sus propias decisiones a partir de una relación equilibrada con los demás.

Esta autonomía, es fundamental en el plano operativo, en la toma de decisiones, pero también debe abarcar los aspectos intelectuales y afectivos. Por eso, la formación



Tecnólogos Analistas de Sistemas a la que apuntamos a través de esta mirada, desde una dimensión histórica, tiene en cuenta todas las situaciones analizadas.

De manera similar, aquel que ingresa a esta carrera portando su propia historia escolar, que le pertenece como sujeto, y a partir de la cual construirá su propia formación, el Instituto Tecnológico Cordillera, reafirma su dimensión histórica sintetizada en estas páginas como uno de sus puntos de partida fundamentales, a partir de los cuales será factible concretar la formación de verdaderos transformadores de la realidad y no meros transmisores de conocimientos, incluso en momentos de crisis y convulsión como los que se viven actualmente.

En términos de conocimiento, es indudable que su construcción adquiere sentido cuando se relaciona con la realidad y es transformadora.

¿De qué sirve el conocimiento si no es para ponerlo a disposición del mejoramiento de la calidad de vida de toda la humanidad?

¿De dónde se nutre el conocimiento, sino del estudio de la realidad en el más amplio de los sentidos?

¿Cómo plantearnos la formación de un Tecnólogo Analista de Sistemas que no solo sea transmisor de conocimientos sino transformador de su propia realidad?

Para contestar estos interrogantes una de las cuestiones importantes a tener en cuenta son las dificultades con las que nos vamos a encontrar ante esta necesidad de integrar conocimiento y acción.

Misión: El ITSCO forma profesionales con un perfil de personas cultas, educado y decente, preparadas: moral, científica y técnicamente para afrontar los desafíos de un mundo en constante cambio y asumir la responsabilidad de guiar y salvar a la familia y a la sociedad del caos que amenaza a la humanidad, para optar por una vida de dignidad y libertad.



Visión: A la Distancia el ITSCO ve en cada alumno un caballero y en cada alumna una dama, ciudadanos decentes y cabales, con la mirada en el mundo del mañana donde está por hacerse y en el que serán los profesionales del país que debe ser Ecuador.

Valores:

- Emprendimiento, entendido como asumir el reto de crear y desarrollar nuevas posibilidades en el área del desarrollo social.
- Honestidad, entendida como el equilibrio perfecto entre el conocimiento, el sentimiento, la comunicación y la acción acorde a los valores de verdad y justicia.
- Responsabilidad, entendida como la capacidad de actuar con libertad acorde a sus deberes frente a sus requerimientos y los del entorno.
- Respeto, entendida como el reconocimiento, cumplimiento de las normas y políticas institucionales.
- Tolerancia entendida como el respeto de la diversidad de opiniones, pensamiento, género, cultura, religión, tendencias y criterios; posibilitando una mejor convivencia para la comunidad académica y la sociedad.

2.3 Marco Referencial

Es muy importante implementar este Sistema, para el beneficio de nuestro Instituto, en todos los aspectos tanto para el mejor manejo de los proyectos, como también para mejorar recursos la calidad de los mismos. Para ello podemos destacar los siguientes puntos que son importantes para este sistema de control.

My SQL

Es un datos relacional, [multihilo](#) y [multiusuario](#) con más de seis millones de instalaciones. [MySQL AB](#) —desde [enero de 2008](#) una subsidiaria de [Sun Microsystems](#) y ésta a su vez de [Oracle Corporation](#) desde [abril de 2009](#)— desarrolla MySQL como [software libre](#) en un esquema de licenciamiento dual.



Por un lado se ofrece bajo la [GNU GPL](#) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos [privativos](#) deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en [ANSI C](#).

Al contrario de proyectos como [Apache](#), donde el software es desarrollado por una comunidad pública y el [copyright](#) del código está en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el [copyright](#) de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía [Internet](#). MySQL AB fue fundado por [David Axmark](#), [Allan Larsson](#) y [Michael Widenius](#).

Lenguajes de programación

Existen varias [APIs](#) que permiten, a aplicaciones escritas en diversos [lenguajes de programación](#), acceder a las bases de datos MySQL, incluyendo [C](#), [C++](#), [C#](#), [Pascal](#), [Delphi](#) (viadbExpress), [Eiffel](#), [Smalltalk](#), [Java](#) (con una implementación nativa del driver de Java), [Lisp](#), [Perl](#), [PHP](#), [Python](#), [Ruby](#), [Gambas](#), [REALbasic](#) ([Mac](#) y [Linux](#)), [\(x\)Harbour](#) ([Eagle1](#)), [FreeBASIC](#), y [Tcl](#); cada uno de estos utiliza una [API](#) específica. También existe una interfaz [ODBC](#), llamado [MyODBC](#) que permite a cualquier [lenguaje de programación](#) que soporte [ODBC](#) comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema [SAP](#), lenguaje [ABAP](#).

Aplicaciones

MySQL es muy utilizado en [aplicaciones web](#), como [Drupal](#) o [phpBB](#), en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a [PHP](#), que a menudo aparece en combinación con MySQL. MySQL es una [base de datos](#) muy rápida en la lectura cuando utiliza el motor no transaccional [MyISAM](#), pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreos sobre el desempeño para detectar y corregir errores tanto de SQL como de programación.

Características de la versión 5.0.22

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.



- Soporte a [multiplataforma](#).
- [Procedimientos almacenados](#)
- [Disparadores](#) (*triggers*).
- [Cursores](#)
- [Vistas](#) actualizables.
- Soporte a VARCHAR
- INFORMATION_SCHEMA
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor [InnoDB](#) de Oracle.
- Motores de almacenamiento independientes ([MyISAM](#) para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- Soporte para [SSL](#).

Características adicionales

- Usa GNU Automake, Autoconf, y Libtool para portabilidad
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas groupby y orderby, soporte de funciones de agrupación
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.



- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.

Mejoras futuras

El [mapa de ruta de MySQL 5.1](#) indica soporte para:

- Particionado de la base de datos.
- Backup en línea para todos los motores de almacenamiento.
- Replicación segura
- Restricciones a nivel de columna.
- Planificación de eventos.
- Funciones XML

Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento ([MyISAM](#), Merge, [InnoDB](#), [BDB](#), Memory/heap, [MySQLCluster](#), Federated, Archive, [CSV](#), Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de [transacciones](#) por segundo.

Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándares de **MySQL** son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento [InnoDB](#).
- Max (No se trata de [MaxDB](#), que es una cooperación con [SAP](#)): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de [depuración](#) extra. No debe ser usada en sistemas en producción porque el código de [depuración](#) puede reducir el rendimiento.

Especificaciones del código fuente

MySQL está escrito en una mezcla de [C](#) y [C++](#)

Otras funcionalidades de las listas de correo

- Anuncios: informan sobre nuevas versiones de MySQL y programas relacionados.



- MySQL: lista principal de discusión de MySQL.
- Bugs: permite a la gente buscar y arreglar bugs.
- Temas internos: para gente que trabaja con el código de MySQL. Es el fórum para discutir sobre el desarrollo de MySQL.
- MySQLdoc: para gente que trabaja en documentación.
- Pruebas de rendimiento: para gente interesada en temas de rendimiento no sólo de MySQL, sino de otros motores de bases de datos.
- Empaquetadores: para discusiones sobre empaquetamiento y distribución de MySQL.
- Java: discusiones sobre MySQL Server y Java.

Estado actual

La serie en desarrollo de MySQL Server actualmente, es la 5.1 a la cual se añaden nuevas características en relación a la serie 5.0. La serie de producción actual de MySQL es 5.0, cuya penúltima versión estable es la 5.0.26 lanzada en octubre de 2006. Actualmente, se puede descargar la serie 5.0.27. La serie de producción anterior fue la 4.1, cuya versión estable es 4.1.7 lanzada en octubre de 2004. A estas versiones de producción sólo se arreglan problemas, es decir, ya no se añaden nuevas características. Y a las versiones anteriores solamente se les corrigen bugs críticos.

Active Server Pages (ASP)

Active Server Pages (ASP), también conocido como ASP clásico, es una tecnología de [Microsoft](#) del tipo "lado del servidor" para [páginas web](#) generadas [dinámicamente](#), que ha sido comercializada como un anexo a [Internet Information Services](#) (IIS).

Descripción

La tecnología ASP está estrechamente relacionada con el modelo tecnológico y de [negocio](#) de su fabricante. Intenta ser solución para un modelo de programación rápida ya que "programar en ASP es como programar en [Visual Basic](#) y [C#](#)", por supuesto con muchas limitaciones y algunas ventajas específicas en entornos web.

Lo interesante de este modelo tecnológico es poder utilizar diversos componentes ya desarrollados como algunos controles [ActiveX](#) así como componentes del lado del servidor, tales como [CDONTS](#), por ejemplo, que permite la interacción de los [scripts](#) con el servidor [SMTP](#) que integra [IIS](#).



Se facilita la programación de [sitios web](#) mediante varios objetos integrados, como por ejemplo un objeto de sesión basada en [cookies](#), que mantiene las variables mientras se pasa de página a página.

Es limitado a solo funcionar con [IIS](#), por lo que su uso es cuestionado por la mayoría de los programadores web quienes prefieren otros [lenguajes de programación](#) del lado del servidor como por ejemplo [PHP](#), [Perl](#), [Java](#) Etc.

Versiones

Ha pasado por cuatro versiones mayores:

- ASP 1.0 (distribuido con IIS 3.0)
- ASP 2.0 (distribuido con IIS 4.0)
- ASP 3.0 (distribuido con IIS 5.0)
- [ASP.NET](#) (parte de la plataforma [.NET](#) de Microsoft).

Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP *clásico*.

En el último ASP clásico, ASP 3.0, hay siete [objetos](#) integrados disponibles para el programador: Application, ASPError, Request, Response, Server, Session yObjectContext. Cada objeto tiene un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas web dinámicas.

Desde 2002, el ASP *clásico* está siendo reemplazado por [ASP.NET](#), que entre otras cosas, reemplaza los lenguajes [interpretados](#) como [VBScript](#) o [JScript](#) por lenguajes compilados a código intermedio (llamado MSIL o [Microsoft IntermediateLanguage](#)) como [Visual Basic .NET](#), [C#](#), o cualquier otro lenguaje que soporte la plataforma .NET. El código MSIL se compila con posterioridad a código nativo.

Microsoft Visual Studio

[Microsoft](#) Visual Studio es un [entorno de desarrollo integrado](#) (IDE, por sus siglas en inglés) para sistemas operativos [Windows](#). Soporta varios lenguajes de programación tales como [Visual C++](#), [Visual C#](#), Visual J#, [ASP.NET](#) y [Visual Basic .NET](#), aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Versiones

- [Microsoft Visual Studio 6](#)



- [Microsoft Visual Studio .NET \(2002\)](#)
- [Microsoft Visual Studio .NET 2003](#)
- [Microsoft Visual Studio 2005](#)
- [Microsoft Visual Studio 2008](#)
- [Microsoft Visual Studio 2010](#)

A partir de la versión 2005 Microsoft ofrece gratuitamente las *Express Editions*. Estas son varias ediciones básicas separadas por lenguajes de programación o plataforma enfocadas para novatos y entusiastas. Estas ediciones son iguales al entorno de desarrollo comercial pero sin características avanzadas. Las ediciones que hay son:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition (Desapareció en Visual Studio 2008)
- Visual Web Developer Express Edition (para programar en ASP.NET)
- Visual F# (Apareció en Visual Studio 2010, es parecido al J#)*

Adicionalmente, Microsoft ha puesto gratuitamente a disposición de todo el mundo una versión reducida de [MS SQL Server](#) llamada [SQL Server Express Edition](#) cuyas principales limitaciones son que no soporta bases de datos superiores a 4 GB de tamaño, únicamente utiliza un procesador y un Gb de Ram, y no cuenta con el Agente de SQL Server.

En el pasado se incluyeron los siguientes productos:

- [Visual InterDev](#)
- [Visual J++](#)
- [Visual FoxPro](#)
- [Visual SourceSafe](#)

Historia

Visual Studio 6.0

Se lanzó en 1998 y fue la última versión en ejecutarse en la plataforma [Win9x](#).^[1] Los números de versión de todas las partes constituyentes pasaron a 6.0, incluyendo Visual J++ y Visual InterDev que se encontraban en las versiones 1.1 y 1.0 respectivamente. Esta versión fue la base para el sistema de desarrollo de Microsoft para los siguientes 4 años, en los que Microsoft migró su estrategia de desarrollo al [.NET Framework](#).

Visual Studio 6.0 fue la última versión en que Visual Basic se incluía de la forma en que se conocía hasta entonces; versiones posteriores incorporarían una versión muy diferente del lenguaje con muchas mejoras, fruto de la plataforma .NET. También supuso la última versión en incluir Visual J++, que proporcionaba extensiones de la plataforma Java, lo que lo hacía



incompatible con la versión de [Sun Microsystems](#). Esto acarreó problemas legales a Microsoft, y se llegó a un acuerdo en el que Microsoft dejaba de comercializar herramientas de programación que utilizaran la máquina virtual de Java.

Aunque el objetivo a largo plazo de Microsoft era unificar todas las herramientas en un único entorno, esta versión en realidad añadía un entorno más a Visual Studio 5.0: Visual J++ y Visual Interdev se separaban del entorno de Visual C++, al tiempo que Visual FoxPro y Visual Basic seguían manteniendo su entorno específico.

Visual Studio .NET (2002)

En esta versión se produjo un cambio sustancial, puesto que supuso la introducción de la plataforma .NET de Microsoft. .NET es una plataforma de ejecución intermedia multilenguaje, de forma que los programas desarrollados en .NET no se compilan en [lenguaje máquina](#), sino en un lenguaje intermedio (CIL - [CommonIntermediateLanguage](#)) denominado [Microsoft IntermediateLanguage](#) (MSIL). En una aplicación MSIL, el código no se convierte a lenguaje máquina hasta que ésta se ejecuta, de manera que el código puede ser independiente de plataforma (al menos de las soportadas actualmente por .NET). Las plataformas han de tener una implementación de [Infraestructura de Lenguaje Común](#) (CLI) para poder ejecutar programas MSIL. Actualmente se pueden ejecutar programas MSIL en [Linux](#) y [Mac OS X](#) usando implementaciones de .NET que no son de Microsoft, tales como [Mono](#) y [DotGNU](#).

Visual Studio .NET 2002 supuso también la introducción del lenguaje C#, un lenguaje nuevo diseñado específicamente para la plataforma .NET, basado en C++ y Java. Se presentó también el lenguaje J# -sucesor de J++- el cual, en lugar de ejecutarse en una máquina virtual de Java, se ejecuta únicamente en el framework .NET. El lenguaje Visual Basic fue remodelado completamente y evolucionó para adaptarse a las nuevas características de la plataforma .NET, haciéndolo mucho más versátil y dotándolo con muchas características de las que carecía. Algo similar se llevó a cabo con C++, añadiendo extensiones al lenguaje llamadas [ManagedExtensionsfor C++](#) con el fin de que los programadores pudieran crear programas en .NET. Por otra parte, Visual FoxPro pasa a comercializarse por separado.

Todos los lenguajes se unifican en un único entorno. La interfaz se mejora notablemente en esta versión, siendo más limpia y personalizable.

Visual Studio .NET puede usarse para crear programas basados en Windows (usando [Windows Forms](#) en vez de [COM](#)), aplicaciones y sitios web (ASP.NET y [servicios web](#)), y dispositivos móviles (usando el [.NET Compact Framework](#)).

Esta versión requiere un sistema operativo basado en [NT](#). La versión interna de Visual Studio .NET es la 7.0.



Visual Studio .NET 2003

Visual Studio .NET 2003 supone una actualización *menor* de Visual Studio .NET. Se actualiza el .NET Framework a la versión 1.1. También se añade soporte con el fin de escribir aplicaciones para determinados dispositivos móviles, ya sea con ASP.NET o con el .NET Compact Framework. Además el compilador de Visual C++ se mejora para cumplir con más estándares, el Visual C++ Toolkit 2003.

Visual Studio 2003 se lanza en 4 ediciones: Academic, Professional, Enterprise Developer, y Enterprise Architect. La edición Enterprise Architect incluía una implementación de la tecnología de modelado [Microsoft Visio](#), que se centraba en la creación de representaciones visuales de la arquitectura de la aplicación basadas en [UML](#). También se introdujo "Enterprise Templates", para ayudar a grandes equipos de trabajo a estandarizar estilos de programación e impulsar políticas de uso de componentes y asignación de propiedades.

Microsoft lanzó el *Service Pack 1* para Visual Studio 2003 el 13 de Septiembre de 2006.

La versión interna de Visual Studio .NET 2003 es la 7.1 aunque el formato del archivo es 8.0.

Visual Studio 2005

Visual Studio 2005 se empezó a comercializar a través de [Internet](#) a partir del 4 de Octubre de 2005 y llegó a los comercios a finales del mes de Octubre en inglés. En castellano no salió hasta el 4 de Febrero de 2006. Microsoft eliminó *.NET*, pero eso no indica que se alejara de la plataforma .NET, de la cual se incluyó la versión 2.0.

La actualización más importante que recibieron los lenguajes de programación fue la inclusión de *tipos genéricos*, similares en muchos aspectos a las plantillas de C++. Con esto se consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en áreas donde antes no era posible. C++ tiene una actualización similar con la adición de C++/CLI como sustituto de C# manejado.

Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Visual Studio 2005 también añade soporte de 64-bit. Aunque el entorno de desarrollo sigue siendo una aplicación de 32 bits Visual C++ 2005 soporta compilación para x86-64 ([AMD64](#) e [Intel 64](#)) e [IA-64 \(Itanium\)](#). El [SDK](#) incluye compiladores de 64 bits así como versiones de 64 bits de las librerías.

Visual Studio 2005 tiene varias ediciones radicalmente distintas entre sí: Express, Standard, Professional, Tools for Office, y 5 ediciones Visual Studio TeamSystem. Éstas últimas se



proporcionaban conjuntamente con suscripciones a [MSDN](#) cubriendo los 4 principales roles de la programación: Architects, Software Developers, Testers, y DatabaseProfessionals. La funcionalidad combinada de las 4 ediciones TeamSystem se ofrecía como la edición Team Suite.

Tools for the Microsoft Office System está diseñada para extender la funcionalidad a Microsoft Office.

Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas disponibles gratuitamente a través de la página de Microsoft^[2] se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows, y Visual Web Developer para la creación de sitios web ASP.NET. Las ediciones express carecen de algunas herramientas avanzadas de programación así como de opciones de extensibilidad.

Visual Studio 2008

Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.^[3]

El nuevo framework (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows CommunicationFoundation" (WCF) y "Windows PresentationFoundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.^[4]

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades.

- La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para [depurar](#) y optimizar.
- Con Visual Studio Tools for Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.



- Visual Studio 2008 permite incorporar características del nuevo Windows PresentationFoundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft FoundationClass Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.
- LINQ (LanguageIntegratedQuery) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a Base de Datos, a través de extensiones para C++ y Visual Basic así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando todos la misma sintaxis, prescindiendo del uso de lenguajes especializados como SQL o XPath.
- Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).
- .NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

Visual Studio 2010

Visual Studio 2010 es la versión más reciente de esta herramienta, acompañada por .NET Framework 4.0. La fecha prevista para el lanzamiento de la versión final ha sido el 12 de abril de 2010.^[5]

Hasta ahora, uno de los mayores logros de la versión 2010 de Visual Studio ha sido el de incluir las herramientas para desarrollo de aplicaciones para [Windows 7](#), tales como herramientas para el desarrollo de las características de [Windows 7 \(System.Windows.Shell\)](#) y la RibbonPreview para WPF.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo. Además de esto, aparece una edición que compila las características de todas las ediciones comunes de Visual Studio: Professional, Team Studio, Test, conocida como Visual Studio Ultimate.

Rational Rose Enterprise

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte UnifiedModelingLanguage™ (UML™).



Rational Rose Enterprise es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Características adicionales incluidas:

Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "DesignPatterns: Elements of Reusable Object-Oriented Software"

Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos

Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5

La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables

Soporte Enterprise Java Beans™ 2.0

Capacidad de análisis de calidad de código

El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web

Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos

Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación

Integración con otras herramientas de desarrollo de Rational

Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a RationalClearCase

Publicación web y generación de informes para optimizar la comunicación dentro del equipo

Metodologías RUP

El Proceso Racional Unificado. Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado [UML](#), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de [IBM](#), el cual incluye información entrelazada de diversos [artefactos](#) y descripciones de las



diversas actividades. Está incluido en el RationalMethodComposer (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el [Proceso Unificado](#), y una especificación más detallada, el RationalUnifiedProcess, que se vendiera como producto independiente.

Principios de desarrollo

El RUP está basado en 6 principios clave que son los siguientes:

Adaptar el proceso

El proceso deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él. Las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto en un área subformal.

Equilibrar prioridades

Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. *Debe encontrarse un equilibrio que satisfaga los deseos de todos.* Gracias a este equilibrio se podrán corregir desacuerdos que surjan en el futuro.

Demostrar valor iterativamente

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados

Colaboración entre equipos

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.

Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes [4GL](#) o marcos de referencia ([frameworks](#)) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requisitos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y



soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje [UML](#).

Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

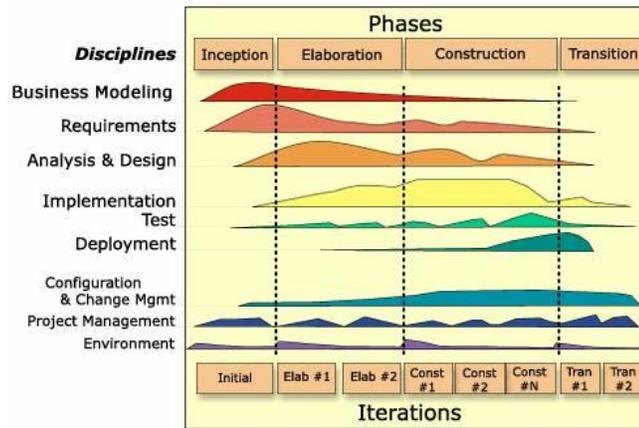
RUP: El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

- Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso a lo largo de su desenvolvimiento
- Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo a su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos (milestones).

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.



Nombre:Fases de RUP.

Gráfico: N.- 02.

Fuente: Internet.

El Proceso Unificado se basa en componentes (component-based), lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas (well-defined interfaces).

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave: dirigido por casos de uso (use-case driven), centrado en la arquitectura (architecture-centric), iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

El Proceso Unificado es dirigido por casos de uso

Un sistema de software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios prospectos.

El término usuario se refiere no solamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término usuario representa algo o alguien que interactúa con el sistema por desarrollar.



Un *caso de uso* es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el *modelo de casos de uso* el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Una especificación funcional tradicional se concentra en responder la pregunta: ¿Qué se supone que el sistema debe hacer? La estrategia de casos de uso puede ser definida agregando tres palabras al final de la pregunta: ¿por cada usuario? Estas tres palabras tienen una implicación importante, nos fuerzan a pensar en términos del valor a los usuarios y no solamente en términos de las funciones que sería bueno que tuviera. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo.

Aún y cuando los casos de uso dirigen el proceso, no son elegidos de manera aislada. Son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida.

El Proceso Unificado está centrado en la arquitectura

El papel del arquitecto de sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. El edificio se mira desde diferentes puntos de vista: estructura, servicios, plomería, electricidad, etc. Esto le permite al constructor ver una radiografía completa antes de empezar a construir. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido.

El concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros stakeholders, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales (ej. desempeño, confiabilidad). La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. Ya que lo importante depende en parte del criterio, el cual a su vez viene con la experiencia, el valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo, el proceso



ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad (understandability), flexibilidad en los cambios futuros (resilience) y reuso.

¿Cómo se relacionan los casos de uso con la arquitectura? Cada producto tiene función y forma. Uno sólo de los dos no es suficiente. Estas dos fuerzas deben estar balanceadas para obtener un producto exitoso. En este caso función corresponde a los casos de uso y forma a la arquitectura. Existe la necesidad de intercalar entre casos de uso y arquitectura. Es un problema del “huevo y la gallina”. Por una parte, los casos de uso deben, cuando son realizados, acomodarse en la arquitectura. Por otra parte, la arquitectura debe proveer espacio para la realización de todos los casos de uso, hoy y en el futuro. En la realidad, ambos arquitectura y casos de uso deben evolucionar en paralelo.

El Proceso Unificado es Iterativo e Incremental

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada.

Los desarrolladores basan su selección de qué van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

UML

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de [modelado](#) de sistemas de [software](#) más conocido y utilizado en la actualidad; está respaldado por el [OMG](#) (Object Management



Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o [RUP](#)), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Diagramas

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la figura de la derecha.

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

- [Diagrama de clases](#)
- [Diagrama de componentes](#)
- [Diagrama de objetos](#)
- [Diagrama de estructura compuesta](#) (UML 2.0)
- [Diagrama de despliegue](#)



- [Diagrama de paquetes](#)

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- [Diagrama de actividades](#)
- [Diagrama de casos de uso](#)
- [Diagrama de estados](#)

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- [Diagrama de secuencia](#)
- [Diagrama de comunicación](#), que es una versión simplificada del [Diagrama de colaboración](#) (UML 1.x)
- [Diagrama de tiempos](#) (UML 2.0)
- [Diagrama global de interacciones o Diagrama de vista de interacción](#) (UML 2.0)

2.4 Marco Legal

La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. Es importante la implantación de este proyecto como tal, en donde esencialmente organizaremos un conjunto de acciones y actividades a realizar, que implican el uso y aplicación de recursos humanos, de la mano a la tecnología, donde la "acción-aprendizaje" será el factor característico de dicho proyecto.

Un derecho humano universal y un deber social fundamental, que debe estar orientada al desarrollo del potencial creativo, y liberador del ser humano y de la sociedad.

Ley de Propiedad Intelectual

Art.1. El Estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las Decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes en el Ecuador.

La propiedad intelectual comprende:

1. Los derechos de autor y derechos conexos.
2. La propiedad industrial, que abarca, entre otros elementos, los siguientes:
 - a. Las invenciones;



- b. Los dibujos y modelos industriales;
 - c. Los esquemas de trazado (topografías) de circuitos integrados;
 - d. La información no divulgada y los secretos comerciales e industriales;
 - e. Las marcas de fábrica, de comercio, de servicios y los lemas comerciales;
 - f. Las apariencias distintivas de los negocios y establecimientos de comercio;
 - g. Los nombres comerciales;
 - h. Las indicaciones geográficas; e,
 - i. Cualquier otra creación intelectual que se destine a un uso agrícola, industrial o comercial.
3. Las obtenciones vegetales.

Las normas de esta Ley no limitan ni obstaculizan los derechos consagrados por el Convenio de Diversidad Biológica, ni por las leyes dictadas por el Ecuador sobre la materia.

Art. 2. Los derechos conferidos por esta Ley se aplican por igual a nacionales y extranjeros, domiciliados o no en el Ecuador

Art. 3. El Instituto Ecuatoriano de la Propiedad Intelectual (IEPI), es el Organismo Administrativo Competente para propiciar, promover, fomentar, prevenir, proteger y defender a nombre del Estado Ecuatoriano, los derechos de propiedad intelectual reconocidos en la presente Ley y en los tratados y convenios internacionales, sin perjuicio de las acciones civiles y penales que sobre esta materia deberán conocerse por la Función Judicial.

Ley de Ordenanza Municipal

Art. P.2.- ORDENANZAS.- El Concejo Metropolitano de Quito solo podrá expedir como ordenanzas normas de carácter general que serán, necesariamente, reformatorias de este Código, ya por modificar sus disposiciones, ya por agregarle otras nuevas, y se denominará ordenanzas metropolitanas. Se excluyen de lo previsto en el inciso anterior las siguientes ordenanzas:

- a) Las relacionadas con el presupuesto municipal, que se denominarán ordenanzas presupuestarias;
- b) La Ordenanza de Zonificación y sus reformas;
- c) El Código de Arquitectura y sus reformas; las ordenanzas en virtud de las cuales se de nombre a calles, plazas u otros bienes de dominio público; las ordenanzas que, pese a no contener disposiciones de carácter general, se expiden como tales por mandato de norma expresa; y las ordenanzas que regulan urbanizaciones sujetas a reglamentación particular. Las tres fases de ordenanzas a las que se refiere este artículo tendrán, cada una de ellas, una numeración distinta e independiente.



Art. P.3.- VIGENCIA.- Las ordenanzas, salvo que en ellas se indique lo contrario, entrarán en vigencia a partir de la fecha de su sanción. Esta norma no es aplicable a las ordenanzas de carácter tributario, que se someterán a las reglas del Código Tributario y la Ley de Régimen Municipal.

Art. P.4.- PUBLICACION.- Todas las ordenanzas se publicarán en la Gaceta Municipal y las ordenanzas metropolitanas, además, en el Registro Oficial.

Art. P.5.- OBLIGACION DEL SECRETARIO DEL CONCEJO.- El Secretario del Concejo Metropolitano está obligado a emitir una copia de toda Ordenanza sancionada por el Alcalde al Administrador General, al Procurador Metropolitano, a los directores generales, administradores zonales, gerentes de empresas metropolitanas y comisarios metropolitanos.

Art. P.6.- ACUERDOS Y RESOLUCIONES.- Las decisiones del Concejo Metropolitano que no tengan carácter general, se expedirán mediante acuerdos o resoluciones. Se expedirán también como acuerdos o resoluciones las reformas o derogatorias de disposiciones que, pese a no tener carácter general, hayan sido expedidas como ordenanzas municipales, antes de la vigencia de este Código.

Ley de Educación Superior

Art. 1.- Forman parte del Sistema Nacional de Educación Superior ecuatoriano:

a) Las universidades y escuelas politécnicas creadas por ley y las que se crearen de conformidad con la Constitución Política y la presente ley. Estas podrán ser públicas financiadas por el Estado, particulares cofinanciadas por el Estado y particulares autofinanciadas.

b) Los institutos superiores técnicos y tecnológicos que hayan sido autorizados por el Ministerio de Educación y Cultura y que sean incorporados al Sistema, así como los que se crearen de conformidad con la presente ley.

Art. 2.- Las instituciones del Sistema Nacional de Educación Superior ecuatoriano, esencialmente pluralistas, están abiertas a todas las corrientes y formas del pensamiento universal expuestas de manera científica. Dirigen su actividad a la formación integral del ser humano para contribuir al desarrollo del país y al logro de la justicia social, al fortalecimiento de la identidad nacional en el contexto pluricultural del país, a la afirmación de la



democracia, la paz, los derechos humanos, la integración latinoamericana y la defensa y protección del medio ambiente.

Art. 3.- Las instituciones del Sistema Nacional de Educación Superior ecuatoriano, en sus diferentes niveles, tienen los siguientes objetivos y estrategias fundamentales:

- a) Formar, capacitar, especializar y actualizar a estudiantes y profesionales en los niveles de pregrado y posgrado, en las diversas especialidades y modalidades;
- b) Preparar a profesionales y líderes con pensamiento crítico y conciencia social, de manera que contribuyan eficazmente al mejoramiento de la producción intelectual y de bienes y servicios, de acuerdo con las necesidades presentes y futuras de la sociedad y la planificación del Estado, privilegiando la diversidad en la oferta académica para propiciar una oportuna inserción de los profesionales en el mercado ocupacional.

CAPITULO III INVESTIGACIÓN CIENTÍFICA

3.1 Tipos de Investigación

3.1.1 Investigación Cuantitativa

Bibliográfica Documental

Este tipo de documentación documental se aplica en su totalidad en el sistema de control de proyectos, ya que con la finalidad de extraer datos de información del mismo tengo que recurrir a fuentes de información que por tratarse de un asunto técnico están almacenados en una categoría especial, la recopilación de información necesaria de libros técnicos, manuales, revistas, brochurts, catálogos me permitirá analizar los procesos y procedimientos propios del sistema que tengo que desarrollar e igualmente me permitirá conocer el origen de la información del flujo que sigue la misma y donde necesita ser almacenada mediante un diseño lógico y coherente materializado la información la información recopilada de los documentos antes mencionados.



Igual tratamiento se lo realizara con la información que obtenga a través de internet, CD y material magnético que pueden aportar al desarrollo y estructuración de mi proyecto.- cabe recalcar que toda información referente a este tema será como valido, ya que la estructuración de la misma a través de los modelos lógicos que son de entendimiento para los usuarios programadores para la generación de base de datos que finalmente darán como resultado sistemas informáticos consistentes rápidos y confiables.

La investigación se desarrolla mediante un proceso que ordena una serie de actividades que se realizan en varias fases o etapas: La selección el tema y la consulta bibliográfica preliminar. En todos estos pasos se han de usar las fuentes; en realidad las fuentes es algo que usamos en nuestras vidas cotidianas, para consultar una palabra, buscamos en algún diccionario o como ya lo hemos utilizado fuentes de internet, que como se sabe, ya se había realizado con anterioridad.

Como se ha dicho precedentemente, las fuentes son materiales que existen con anterioridad y que en su mayoría nos dicen las procedencias y origen de las cosas, la finalidad de las fuentes es el de orientarnos para salir de las dudas que tengamos del origen de algo.

3.1.2 Investigación de Campo

Descriptiva

Este tipo de investigación se aplica en mi proyecto de investigación ya que con el mismo me permite analizar y describir todos los procesos y procedimientos que son parte del sistema a desarrollarse igualmente me permite determinar los rangos y cualidades del recurso humano que forman parte de estos procesos; poniendo especial énfasis en el comportamiento del recurso humano pero anteponiendo lo que nos interesa en cada uno de los procedimientos estructurales de la organización es muy importante la aplicación de este tipo de investigación porque me permitirá ir analizando los tiempos de ejecución de cada tarea por parte del recurso humano, y al mismo tiempo el aporte que significa estos tipos de información para el



proyecto presentado. Una parte importante de la investigación se incluye en el primero de estos objetivos.

Con el fin de describir las características de los usuarios de un producto o servicio, muchas empresas realizan estudios sobre el perfil de los consumidores. Tales perfiles hacen uso de características demográficas, socioeconómicas, geográficas y psicográficas, así como tasas de consumo. Los estudios descriptivos determinan las percepciones del comprador acerca de las características de los productos.

Una investigación descriptiva se caracteriza por la enunciación clara del problema de decisión, objetivos específicos de investigación y necesidades de información detalladas. Se caracteriza, además, por un diseño de investigación cuidadosamente planeado y estructurado.

Explicativa

Este tipo de investigación se aplicará en el estudio del planteamiento del problema ya que me permitió determinar las causas que generaron la problemática por pérdida de la información, y de igual manera el efecto que tuvieron las mismas con el desarrollo del proyecto, que es la implementación de la plataforma virtual tres capas para la gestión de proyectos; normalmente pudiendo contestar las interrogantes porque ocurrió el fenómeno de pérdida de la información, tiempos de espera prolongados de los clientes, y a su vez la misma relacionándola con los motivos de quien lo generó a fin de poder descubrir las causas y efectos que motivaron la ejecución de los mismos. Igualmente en la aplicación de este tipo de investigación que se fundamenta en la observación directa permitiendo determinar los conocimientos adquiridos en forma teórica y poderlos materializar en una forma práctica a través de las soluciones planteadas y de soluciones informáticas; aportes a solucionar el problema planteado.



Histórica

Este tipo de investigación es muy utilizada ya que me permite analizar todos los eventos que se llevaron a cabo en el diseño y construcción del sistema original en el caso de existir y por otro lado en los procesos manuales que se realizaban, en el caso de no contar con un sistema. Este análisis por memorizado de información relevante de origen, proceso y destino de la misma y quien se vara afectada por la sistematización de tareas y simplificación de procesos que se ejecutaban.

Por otro lado este tipo de investigación también facilita la relación que va a existir con los procedimientos actuales, lo que significa que realizando un análisis de procesos y procedimientos anteriores estos puede adquirir un valor agregado, lo que me permitirá discernir qué tipo de información es la que tiene que ser automatizada y que información es la que debe ser relacionada y qué tipo de información tiene que ser almacenada y visualizada; bajo esta premisas podemos establecer la importancia que tiene el tipo de información histórica en el análisis de información y restructuración de eventos del pasado con los que estamos ejecutando actualmente y que requieren mucha atención en la recopilación detallada de datos y el análisis por memorizado de su integración a los módulos automatizados.

3.2 Métodos De Investigación

3.2.1 Métodos Teóricos

Estos métodos de investigación se aplica en su totalidad a mi proyecto de Desarrollo e Implementación del Control de Proyectos de Grado, ya que con la finalidad de extraer datos e información de mismo, para ello tengo que recurrir a fuentes de información que por tratarse de un asunto técnico están almacenados en una categoría especial, la recopilación



de información necesaria de libros técnicos, manuales, revistas, catálogos, me permitirán analizar los procesos y procedimientos propios del sistema que tengo que desarrollar, igualmente me permitirá conocer el origen de la información, el flujo que sigue la misma, y donde necesita ser almacenado, mediante un diseño lógico y coherente materializare la información recopilada de los documentos antes mencionados para la elaboración de este proyecto de grado.

Igual tratamiento se lo realizara con la información que obtenga a través de CD, Internet y material magnético que puedan aportar al desarrollo estructuración de mi proyecto.

Cabe recalcar que toda información referente a este tema, será analizado profundamente antes de ser considerado como valido, ya que la estructuración de un sistema informático depende mucho de la toma de datos en tiempo real la estructuración de la misma, se realizará a través de modelos lógicos que son de entendimiento para los usuarios programadores como, la generación de base de datos que finalmente darán como resultado sistemas informáticos consistentes, rápidos, y confiables

Método Deductivo

Es un tipo de razonamiento que nos lleva:

- a) De lo general a lo particular.
- b) De lo complejo a lo simple.

Pese a que el razonamiento deductivo es una maravillosa herramienta del conocimiento científico, si el avance de la ciencia se diera sólo en función de él, éste sería muy pequeño. Esto se debe a que nuestra experiencia como humanos es limitada, depende de nuestros sentidos y de nuestra memoria.

Este método va de lo general a lo particular, y me permite analizar la información de una mejor manera la interpretación del tema denominado Desarrollo e Implementación del Control de Proyectos de Grado, lo general



constituye el almacenamiento de la información en una bases de datos estructurada, la misma que para su estructuración surgió un proceso de análisis, diseño que se definieron ordenadamente de acuerdo a los tipos de datos que van a ser almacenados así como la cantidad de información a ser receptada en la elaboración de este proyecto.

Método Inductivo

Es un modo de razonar que nos lleva:

- a) De lo particular a lo general.
- b) De una parte a un todo.

Inducir es ir más allá de lo evidente. Dado que este método de investigación es muy importante, podemos decir que lo particular sería la información que podemos sacar de provecho de la empresa, en este caso realizando encuestas que nos permitirán recolectar desde la más mínima información para llegar a lo general que en este caso sería nuestro proyecto, razón por la cual para ello utilizaremos este método para realizar este proyecto de una manera ordenada y segura.

El razonamiento inductivo es una modalidad del razonamiento no deductivo que consiste en obtener conclusiones generales a partir de premisas que contienen datos particulares, el cual lo particular sería la obtención de la mínima información para almacenarla en la base de datos, para concluirla con lo general que en este caso sería, la elaboración de todo el proyecto.

Método Histórico Lógico

El método Histórico lógico se aplica a mi proyecto, desarrollo de una plataforma virtual tres capas para la gestión de proyectos. Ya que con la ayuda del mismo me ha permitido validar todos los procesos y procedimientos que se encuentran materializados en un plan y que me permiten reproducir cronológicamente toda la sesión de tareas que se tienen que ejecutar



en un macro proceso; al mismo tiempo me permite definir claramente el comportamiento de cada uno de los procesos, es decir verificando el alcance que tiene cada uno de ellos, en la realización de determinada tarea, y al mismo tiempo definida en un tiempo de ejecución. Por consiguiente este método me ayudo a delimitar las áreas automatizables y al mismo tiempo las actividades, que tienen que cumplir cada uno de los actores involucrados.

Método Analítico Sintético

El método analítico sintético se aplica al proyecto, desarrollo de una plataforma virtual tres capas para la gestión de proyectos, ya que con el mismo me permite realizar un análisis a profundidad de cada uno de los elementos que intervienen en el desarrollo de un software informático es decir una metodología diseño estructuración de base de datos, etc. lo que llevado a la incorporación de un lenguaje de programación van a permitir la sistematización en la organización; por otro lado al realizar una síntesis de la información teórica y conceptual nos permitirá reunir todos los argumentos válidos a fin de poder estructurar toda la lógica didáctica y conceptual de todo elemento que interviene en el desarrollo de mi tema desarrollo de una plataforma virtual tres capas para la gestión de proyectos, pudiendo concentrar el mayor esfuerzo en la búsqueda de información y análisis de la misma pero únicamente con la salvedad de que sea aplicable estrictamente al tema en ejecución. Por consiguiente el método analítico y sintético tiene su fundamentación de aplicación en cómo vamos a tratar la información teórica necesaria para la estructuración del capítulo segundo de este tema de proyecto.

Método Analítico Sistémico

Este método nos conduce a la estructuración lógica del proyecto, desarrollo de una plataforma virtual tres capas para la gestión de proyectos, ya que tendrán que seguir un orden definido en la estructuración de cada capítulo lo que permitirá a cualquier lector ir verificando paso a paso el análisis de la problemática que objetivos se persigue como voy a



aplicar mi investigación y finalmente en el desarrollo de un software informático que será la vinculación de las partes constitutivas del proyecto y al mismo tiempo el objetivo general que se consigue al automatizar todos los procesos y procedimientos de la entidad.

3.3 Herramientas de Recolección de Información

Las Herramientas o Instrumentos son los medios materiales que se emplean para recoger y almacenar la información. Para recoger datos e información relevantes, la investigadora utilizó como instrumentos de recolección de datos el guión de entrevista, el registro de observación documental e igualmente el registro del diario de observación directa, cuyos resultados fueron satisfactorios. El guión de entrevista estuvo conformado de preguntas relacionadas directamente con el objetivo general y los objetivos específicos de la investigación. Mediante el guión de entrevista se obtuvo información cualitativa

Las técnicas de recolección de datos son las distintas formas de obtener información. Las técnicas de recolección de datos que fueron utilizadas en la presente investigación son la observación directa y la entrevista.

Observación

La observación consiste en el registro sistemático, cálido y confiable de comportamientos o conductas manifiestas. En relación a la observación se hace "a través de formularios, los cuales tienen aplicación a aquellos problemas que se pueden investigar por métodos de observación, análisis de fuentes documentales y demás sistemas de conocimiento. La observación se realizó en nuestro caso por la necesidad de implementar un sistema de control de proyectos (Tesis), automatizando de esta manera los procesos que comúnmente se habían estado llevando. Con este sistema se mejorara el orden y el control de todos los



proyectos de grado, de todas las escuelas del ITSCO dando así un mejor manejo de la información, más rápida, precisa y confiable a nuestra institución.

Entrevista

La entrevista es uno de los pasos más importante al que nos enfrentaremos a la hora de realizar nuestro proyecto. Todo este proceso tienen el único objetivo de conseguir llegar a una entrevista personal, así que si lo conseguimos nos servirá para seguir todas las instrucciones que saquemos de la encuesta y aplicarlas a nuestro proyecto.

Para ello realizamos las siguientes preguntas que son muy necesarias para cumplir con este propósito:

Pregunta:

1. Para que es necesario crear un sistema de revisión de Proyectos de Grado?

Respuesta:

Para poder determinar con exactitud la cantidad y calidad de proyectos que se ejecutan semestralmente en las diferentes escuelas igualmente para verificar la gestión que existe entre los temas asignados y su tutor.

Análisis:

Esto nos servirá para especificar el modelo que se diseñara y a su vez la estructuración de la base de datos para establecer las reglas del negocio

Pregunta:

2. La institución cuenta con un manual de procesos y procedimientos con respecto a esta temática?

Respuesta:

No se cuenta por el momento se está levantando.



Análisis:

Esto serviría para saber si contamos con procesos con los cual vamos a trabajar y si no los hay tendremos que empezar a levantarlos.

Pregunta:

3. Los procesos que se están definiendo para el presente proyecto serán validados e integrados a la institución?

Respuesta:

Sí ya que este es uno de los primeros macro procesos que se están realizando y por ende mantiene procesos y procedimientos apegados a la realidad institucional

Análisis:

Esto serviría para saber dónde aplicaremos el proyecto con todos sus respectivos procesos.

Pregunta:

4. Se asignara una ip publica para este proyecto o será parte de los sistemas ya existentes?

Respuesta:

Como el desarrollo de un sistema nuevo implica un grado de madures es necesario que funcione como una aplicación independiente y que a medida que el mismo obtenga la madurez necesaria pase a ser parte de la sistematización de la Institución.

Análisis:

Esto nos servirá para poder saber si el sistema se va a aplicar sobre el actual sistema o será independiente del actual sistema.

Pregunta:

- 5.Cuál es el front end y el back end estándar de la Institución?

Respuesta:

El back end es Oracle y el front end es Java.



Análisis:

Esto nos permitirá saber con qué front end y back end cuenta la Institucional la realización del Sistema.

Pregunta:

6. Las interfaces de diseño serán entregadas por la institución o tendremos que desarrollarlas?

Respuesta:

Es preferible desarrollarlas nosotros mismos ya que va a funcionar como un aplicativo independiente pero su desarrollo debe estar apegado a los estándares que maneja el Instituto.

Análisis:

Esto nos servirá para saber si las interfaces de usuario nos entregara el Instituto o tendremos que hacerlas por nuestra propia cuenta.

Conclusión

La base de todo nuestro proyecto tiene como finalidad desarrollar un sistema para el Instituto Tecnológico Superior Cordillera, el cual encierra un conjunto de actividades y procesos mediante las preguntas realizadas en la encuesta, que desarrolladas secuencialmente de una forma eficiente y planificadas correctamente con la autoridad, busca, como principal objetivo cumplir con todo este sistema para nuestro instituto, y también de esta manera suministrar una base de datos para construir un software de alta calidad en una forma eficiente, y así facilitar el proceso de control de proyectos que satisfaga las necesidades de todo el Instituto Tecnológico Superior Cordillera.



CAPITULO IV

DESARROLLO DE LA PROPUESTA

4.1 Diagnóstico Situacional

Actualmente el Instituto Tecnológico Superior Cordillera, cuenta con un sistema de control de proyectos 2 capas (Cliente - Servidor), el cual no es muy eficiente, es por ello que tanto, los directores, los tutores, y lectores, deben estar en el Instituto, obligatoriamente para hacer el control de los proyectos de grado, es por ello la necesidad de implementar un nuevo sistema, tres capas que se permita utilizarlo mediante la web, facilitando así, la utilización del sistema en cualquier parte del mundo mediante el internet.

El control de los proyectos es muy fundamental en nuestro Instituto, ya que cada alumno deberá escoger un tema para poder desarrollarlos, vemos inconvenientes en controlar estos temas ya que hay una gran cantidad de alumnos, es por ello la necesidad de implementar nuestro sistema.

Razón por la cual todo este proceso de control no está totalmente automatizado, lo cual queremos que se realice de una manera mucho más fácil, sencilla, ordenada y rápida el control de los proyectos que realizan los estudiantes para graduarse.

La base de todo nuestro proyecto tiene como finalidad desarrollar un sistema para el Instituto Tecnológico Superior Cordillera, el cual encierra un conjunto de actividades y procesos, que desarrolladas secuencialmente de una forma eficiente y planificadas correctamente con autoridad, busca, como principal objetivo, suministrar una base de datos para construir un software de alta calidad en una forma eficiente, y así facilitar el proceso de control de proyectos que satisfaga las necesidades de todo el Instituto Tecnológico Superior Cordillera.



Otro de los problemas que tiene nuestra Institución es el control de avances de proyectos, ya que cada estudiante debe presentar el avance en una fecha asignada por el profesor de proyectos, y mientras no se entregue dicho avance no podrá presentar los siguientes, por medio de esto se lograra controlar de una manera más ordenada, todos los avances de proyectos por alumnos.

Las tutorías que realizan los directores del Instituto son deficientes, por la razón de que no tienen un control adecuado de las tutorías que está manejando cada director, por ello en nuestro sistema se lograra controlar este problema de una manera más ordenada.

También de acuerdo a nuestro análisis e investigación de este problema, pudimos observar que no pueden controlar la cantidad de alumnos que están presentando los proyectos, y también por escuelas.

Otra de las deficiencias y dificultades que tiene nuestro Instituto es la de controlar los temas repetidos que en cada escuela se presentan, para ello con nuestro sistema se lograra controlar toda esta redundancia de información, que anteriormente era una gran dificultad saber cuántos alumnos estaban presentando los mismos temas.

Con todo este proceso ordenado en nuestro sistema no solo lograremos controlar los temas que están repetidos, sino también el evaluar la cantidad de temas que existen y ver si están o no aprobados, tanto por los directores, como también por el jefe de proyectos.

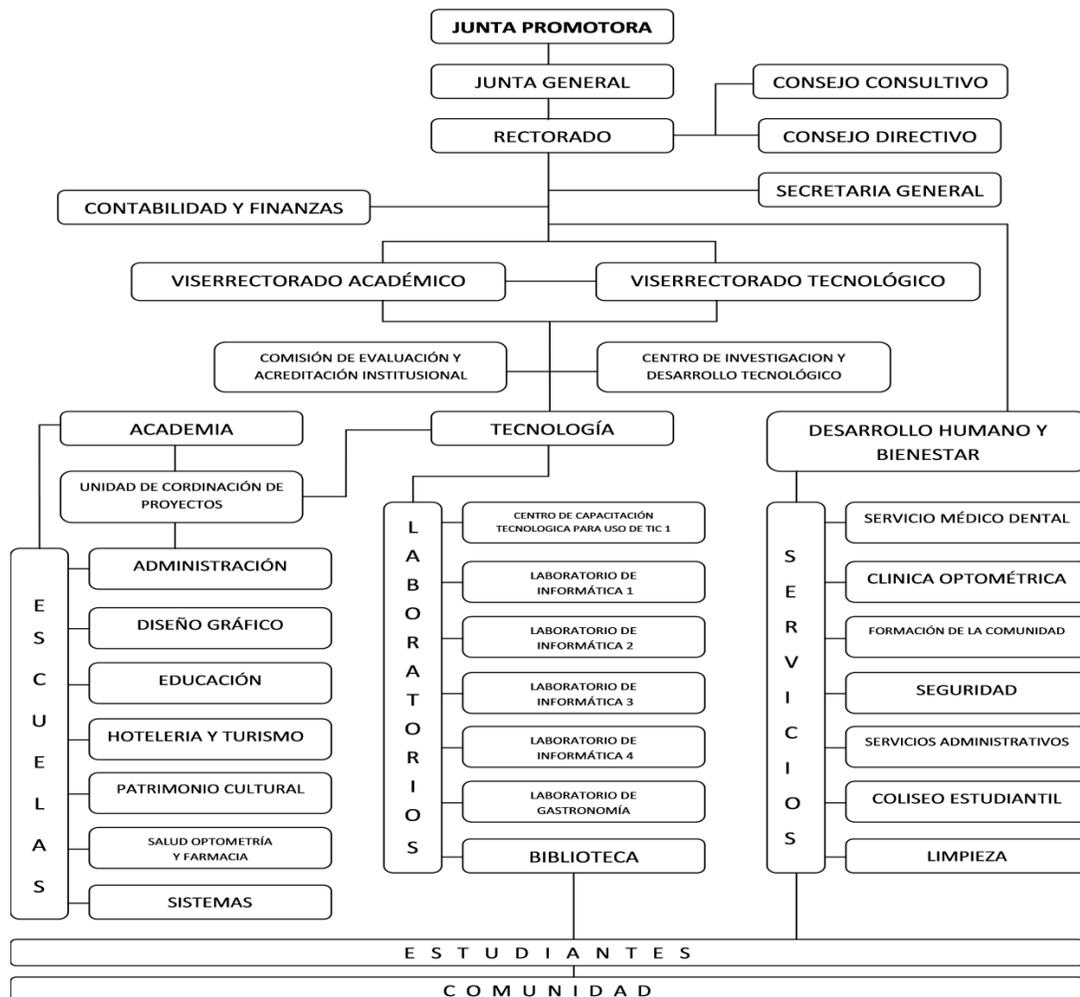
Otra causa que se puede destacar en el Instituto es también que muchos temas son enfocados en solo ámbito, es decir no existe diversidad de temas por lo que también es muy importante controlar.

4.2 Estructura Organizacional Jerárquica



INSTITUTO TECNOLÓGICO SUPERIOR “CORDILLERA”

El presente proyecto está desarrollado con la finalidad de fomentar un mejor servicio para controlar los proyectos de tesis de cada estudiante. El Instituto cuenta con una estructura Organizacional Jerárquica que va desde el Rector, Vicerrector, Directivos, Directores, Lectores, y Tutores, que deben estar al tanto, en los avances del Proyecto, Tanto en los capítulos, como también en el desarrollo físico del mismo. Razón por la cual todo este proceso de control no está totalmente automatizado, lo cual queremos que se realice de una manera mucho más fácil, sencilla, ordenada y rápida el control de los proyectos que realizan los estudiantes para graduarse.



Nombre:Diagrama Organizacional del Instituto Cordillera.



Gráfico: N.- 03.

Fuente: Instituto Cordillera.

4.3 Infraestructura Informática

4.3.1 HARDWARE

<i>Características</i>	<i>Ubicación</i>	<i>Arquitectura</i>
<i>Servidores Torre</i>	<i>Servidor</i>	<i>CISCO SISTIMAX</i>
<i>Servidores Rack</i>	<i>Servidor</i>	<i>CISCO SISTIMAX</i>

Tabla No1 Hardware

4.3.2 Software

Software	Ubicación
My SQL	Base de Datos
Centos 5.5	Sistema Operativo

Tabla No2 Software

4.3.3 Comunicaciones

Equipo	Ubicación
Router	Servidor
SwitchCatalist 2966	Servidor
SwitchCatalist 3566	Servidor
Firewall ASA	Servidor

Tabla No3 Comunicaciones

4.3.4 Recurso Humano Técnico

Nombre	Descripción
Fernando Jarrín	Alumno
Víctor Chiza	Alumno
Ing. Octavio Córdor	Administrador de Red
Ing. Jaime Basantes	Tutor

Tabla No4 Recurso Humano Técnico

4.4 Descripción De Alternativas



A fin de poder evaluar las alternativas se han desarrollado unos ítems de especificaciones técnicas, que se necesita desarrollar, con la finalidad de implementar el prototipo, también son necesarias garantías, soporte técnico que se pueden ofrecer.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		
My SQL		
Modelo Negocio*		
Casos de Uso		
Clases		
Componentes		
etc.		
WEB SERVICE		
ASP.NET C#		
WEB		
Implementación		
Secuencia		
Estado		
Colaboración		
Capacitación Usuario		
Manual de Usuario		
METODOLOGIA		
RUP		
FronEnd*		
Punto net (C#)		
Manuales Técnicos		
Diccionario de Datos		
Pruebas de Funcionamiento		
Integración		
Validación		
Seguridades		
Interfaz Gráfica		
Estándares Programación		
Variables		
Constantes		
Controles Visuales		



Código Fuente Comentado		
-------------------------	--	--

Tiempo de Desarrollo		
Garantía Técnica		
Soporte Técnico		
Costo		
Código Fuente		

Tabla No5 Descripción de Alternativas

4.4.1 Nombre Alternativa 1

La primera alternativa para desarrollar este tipo de proyecto, es la empresa NETLINE.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		
My SQL	x	
Modelo Negocio		
Casos de Uso		x
Clases		x
Componentes		x
Plataforma		
ASP Punto Net	x	
WEB	x	
Implementación		
Secuencia	x	
Estado	x	
Colaboración	x	
Capacitación Usuario		
Manual de Usuario		x
Metodología		
RUP	x	
FronEnd		
Punto net (C#)	x	
Manuales Técnicos		
Diccionario de Datos	x	
Pruebas de Funcionamiento		



Integración	x	
Validación	x	
Seguridades	x	
Interfaz Gráfica	x	
Estándares Programación		
Variables	x	
Constantes	x	
Controles Visuales	x	
Código Fuente Comentado	x	

Tiempo de Desarrollo	6 meses	
Garantía Técnica	6 meses	
Soporte Técnico	2 meses	
Costo	\$ 2500	
Código Fuente	si	

Tabla No6 Descripción de Alternativas 1

4.4.2 Nombre Alternativa 2

La segunda alternativa para desarrollar este tipo de proyecto, es la empresaviasoft

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		
My SQL	x	
Modelo Negocio		
Casos de Uso	x	
Clases	x	
Componentes		x
Plataforma		
ASP Punto Net	x	
WEB	x	
Implementación		
Secuencia		x
Estado		x
Colaboración		x
Capacitación Usuario		



Manual de Usuario	x	
Metodología		
RUP	x	
FronEnd		
Punto net (C#)		x
Manuales Técnicos		
Diccionario de Datos	x	
Pruebas de Funcionamiento		
Integración	x	
Validación	x	
Seguridades	x	
Interfaz Gráfica	x	
Estándares Programación		
Variables	x	
Constantes	x	
Controles Visuales	x	
Código Fuente Comentado	x	

Tiempo de Desarrollo	8 meses	
Garantía Técnica	4 meses	
Soporte Técnico	4 meses	
Costo	\$ 2000	
Código Fuente	si	

Tabla No7 Descripción de Alternativas 2

4.4.3 Nombre Alternativa 3

La tercera alternativa para desarrollar este tipo de proyecto, es la empresa NETLIFE.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		
My SQL	x	
Modelo Negocio		
Casos de Uso	x	
Clases		x



Componentes	x	
Plataforma		
ASP Punto Net	x	
WEB	x	
Implementación		
Secuencia		x
Estado		x
Colaboración		x
Capacitación Usuario		
Manual de Usuario	x	
Metodología		
RUP	x	
FronEnd		
Punto net (C#)	x	
Manuales Técnicos		
Diccionario de Datos	x	
Pruebas de Funcionamiento		
Integración	x	
Validación	x	
Seguridades	x	
Interfaz Gráfica	x	
Estándares Programación		
Variables	x	
Constantes	x	
Controles Visuales	x	
Código Fuente Comentado	x	

Tiempo de Desarrollo	15 meses	
Garantía Técnica	8 meses	
Soporte Técnico	8 meses	
Costo	\$ 1500	
Código Fuente	si	

Tabla No8 Descripción de Alternativas 3

4.5 Evaluación Y Selección De Alternativas



Con la finalidad de poder evaluar las dos alternativas presentadas se determinaron algunos parámetros para la calificación de cada uno de los ítems presentados y de acuerdo a la importancia en la construcción de los mismos, por consiguiente se determinaron de la siguiente manera:

- | | | |
|----|----------------------|------|
| 1. | Evaluación técnicas | 70 % |
| 2. | Evaluación Económica | 20 % |
| 3. | Garantía técnica | 5 % |
| 4. | Soporte técnico | 5 % |

ACTIVIDAD	Pesos	NETLINE	SAVIASOFT	NETLIFE
Back End				
My SQL	5	4	3	5
Modelo Negocio				
Casos de Uso	2	1	5	2
Clases	2	1	5	2
Componentes	3	2	7	3
Plataforma				
ASP Punto Net	13	10	6	13
WEB	15	13	1	15
Implementación				
Secuencia	5	4	1	5
Estado	5	4	1	5
Colaboración	1	2	1	1
Capacitación Usuario				
Manual de Usuario	9	8	2	9
Metodología				
RUP	5	4	3	5
FronEnd				
Punto net (C#)	5	6	3	5
Manuales Técnicos				
Diccionario de Datos	5	6	3	5
Pruebas de Funcionamiento				
Integración	5	2	5	0



Validación	5	2	5	5
Seguridades	5	4	6	5
Interfaz Gráfica	5	4	5	5
Estándares Programación				
Variables	1	2	4	1
Constantes	1	2	2	1
Controles Visuales	1	2	2	1
Código Fuente Comentado	2	1	3	2
Total	100	84	73	95

Tabla No9 Evaluación y Selección de Alternativas

Tabla de costos

Empresa	Costo	Porcentaje %
NETLINE	2500	20
SAVIASOFT	2000	15
NETLIFE	1500	10

Tabla No10 Tabla de Costos

Tabla de Garantías

Empresa	Tiempo	Porcentaje %
NETLINE	6 meses	10
SAVIASOFT	8 meses	15
NETLIFE	15 meses	20

Tabla No11 Tabla de Garantías

Tabla de Soporte Técnico

Empresa	Tiempo	Porcentaje %
---------	--------	--------------



NETLINE	6 meses	15
SAVIASOFT	4 meses	10
NETLIFE	8 meses	20

Tabla No12 Soporte Técnico

Realizando la evaluación definitiva de las dos propuestas se puede concluir claramente que la empresa NETLINE alcanzo los siguientes resultados, en la parte técnica sumo 84 puntos lo por lo que se le asigna un puntaje de 33.33 puntos, en la evaluación económica, soporte y garantía técnica un valor de 60 puntos por consiguiente obtiene un resultado general de **84 puntos**.

Realizando la evaluación definitiva de las dos propuestas se puede concluir claramente que la empresa SAVIASOFT alcanzo los siguientes resultados, en la parte técnica sumo 73 puntos lo por lo que se le asigna un puntaje de 28.96 puntos, en la evaluación económica, soporte y garantía técnica un valor de 60 puntos por consiguiente obtiene un resultado general de **73 puntos**.

Lo referente a la calificación de la propuesta presentada por la empresa NETLIFE es la siguiente en la evaluación técnica obtuvo un puntaje de 100, por lo que se le asigna un valor de 95 puntos, en la parte económica y garantía y soporte técnico alcanzo un puntaje de 37.69 puntos, dando un total de **95 puntos**.

4.6 Factibilidad Técnica

De lo expuesto anteriormente se desprende claramente que la opción de desarrollo propuesto como proyecto de tesis de grado, es la más conveniente para realizar el desarrollo del sistema, tanto en lo técnico como el económico.

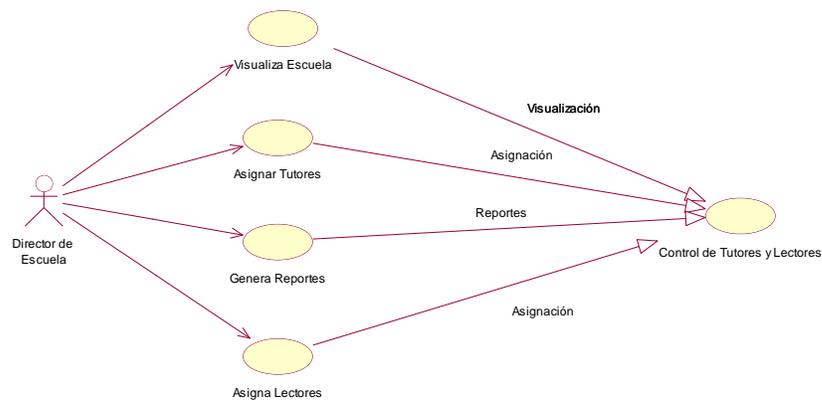
La misma establece que el aspecto técnico es la principal para poder realizar una calificación coherente y acertada, por otro lado es muy conveniente por el aspecto económico, lo que se traduce en la conveniencia de la realización con la segunda opción que es la más adecuada



en todos los aspectos, estableciendo claramente su factibilidad en tiempo y aspecto técnico de acuerdo a las evaluaciones realizadas en el ítem anterior.

4.7 Descripción De Procesos.

Asignación de Tutores



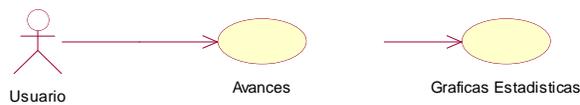
Nombre: Asignación de Tutores.

Gráfico: N.- 04.

Fuente: Víctor Chiza.

Este proceso cumple con la finalidad de asignar los tutores y a los lectores a los alumnos de cada escuela, y a su vez asignar los temas que debe controlar el tutor a sus tutorados.

Procesos Estadísticos



Nombre: Procesos Estadísticos.

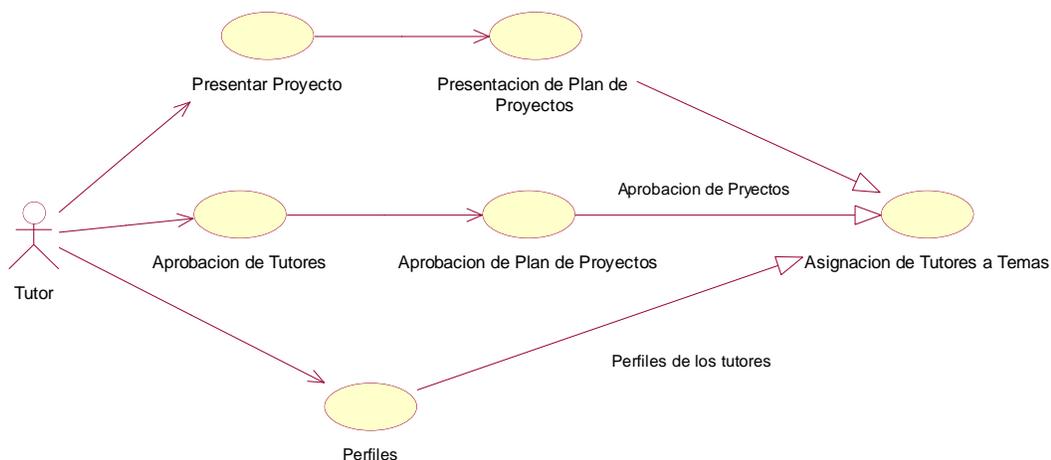
Gráfico: N.- 05.

Fuente: Víctor Chiza.



Este es un proceso principal que hace nuestro sistema, mediante el cual podemos determinar, los avances de cada alumno, los avances que tienen también por escuela, y a su vez un avance global de todo el Instituto.

Procesos de Tutores



Nombre: Procesos de Tutores.

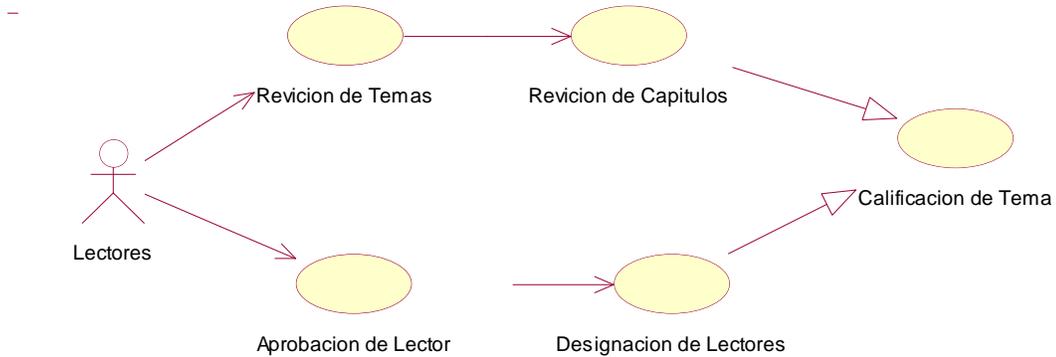
Gráfico: N.- 06.

Fuente: Víctor Chiza.

Los Tutores son los encargados de dirigir el proyecto de los alumnos tanto en el desarrollo como en el diseño de los procesos del proyecto, en nuestro sistema el tutor es el encargado de revisar los avances de cada capítulo de la Tesis, el cual esta validado por fechas de entrega de cada capítulo, que a su vez los capítulos también son calificados, el tutor también puede generar un reporte estadístico del avance de los capítulos, que el tutor esta tutorando.



Procesos del Lector



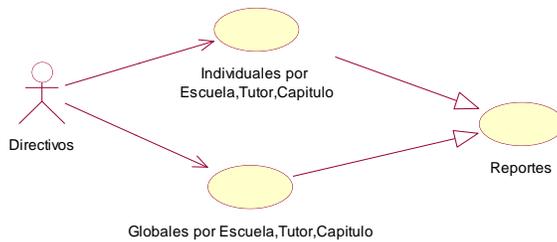
Nombre: Proceso de Lector.

Gráfico: N.- 07.

Fuente: Víctor Chiza.

Los lectores son los encargados de calificar el capítulo que fue revisado antes por el tutor, a su vez ellos tienen una fecha para poder emitir la calificación, también pueden sacar reportes de los alumnos que ha revisado y emitido una calificación.

Procesos de Directivos



Nombre: Proceso de Directivos.

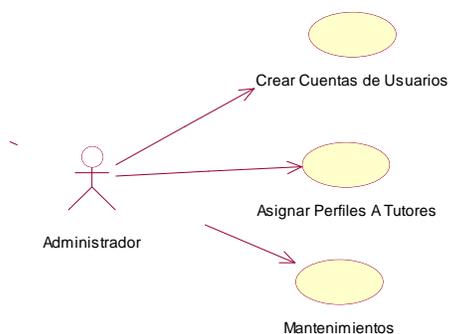
Gráfico: N.- 08.

Fuente: Víctor Chiza.



Los directivos pueden visualizar el avance estadístico de todas las escuelas, para ver el avance de los proyectos, a su vez pueden sacar un reporte estadístico de ello para verificar así el avance que tiene hasta culminar el semestre.

Proceso de Administrador



Nombre:Proceso de Administrador.

Gráfico: N.- 09.

Fuente: Víctor Chiza.

El Administrador crea cuentas de los usuarios asignando los roles respectivos a cada usuario

Procesos de Reportes



Nombre:Procesos de Reportes.

Gráfico: N.- 10.

Fuente: Víctor Chiza.

Los reportes se realizan sacando los valores estadísticos tomados de las calificaciones que emite tanto el tutor como el lector de cada escuela.



4.8 Descripción De Metodología De Desarrollo

PROCESO RACIONAL UNIFICADO (RUP)

(RationalUnifiedProcess en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el RationalMethodComposer (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el RationalUnifiedProcess, que se vendiera como producto independiente.

FASES DE METODO RUP

- Establece oportunidad y alcance
- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

'**Proceso**': Las etapas de esta sección son: (Revise nuevamente la gráfica)

- Modelado de negocio



- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

Soporte: En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio(También llamado Incepción o Concepción)
- Elaboración
- Desarrollo(También llamado Implementación, Construcción)
- Cierre (También llamado Transición)

Fase de Inicio: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la



especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de Desarrollo: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

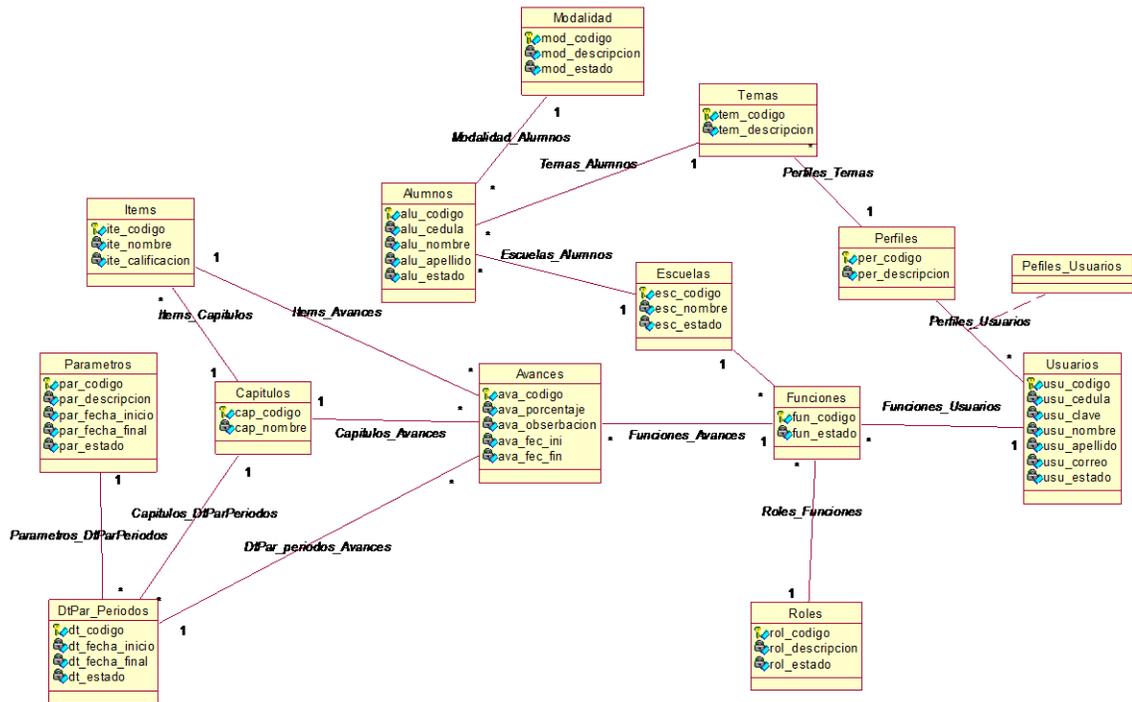
Fase de Cierre: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario.

4.9 Modelo Lógico

El modelo lógico describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. Sirve para el administrador, para decidir qué datos se van a guardar en la Base, en este modelo se ha diseñado las tablas que Utiliza el Sistema de Control de Tesis. En este modelo se ha diseñado las tablas que mediante la investigación y a su vez observando las necesidades de nuestro Instituto hemos diseñado la Base de Datos, el cual consta con las siguientes tablas: alumnos, avances, capítulos, departamento – periodos, escuelas, funciones, ítems, modalidad, parámetros, perfiles, Perfil - usuario, roles, temas, tema – autor, usuarios. En donde la tabla principal, es avances la cual está relacionada con la tabla alumnos esta tabla nos permitirá realizar la consulta de los alumnos con su avance en cuanto al proyecto controlando así cada ítem de cada capítulo de la tesis, otra de las tablas con mucha importancia es también de ítems, ya que en esta tabla consta de todos los ítems que tiene cada capítulo, que a su vez cumple con la función de que el tutor pueda calificar por ítems de capítulos el avance del proyecto, que a su vez se podrá generar reportes, de



manera global, por escuela, y por alumno, controlando de esta manera el avance de los proyectos.



Nombre:Modelo Lógico.

Gráfico: N.- 11.

Fuente: Víctor Chiza.

4.10 Modelo Físico.

Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).



A la descripción de una base de datos mediante un modelo de datos se le denomina esquema de la base de datos. Este esquema se especifica durante el diseño, y no es de esperar que se modifique a menudo.

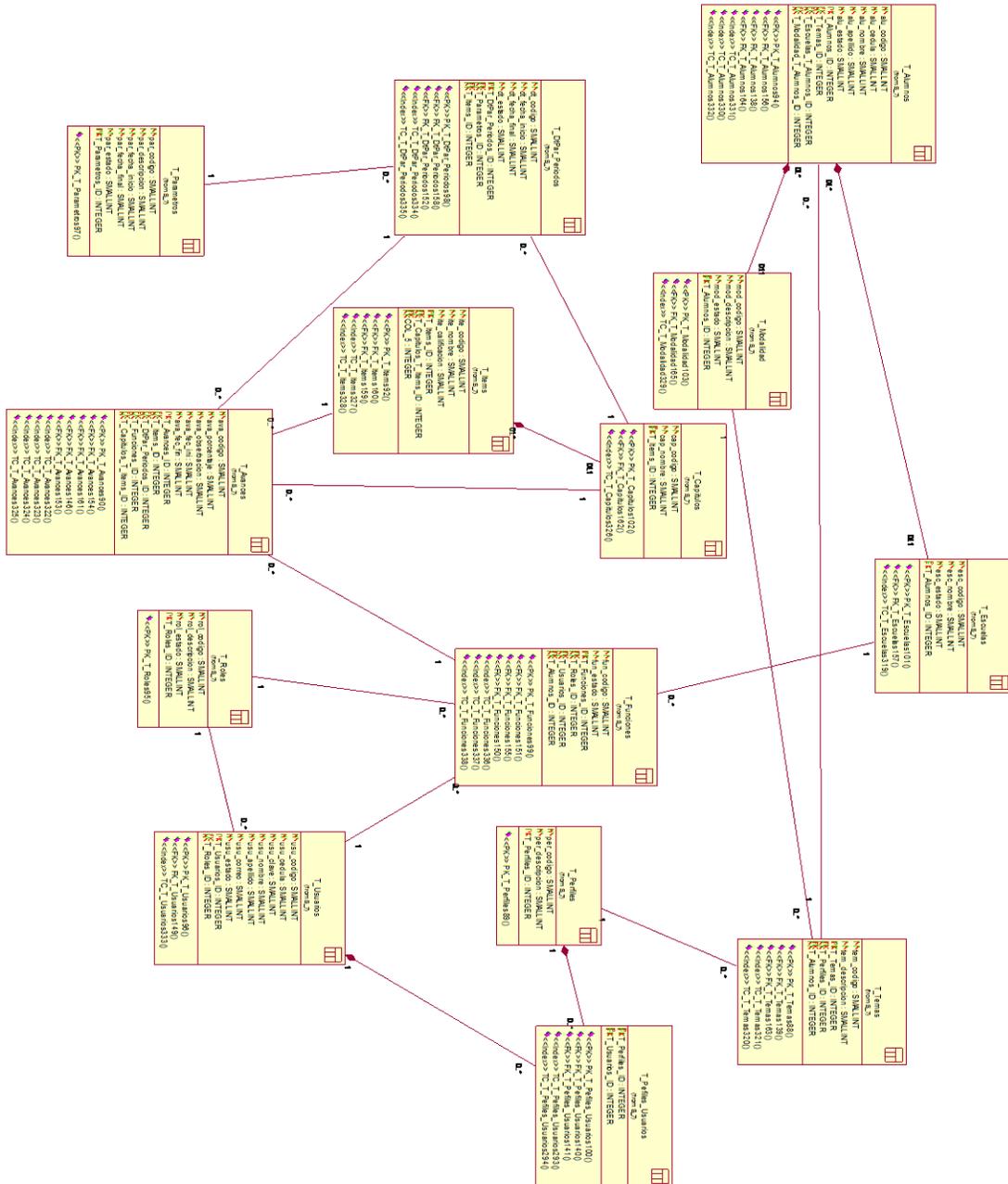
Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, etc. Los datos que la base de datos contiene en un determinado momento se denominan estado de la base de datos u ocurrencia de la base de datos.

La distinción entre el esquema y el estado de la base de datos es muy importante. Cuando definimos una nueva base de datos, sólo especificamos su esquema al Sistema de Gestión de Base de Datos. En ese momento, el estado de la Base de Datos es el "estado vacío", sin datos. Cuando se cargan datos por primera vez, la base de datos pasa al "estado inicial".

De ahí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado.

El Sistema de gestión de Base de Datos se encarga, en parte, de garantizar que todos los estados de la Base de Datos sean estados válidos que satisfagan la estructura y las restricciones especificadas en el esquema.

Por lo tanto, es muy importante que el esquema que se especifique al Sistema de Gestión de Base de Datos sea correcto y se debe tener muchísimo cuidado al diseñarlo el mismo.



Nombre:Modelo Físico.

Gráfico: N.- 12.

Fuente: Víctor Chiza.



4.11 Diccionario De Datos.

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

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

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

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

Lista de Entidades

Name	Code	Parent	Generate	Number
Alumnos	ALUMNOS	Logical Data Model 'itsco'	X	
Avances	AVANCES	Logical Data Model 'itsco'	X	
Capitulos	CAPITULOS	Logical Data Model 'itsco'	X	
DtPar_Periodos	DTPAR_PERIODOS	Logical Data Model 'itsco'	X	
Escuelas	ESCUELAS	Logical Data Model 'itsco'	X	
Funciones	FUNCIONES	Logical Data Model 'itsco'	X	
Items	ITEMS	Logical Data Model 'itsco'	X	



Modalidades	MODALIDADES	Logical Data Model 'itsco'	X	
Parametros	PARAMETROS	Logical Data Model 'itsco'	X	
Perfiles	PERFILES	Logical Data Model 'itsco'	X	
PerfilUsuario	PERFILUSUARIO	Logical Data Model 'itsco'	X	
Roles	ROLES	Logical Data Model 'itsco'	X	
Temas	TEMAS	Logical Data Model 'itsco'	X	
TemaTutor	TEMATUTOR	Logical Data Model 'itsco'	X	
Usuarios	USUARIOS	Logical Data Model 'itsco'	X	

Tabla No13 Lista de Entidades

Lista de Atributos de Entidades

Name	Code	Data Type	Mandato ry	InheritedFrom
esc_codigo	ESC_CODIGO	Integer	X	
esc_nombre	ESC_NOMBRE	Variable characters (50)	X	
esc_estado	ESC_ESTADO	Variable characters (10)		
alu_codigo	ALU_CODIGO	Integer	X	
alu_cedula	ALU_CEDULA	Variable characters (10)		
esc_codigo	ESC_CODIGO	Integer		
mod_codigo	MOD_CODIGO	Integer		
alu_nombre	ALU_NOMBRE	Variable characters (50)	X	
alu_apellido	ALU_APELLIDO	Variable characters (50)	X	



alu_estado	ALU_ESTADO	Variable characters(10)	
usu_codigo	USU_CODIGO	Integer	X
usu_cedula	USU_CEDULA	Variable characters (10)	X
usu_clave	USU_CLAVE	Variable characters (50)	X
usu_nombre	USU_NOMBRE	Variable characters (50)	X
usu_apellido	USU_APELLIDO	Variable characters (50)	X
usu_telefono	USU_TELEFONO	Variable characters (10)	X
usu_correo	USU_CORREO	Variable characters (50)	X
usu_estado	USU_ESTADO	Variable characters (10)	
fun_codigo	FUN_CODIGO	Integer	X
usu_codigo	USU_CODIGO	Integer	
rol_codigo	ROL_CODIGO	Integer	
esc_codigo	ESC_CODIGO	Integer	
fun_estado	FUN_ESTADO	Variable characters (10)	
rol_codigo	ROL_CODIGO	Integer	X
rol_descripcion	ROL_DESCRIPCION	Variable characters (50)	X
rol_estado	ROL_ESTADO	Variable characters (10)	
tem_codigo	TEM_CODIGO	Integer	X
per_codigo	PER_CODIGO	Integer	
fun_codigo	FUN_CODIGO	Integer	
alu_codigo	ALU_CODIGO	Integer	
tem_descripcion	TEM_DESCRIPCION	Variable characters (100)	X
tem_calificacion	TEM_CALIFICACION	Variable characters (2)	X
tem_estado	TEM_ESTADO	Variable characters (10)	
per_codigo	PER_CODIGO	Integer	X
per_descripcion	PER_DESCRIPCION	Variable characters (50)	X
per_estado	PER_ESTADO	Variable characters (10)	
ava_codigo	AVA_CODIGO	Integer	X
ite_codigo	ITE_CODIGO	Integer	
fun_codigo	FUN_CODIGO	Integer	
cap_codigo	CAP_CODIGO	Integer	
tem_codigo	TEM_CODIGO	Integer	
dt_codigo	DT_CODIGO	Integer	



ava_porcentaje	AVA_PORCENTAJE	Integer	X
ava_observacion	AVA_OBSERVACION	Variable characters (100)	
ava_calificacion	AVA_CALIFICACION	Integer	
ava_estado	AVA_ESTADO	Variable characters (10)	
cap_codigo	CAP_CODIGO	Integer	X
cap_nombre	CAP_NOMBRE	Variable characters (50)	X
cap_estado	CAP_ESTADO	Variable characters (10)	
ite_codigo	ITE_CODIGO	Integer	X
cap_codigo	CAP_CODIGO	Integer	
ite_nombre	ITE_NOMBRE	Variable characters (100)	X
ite_estado	ITE_ESTADO	Variable characters (10)	
mod_codigo	MOD_CODIGO	Integer	X
mod_descripcion	MOD_DESCRIPCION	Variable characters (50)	X
mod_estado	MOD_ESTADO	Variable characters (10)	
p_u_codigo	P_U_CODIGO	Integer	X
per_codigo	PER_CODIGO	Integer	
usu_codigo	USU_CODIGO	Integer	
t_tu_codigo	T_TU_CODIGO	Integer	X
tem_codigo	TEM_CODIGO	Integer	
p_u_codigo	P_U_CODIGO	Integer	
t_tu_estado	T_TU_ESTADO	Variable characters (10)	
par_codigo	PAR_CODIGO	Integer	X
par_descripcion	PAR_DESCRIPCION	Variable characters (50)	X
par_fecha_inicio	PAR_FECHA_INICIO	Date	X
par_fecha_final	PAR_FECHA_FINAL	Date	X



par_estado	PAR_ESTADO	Variable characters (10)		
dt_codigo	DT_CODIGO	Integer	X	
cap_codigo	CAP_CODIGO	Integer		
par_codigo	PAR_CODIGO	Integer		
dt_fecha_inicio	DT_FECHA_INICIO	Date		
dt_fecha_final	DT_FECHA_FINAL	Date		
dt_estado	DT_ESTADO	Variable characters (10)		

Tabla No14Lista de Atributos de Entidades

4.12 Estándares De Programación

Nomenclatura

El nombre de los objetos de base de datos no deberá exceder los 35 caracteres y no deberá terminar en “_”. En el caso que el nombre esté compuesto por más de una palabra, las mismas tendrán que separarse por el signo “_” underscore.

Los nombres de objetos de base de datos deberán estar escritos en español, ser auto-descriptivos (nombres completos). El Área de involucramiento técnico asignará el nombre del proyecto, nombre de esquema y el prefijo de la aplicación a utilizarse en la nomenclatura de los objetos de base de datos, a fin de dar identidad al proceso de seguimiento y rastreo vehicular.

Documentación



Todos los objetos de base de datos y scripts deberán estar debidamente documentados en PowerDesigner, en las siguientes propiedades:

- **Comentario:** descripción clara del uso o funcionalidad del objeto. Cabe notar que esta documentación se verá reflejada en la base de datos.
- **Descripción:** este campo es opcional si el objeto tiene el campo comentarios y contendrá aclaraciones o ejemplos. Caso contrario es obligatorio y debe contener la descripción detallada del uso o funcionalidad del objeto.
- **Notas:** este campo se utilizará exclusivamente en el caso de requerirse la eliminación del objeto y contendrá los siguientes puntos:
 - Eliminación solicitada por:
 - Fecha de Eliminación:
 - Motivo de Eliminación:
 - Solución de Reemplazo: en el caso que aplique se debe detallar que solución reemplaza a la funcionalidad brindada por el objeto a ser eliminado.

Es importante que la documentación permita entender claramente el propósito del objeto y no que sea una simple repetición del nombre.

Tipos de Datos

Tipo de Dato	Cuando se debe utilizar
VARCHAR2	Para campos de texto de tamaño variable de hasta 5000 caracteres. Siempre se utilizará VARCHAR2 en lugar de VARCHAR.
CHAR	Para campos de texto de tamaño fijo, por ejemplo para el uso de estados (SI/NO), (ACT/INA).



NUMBER	Para campos numéricos. Siempre se especificará la precisión, por ejemplo para un número de 5 cifras enteras y 2 decimales será: NUMBER (7,2).
DATE	Para campos de fecha y de fecha y hora.
CLOB	Para campos de texto de más de 5000 caracteres.
BLOB	Para almacenamiento de archivos binarios, por ejemplo imágenes, archivos pdf, Word, etc.
XMLTYPE	Para campos en formato XML sobre los cuales se ejecutarán consultas con XPATH desde la Base de Datos. Si no se requerirá utilizar el motor de XML de la Base de Datos, es preferible utilizar un campo CLOB

Tabla No15 Tipo de Datos

Nombre:	[Prefijo de la aplicación] “_” [Nombre auto-descriptivo] Para roles de Discoverer: [Prefijo de la aplicación] “_DIS_” [Nombre auto-descriptivo]
Documentación	Comments: incluirá el perfil del usuario que utilizará el rol y de manera general la funcionalidad a la que tendrá acceso, por ejemplo: Para el rol RIG_AIR_GESTION: Rol utilizado por usuarios que realizan gestión sobre la presentación del anexo de otras retenciones. Permite la consulta de presentaciones de anexos realizadas por los contribuyentes.
Consideracion:	Toda aplicación deberá tener el rol: [Prefijo de la aplicación]_MANTENIMIENTO con



	permisos de select a todas las tablas del esquema de la aplicación.
--	---

Tablas

Nombre:	[Prefijo de la aplicación] “_” [Nombre auto-descriptivo en “singular”] De manera general el nombre auto-descriptivo de una tabla de rompimiento se conformará con la unión del nombre de las dos tablas padres.
Alias:	Se utilizará el alias generado por Designer. Si ya se encuentra en uso el prefijo asignado, se añadirá al final un número secuencial de 1 a 9.
Consideraciones:	Toda tabla tendrá una “clave primaria numérica de un solo campo” salvo mejor criterio de Diseño.

Tabla No16 Estándares de Tablas

Clave Primaria (Primary Key)

Nombre Campo:	“CODIGO_” [Nombre de la tabla en singular, sin prefijo de la aplicación, hasta donde alcance]
Nombre PK:	Utilizar el nombre asignado por Designer ([alias de la tabla] “_PK”)
Consideraciones:	La clave primaria debe ser numérica y de un solo campo, salvo mejor criterio de Diseño. Como caso de excepción se permitirá la creación de claves primarias compuestas, en el caso en que la mejor representación a nivel de entidades en un



	<p>diagrama de clases sea una relación de muchos a muchos y:</p> <ul style="list-style-type: none"> ➤ No se requiere registrar información adicional en la tabla de rompimiento ➤ No se requiere relacionar la tabla de rompimiento con ninguna otra tabla.
Documentación:	<p>Se requiere únicamente la documentación del campo en la tabla (no se requiere documentar el primarykey). En el caso de utilizar una secuencia, se colocará la referencia a la misma utilizando la sección SECUENCE de Designer.</p>

Tabla No17 Estándares de Campos Primarios

Campos

Nombre:	<p>El nombre de un campo deberá ser auto-descriptivo y estar en singular.</p> <p>Únicamente la clave primaria y los campos que provienen de un foreignkey pueden comenzar con la palabra CODIGO.</p> <p>El nombre no debe incluir valores permitidos, por ejemplo: ESTADO_ACTIVO_INACTIVO, lo correcto sería: ESTADO.</p>
Tipo de Dato:	<p>Revisar las consideraciones del punto “OTipos de Datos”</p>
Consideraciones:	<p>Todo campo con un conjunto de valores limitados que:</p> <p>a) Varían en el tiempo, harán referencia a la tabla o replica de la tabla correspondiente. Por ejemplo ID_VEHICULO debe tener un foreignkey hacia EVENTO, etc.</p>



	<p>Diseño analizará los casos de excepción en los que no es posible crear foreignkeys, por ejemplo:</p> <p>Tablas no replicadas. Tablas replicadas con tipo COMPLETE En tablas de log.</p> <p>b) Son fijos, tendrán obligatoriamente allowablevalues registrados en Designer de la siguiente manera: Value: tres letras representativas de cada palabra, por ejemplo:</p> <p>Para ACTIVO, INACTIVO, serían ACT e INA. Para ACEPTADO_MANUALMENTE y ACEPTADO_AUTOMATICAMENTE serían ACE_MAN y ACE_AUT. Para valores con menos de tres letras se colocará el nombre completo, por ejemplo: SI, NO, 0+.</p> <p>Meaning: nombre completo del valor permitido, por ejemplo ACTIVO, INACTIVO, etc.</p> <p>Caso de Excepción para MDA: en el caso de aplicaciones que cuenten con un modelo UML los allowablevalues deberán ser diseñados en UML a través de Enumeraciones y se cargarán en Designer como checkconstraints en lugar de allowablevalues. La documentación del meaning deberá ser incluida en el comment del checkconstraint, ejemplo: "ACT = Activo, INA = Inactivo"</p>
Documentación:	Comentario: se debe incluir la utilidad del campo y los



	<p>casos de uso, por ejemplo:</p> <p>Código de la Agencia principal del SRI, en la cual se reciben los trámites que llegan de otras oficinas</p> <p>Descripción: este campo es opcional y debe incluir aclaraciones y ejemplos respecto al contenido del campo.</p> <p>En el caso de utilizar una secuencia, se colocará la referencia a la misma utilizando la sección SEQUENCE de Designer.</p>
--	---

Tabla No18 Estándares de Campos

4.13 Pantallas Y Reportes (Prototipo)

LOGIN

INICIAR SESIÓN

Especifique su nombre de usuario y contraseña. [Regístrase](#) si no tiene una cuenta.

Información de cuenta

Nombre de usuario:

Contraseña:

Mantenerme conectado



Nombre de Usuario

Contraseña de Usuario

Nombre: Pantalla Login.

Gráfico: N.- 13.

Fuente: Víctor Chiza.

Esta es la pantalla inicial en la cual se puede observar la portada del sistema, para entrar al LOGIN hay que dar un clic en el botón Iniciar Sesión luego de llenar los campos correctamente.

Formularios De Búsqueda



INGRESE EL NOMBRE DE LA ESCUELA QUE DESEA BUSCAR

CODIGO	NOMBRE
ESC01	SISTEMAS

Campo de Búsqueda

Nombre: Formularios Búsqueda.

Gráfico: N.- 14.

Fuente: Víctor Chiza.

En estos formularios podemos buscar los registros que necesitamos ya sean escuelas, tutores, lectores, etc., para ello tendremos que escribir en el cuadro de texto lo que deseamos buscar y presionamos el botón buscar obteniendo los resultados.

Formularios De Ingresos De Datos

CODIGO :

NOMBRE :

CODIGO	NOMBRE
ESC01	SISTEMAS

Campo de Ingreso de Datos

Nombre: Formularios de Ingreso de Datos.

Gráfico: N.- 15.

Fuente: Víctor Chiza.

Estos formularios nos servirán para poder ingresar datos a nuestra base de datos para ello tendremos que llenar los campos solicitados que estarán marcados con un borde rojo luego de eso presionamos el botón guardar para guardar los datos.

Formularios De Eliminación De Datos

SELECCIONE EL REGISTRO QUE DESEA ELIMINAR

ELIMINAR	CODIGO	NOMBRE
<input type="button" value="X"/>	ESC01	SISTEMAS

Botón Para Eliminar Registros

Nombre: Formularios de Eliminación de Datos.

Gráfico: N.- 16.



Fuente: Víctor Chiza.

En estos formularios podremos eliminar los registros de nuestra base de datos, al momento de presionar el icono con una x.

Formularios De Modificación De Datos

CODIGO : ESC01
 NOMBRE : SISTEMAS
 Modificar

SELECCIONE EL REGISTRO QUE DESEA ACTUALIZAR

SELECCIONAR	CODIGO	NOMBRE
	ESC01	SISTEMAS

Botón para Seleccionar Registros

Nombre: Formularios de Modificación.

Gráfico: N.- 17.

Fuente: Víctor Chiza.

En estos formularios podemos modificar los registros para ello tendremos que presionar el icono en forma de flecha del registro que queremos modificar luego llenar el campo de texto con el nuevo dato para el registro y para guardarlo presionamos el botón Modificar.

Pantalla Principal

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Página principal Acerca de Escuelas Tutores

Iniciar Sesión

Menús de Opciones

Nombre:Formulario Principal.

Gráfico: N.- 18.



Fuente: Víctor Chiza.

Este formulario es la pantalla principal del Sistema donde se encuentran los menús, el link para iniciar sesión.

Formulario Para La Creación De Una Nueva Cuenta

The screenshot shows a web form titled "CREAR UNA NUEVA CUENTA". Below the title, there are instructions: "Use el formulario siguiente para crear una cuenta nueva." and "La contraseña del Usuario Será Enviada al Correo Ingresado." The form is titled "Información de cuenta" and contains the following fields:

- Nombre de usuario: A text input field with a red border.
- Seleccione El Rol que Desempeña : A dropdown menu with "ADMINISTRADOR" selected.
- Seleccione La Escuela a la que Pertenece : A dropdown menu with "SISTEMAS" selected.
- Correo electrónico: A text input field with a red border.

At the bottom right of the form is a button labeled "Crear usuario".

Callouts on the right side of the image point to the following fields:

- Nombre de Usuario (points to the "Nombre de usuario" field)
- Rol que Desempeña el Usuario (points to the "Seleccione El Rol que Desempeña" dropdown)
- Escuela a la que Pertenece el Usuario (points to the "Seleccione La Escuela a la que Pertenece" dropdown)
- Correo Electrónico del Usuario (points to the "Correo electrónico" field)

Nombre:Formulario para Crear Cuentas.

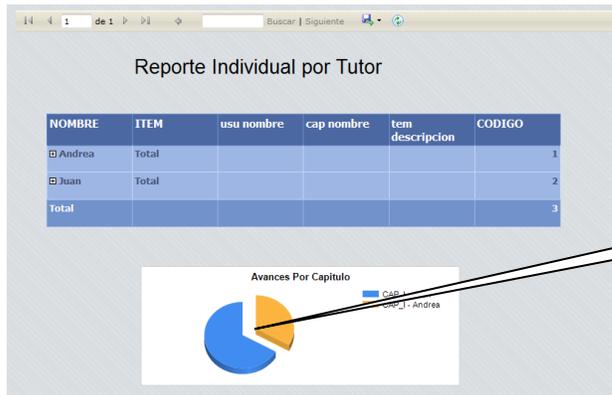
Gráfico: N.- 19.

Fuente: Víctor Chiza.

Este formulario nos sirve para la creación de una nueva cuenta.

Reportes

A callout box with a pointer pointing to the left, containing the text "Datos de los Alumnos".



Graficas Estadísticas

Nombre: Formularios de Reportes.

Gráfico: N.- 20.

Fuente: Víctor Chiza.

Estos Formularios no permiten emitir reportes ya sean globales, individuales dependiendo del rol que desempeñe.

Avances

Datos del Alumno

Datos del Proyecto

Buscar Alumno : Nombres Apellidos Cédula

Alumno :

Capitulo : CAP_II

Items : Planteamiento

Fecha Inicial : 2011/9/1 Fecha Final : 2011/9/15

SELECCIONAR	CODIGO	NOMBRE	APELLIDO	ITEM	OBSERVACION	cap_nombre	tem_descripcion
	1	Juan	Perez	Introduccion	Bien	CAP_1	Pedido Por Mensajes
	3	Juan	Perez				
	4	Andrea	Acosta	Indice	Completo	CAP_1	COMUNITARIO

Búsqueda de Tutoriado

Fechas Para Entregas

Datos de Tutoriados

Nombre: Formularios de Avances.

Gráfico: N.- 21.

Fuente: Víctor Chiza.

En este formulario el Director de Proyectos podrá ingresar los avances de sus tutoriados.

Ingreso de Alumnos con sus Respective Temas

Cedula del Alumno



CÓDIGO : 13
CÉDULA :
NOMBRES :
APELLIDOS :
TEMA :
PERFIL DE TEMA : DESARROLLO
ESCUELA : SISTEMAS
MODALIDAD : MATUTINO
ESTADO :
Guardar

Nombres del Alumno
Apellidos del Alumno
Tema del Alumno
Escuela a la que Pertenece el Alumno

Nombre: Formularios de Ingreso de Alumnos.

Gráfico: N.- 22.

Fuente: Víctor Chiza.

En este formulario el Director de Trabajo podrá ingresar los datos de sus Tutoriados con su respectivo tema, escuela a la que pertenece, y en la modalidad en la que está registrado.

Modificar Datos del Usuario

CÓDIGO : 2
ESCUELA : SISTEMAS
CONTRASEÑA : 111
CEDULA : 1234567890
NOMBRES : Carla
APELLIDOS : Perez
TELÉFONO : 234156
CORREO : cperz@gmail.com
Modificar Datos

Cedula del Usuario
Apellidos del Usuario
Teléfono del Usuario

CODIGO	ROL	ESCUELA	NOMBRES	APELLIDOS	TELÉFONO	CORREO	CLAVE	CEDULA
2	TUTOR.SISTEMAS	Carla	Perez	234156	cperz@gmail.com	111	1234567890	

Nombre: Formularios para Modificar Datos.

Gráfico: N.- 23.

Fuente: Víctor Chiza.

En este formulario el usuario podrá completar los datos de su cuenta que le hagan falta.



4.14 Pruebas Y Depuración

1) Pruebas de Integración

Esta prueba la realice individualmente verificando que todo funciona bien individualmente, por lo tanto las relaciones existentes en el modelo establecido se verificaron la consistencia del modelo indicando las claves primarias y foráneas establecidas, a fin de poder verificar la indexación de las búsquedas o consultas realizadas tanto los alumnos, tutores, lectores y directores.

La prueba de integración es una técnica sistemática para construir la arquitectura del software, mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. La integración de los datos en la tabla, determina la robustez del modelo colocando especial énfasis en las asociaciones de las tablas que tienen relación con la información que maneja los datos.

2) Pruebas de Validación

- Las pruebas de validación empiezan tras la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales. Se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz.

Este tipo de pruebas se la realizo en el módulo de seguridad establecido para poder controlar el acceso de los usuarios al sistema. También se estableció para poder verificar la integración que existe de los datos transaccionales con los datos espaciales.

- La prueba se concentra en las acciones visibles para el usuario y en la salida del sistema que éste puede reconocer.

Este tipo de prueba se realizó con el módulo de seguridad poniendo énfasis en los tipos de usuarios que tiene que manejar el sistema



- Sin embargo, los pasos dados durante el diseño y la prueba del software mejorarán en gran medida la probabilidad de tener éxito en la integración del software del sistema mayor.

La integración de los datos y las relaciones entre cada uno de las tablas se verifico la consistencia de los datos y la integridad de los mismos.

3) Pruebas del Sistema

- Al final del desarrollo el software se incorpora a otros elementos del sistema (hardware, personas, información) y se realiza una serie de pruebas de integración del sistema y de validación.
- Sin embargo, los pasos dados durante el diseño y la prueba del software mejorarán en gran medida la probabilidad de tener éxito en la integración del software del sistema mayor. La integración de los datos y las relaciones entre cada uno de las tablas se verifico la consistencia de los datos y la integridad de los mismos.

4) Prueba de seguridad

- Estas pruebas son muy importantes, realizarlas ya que nuestro sistema al funcionar directamente en la web, debemos tener en cuenta todas las seguridades, principalmente en los inicios de sesión de cada usuario, ya que cada usuario, como los lectores, tutores, directores, y los alumnos tienen roles diferentes. Es por ello tener en cuenta estas seguridades.
- Otra de las seguridades que realizamos las pruebas, es al momento de iniciar sesión con varios navegadores de internet, como son: Opera, Internet Explorer, Google Chrome, y Firefox, siendo los más utilizados. Al realizar esta prueba



pudimos observar que al iniciar sesión en un diferente navegador las validaciones funcionan correctamente.

5) Prueba de Interfaces Gráficas de Usuario (Reportes estadísticos)

Estas pruebas de interfaces gráficas, son los reportes que generaremos de los avances de cada capítulo del proyecto, que será calificado tanto por el tutor, como del lector, estos reportes estadísticos se podrán generar tanto en forma global, individual, y por escuelas. Hemos visto que estos reportes se generan sin ningún problema.

4.15 Recopilación Y Carga De Datos

Con la finalidad de poder establecer los datos adecuados se realizó las pruebas necesarias, con lo que se validó la funcionalidad del sistema especialmente los reportes generados de la base de datos y al despliegue de resultados como consecuencia de los filtros de información que tiene que realizar a fin de poder visualizar reportes estadísticos.

Posteriormente se realizara una carga de datos considerable para demostrar la robustez del prototipo y la rapidez del mismo, igualmente las consultas tanto de los tutores, lectores, directores, y alumnos se lo realiza en la carga de datos inicial, lo que significa que a su vez se pueden realizar los mantenimientos necesarios en cualquiera de estas tablas.

4.16 Pruebas Y Depuración Final En Funcionamiento

Una vez terminado nuestro sistema realizaremos las pruebas y depuración final tanto de la base de datos, como de nuestro sistema, y a su vez probar las seguridades, más importantes para que así no sea jaqueado, para ello probamos primeramente la base de datos que esté correctamente instalada en el servidor de datos del Instituto, luego instalamos nuestro sistema en el servidor web, configurándolo, y verificando si está conectado a la base de datos del servidor del Instituto. Una vez que hemos realizado toda la instalación de la base de datos como también de nuestro sistema,



procedemos a comprobar si hace falta corregir ciertos errores de compilación o configuración para así terminar con todas las pruebas.

4.17 Puesta En Marcha Del Sistema

Una vez que se instala y configura la Base de Datos My SQL, se tiene que definir el funcionamiento del ambiente del mismo que almacenara los datos de las Escuelas, Los Temas, Los alumnos, Los Tutores, Los Lectores y Directores, los propietarios del esquema de la base son el Administrador los cuales tenemos que configurar y desbloquear, posteriormente se tendrá que realizar las interfaces de comunicación entre la base de datos y a su vez, el funcionamiento en la WEB, para probar la comunicación de nuestro sistema a la base de datos del Instituto.

4.18 Capacitación Al Usuario Final

El usuario recibirá la capacitación en el departamento tecnológico del Instituto, iniciando el mismo con la explicación de las tecnologías utilizadas para el aplicativo, como se tiene que realizar las cargas de datos, y como utiliza la información de la base de datos MySQL, igualmente como desplegar los reportes Estadísticos en la interface grafica, también se realizara la explicación del funcionamiento de las herramientas del manejo del sistema, como sacar información de la base de datos, como colocar y a su vez como se realizaran las calificaciones de cada ítem de los capítulos en el caso de los tutores y en la calificación que también la realiza el lector .

La metodología aplicada en la capacitación al usuario final será demostrativa, iniciando la misma con la interface que va a ocupar el usuario en sus consultas diarias, realizando una explicación detallada de cada uno de los elementos que interviene en el sistema.

DIA 1	DIA 2
--------------	--------------



Capacitación de Sistema a los Usuarios	Capacitación de la funcionalidad del Sistema WEB
Directores de Trabajo, Directivos, Lectores	A los Usuarios

Tabla No19 Capacitación de Usuario Final

Metodología

Al tener implementado el Sistema de Control de Proyectos en el Servidor, el siguiente paso será la capacitación a los usuarios que va a utilizar el sistema, que vendrían a ser: Directores de Trabajo, Lectores, Directivos y Directores.

La capacitación se realizara mediante una reunión en conjunto mediante un proyector para hacerles una demostración del funcionamiento de Sistema de Control de Proyectos.

También se tratara de capacitar en forma individual a cada persona en caso de no entenderlo realizando pruebas de ingresos de datos, verificación y autenticidad de la información.

Lugar de la capacitación: La capacitación se dará en el Instituto Tecnológico Superior Cordillera.

Tiempo de la capacitación: El tiempo será un total de 2 horas.

Temario de la capacitación: En el temario incluiremos todos los temas que proporcionaremos en toda la capacitación. Que en este caso los temas principales serán: instalación de la base de datos, configuración del IIS, implementación del sistema.



Material de entrega para la capacitación: El material que se les entregara, es un manual de usuario, detallado de la mejor manera posible para que puedan utilizar de manera correcta nuestro sistema.

Evaluación de la capacitación: Se evaluara con un banco de preguntas referentes a la capacitación, para constatar que toda la capacitación, quedo claro, para todos los usuarios de nuestro sistema.

Cronograma de desarrollo de la capacitación: El cronograma con el que se realizara la capacitación es el siguiente.

4.19 Capacitación Al Personal Técnico

Para la capacitación al personal técnico se tomara en cuenta las herramientas que se están utilizando, en nuestro caso My SQL para la base de datos, el administrador de la base de datos será quien realice los respectivos mantenimientos de la base, para ello es muy importante que tengan el conocimiento de My SQL, ya que el administrador es el encargado de ingresar, modificar, eliminar los daros de la base de datos.

El personal técnico también estará capacitado en la carga de los datos como también en todo el funcionamiento del sistema.

Al estar el Sistema de Control de Proyectos implementado en el servidor, el siguiente paso será proceder a la capacitación del personal técnico del sistema en este caso sería el Administrador del Sistema.



La capacitación se realizara mediante una reunión mediante un proyector para hacer una demostración del funcionamiento de Sistema de Control de Proyectos, de cómo deben ingresar los datos, como crear cuentas, como modificar datos etc.

DIA1	DIA 2	DIA 3	DIA 4
Instalación de la base de Datos	Configuración del servidor Web	Capacitación de Sistema al Administrador	Funcionabilidad del Sistema WEB
Configuración de la base de Datos	Manejo de la base de Datos	Súper Administrador	Mantenimientos

Tabla No20 Capacitación de Personal Técnico

Lugar de la capacitación: La capacitación se dará en el Instituto Tecnológico Superior Cordillera.

Tiempo de la capacitación: El tiempo será un total de 4 horas, que se dará una hora por día, siendo un total de 4 días de capacitación.

Temario de la capacitación: En el temario incluiremos todos los temas que proporcionaremos en toda la capacitación. Que en este caso los temas principales serán: instalación de la base de datos, configuración del IIS, implementación del sistema.

Material de entrega para la capacitación: El material que se les entregara, es un manual de usuario, detallado de la mejor manera posible para que puedan utilizar de manera correcta nuestro sistema.



Evaluación de la capacitación: Se evaluara con un banco de preguntas referentes a la capacitación, para constatar que toda la capacitación, quedo claro, para todos los usuarios de nuestro sistema.

Cronograma de desarrollo de la capacitación: El cronograma con el que se realizara la capacitación es el siguiente.

CAPÍTULO V

PRINCIPALES IMPACTOS

5.1 Científico.

La investigación desarrollada por el proyecto tiene una característica especial ya que aplicativos y tecnologías implementadas en nuestro medio únicamente se lo realizaba anteriormente con un software en dos capas el cual el soporte técnico necesario, es mucho más difícil, por lo tanto al tratarse de una implementación web se podrá beneficiar los técnicos y los demás usuarios, que utilizaran nuestro sistema, ya que cualquier inquietud y soporte se lo podrá arreglar sin para las reglas del negocio.

5.2 Educativo.

El desarrollo de un sistema informático involucra el conocimiento del diseño y la programación, por lo tanto al realizar un proyecto de fusión de tecnologías se realizara un aporte muy importante para el conocimiento de los alumnos de la institución que se encuentran en los niveles inferiores los cuales se pueden motivar y desarrollar proyectos de fusión de muchas tecnologías existentes.



5.3 Técnico.

Lo referente a este tema se considera las herramientas que se utilizaron para el modelamiento y desarrollo del Sistema de Control de Tesis, considerando que las mismas fueron estudiadas en las aulas de la Institución, en consecuencia su aplicación no tuvo mayor dificultad.

5.4 Tecnológico.

El ejecutar temas de innovación tecnológica aplicando los conocimientos recibidos en las aulas hace que cualquier proyecto sea considerado como un aporte institucional hacia la sociedad, mejor aun si lo que se resta realizando es la unión de varias tecnologías en un solo aplicativo, que es producto de la investigación tecnológica.

5.5 Empresarial.

Siendo la visión emprendedora la realización en un futuro de una empresa, se ha cristalizado el primer paso que consiste en tener un aplicativo que maneje las reglas del negocio, por lo tanto a medida que se perfecciona el mismo se continuara con una mentalidad empresarial.

5.6 Social.

La implementación de este sistema y la estructuración de una empresa en lo futuro para brindar este tipo de servicio, lograra crear fuentes de trabajo tanto en el ámbito administrativo como técnico, por consiguiente se estará brindando un aporte al ámbito social.



5.7 Económico.

La generación de impuestos por el servicio brindado con este tipo de empresa beneficiara a la economía nacional ya que su aporte será con todos lo que establece la ley en pagos de aranceles e impuestos en general.

5.8 Conclusiones.

1. La metodología de investigación científica aplicada al proyecto permitirá determinar la utilización de tipos de investigación, métodos y herramientas que facilitaron la obtención de resultados cuantitativos y cualitativos, que proporcionaron la obtención de procedimientos, tareas y procesos propios de la toma de decisiones del Sistema de Control de Tesis.
2. La aplicabilidad de los pasos que se siguen en ingeniería de software fueron determinantes ya que mediante el análisis permitieron discernir todo el flujo de información que se ejecuta en el proceso de Controlar los proyectos y las tesis de los alumnos, al mismo tiempo poder diseñar adecuadamente los procesos sus relaciones, las bases de datos y sus objetos, a fin de poder obtener una organización metódica y bien estructurada de nuestro sistema.
3. Los conocimientos adquiridos en el Instituto tecnológico Superior Cordillera, me permitieron consolidar las teorías pragmáticas en herramientas de ejecución prácticas utilizando lenguajes de programación, y bases de datos materializadas en conocimientos de programación informática. Lo que definió en una concatenación de ideas lógicamente estructuradas y enlazadas hacia la consecución de un objetivo general "Analizar, desarrollar e implementar un Sistema de Control de Tesis en tres capas.



4. En un entorno de desarrollo de software se debe realizar muchas pruebas de campo como validaciones de sistema, validaciones de usuarios, para poder decir que el software está terminado y en funcionamiento, más aun si consideramos que en nuestro caso también se tiene que mantener un especial atención por la fusión de tecnologías ejecutadas en el proyecto, lo que permitirá la consolidación de la información estructural y organizada para la utilización de cada usuario según su rol que desempeña.

5. Hay que mencionar el hecho que para el desarrollo de este tipo de tecnologías es indispensable utilizar lenguajes de tercera generación y orientada a objetos, para poder lograr un alto rendimiento en el manejo de este tipo de ambientes tecnológicos.

6. Con este tipo de sistemas integrados me pude dar cuenta que como alumno de la carrera de sistemas no solo debo instruirme en mi especialidad sino también interactuar con otras, como por ejemplo el manejo y conocimiento de este lenguaje de programación Web Asp .Net que es tema fundamental dentro de mi desarrollo, ya que por medio de este conocimiento logramos realizar nuestro sistema en la Web.

7. Mediante la configuración de la base de datos en un servidor de datos fue muy importante saber cómo realizar la tanto la configuración como la implementación, ya que fue nuestro primer paso para realizar las respectivas pruebas en el sistema y de esta manera validar el ingreso de sesiones por usuario.

8. Otra configuración importante es la del Internet InformationService (IIS), que es muy importante para levantar en un sitio Web nuestro sistema mediante una IP



pública, razón por la cual fue muy importante esta configuración en el servidor Web.

5.9 Recomendaciones.

1. Todo trabajo de investigación debe sustentarse con una metodología métodos y herramientas de la investigación científica por lo tanto es recomendable que las personas se preparen en este campo para realizar cualquier investigación, mas aun considerando que esto es la base para poder realizar el proyecto de grado previo a la obtención del título de Tecnólogo Analista de Sistemas.
2. Seguir la misma secuencia de desarrollo de software a fin de poder llegar a obtener un sistema completo, aplicable y amigable, es el resultado de haber seguido una metodología de desarrollo y técnicas de programación que permita interactuar los datos alfanuméricos (información), con los datos y relacionarlos, a fin de poder estructurar un sistema real de Toma de decisiones.
3. Para cualquier tipo de integración de tecnologías avanzadas es indispensable conocer los diversos lenguajes de programación por lo cual es recomendable instruir a los alumnos en la nuevas herramientas de desarrollo de software que cada día se van actualizando y que nos ofrecen mayores facilidades para integrar sistemas de muy alto rendimiento.
4. Siempre será importante conocer que en el mundo del desarrollo de software existen diversos tipos de caminos o integraciones con otros sistemas y especialidades, por lo que se debería agregar como materia principal el manejo



de diversas tecnologías, ya que en el mundo laboral el tecnólogo debe estar listo para programar o resolver cualquier tipo de problema.

5. Como otra recomendación muy importante podemos destacar que dentro de las seguridades de la zona desmilitarizada, el Instituto solamente cuenta con un firewall, que lo convierte en una red muy vulnerable al hackeo.
6. La zona desmilitarizada tiene como finalidad aislar, físicamente los servicios internos, es decir separándolos de los servicios públicos.
7. Es necesario implementar en la zona desmilitarizada un IDS; ya que el IDS puede ser descrito como un detector que procesa la información proveniente del sistema monitoreado. Es una herramienta de apoyo en procesos de auditoría, entendida como el control del funcionamiento de un sistema a través del análisis de su comportamiento interno.
8. También implementar un IPS, que cumplen con la función de ser dispositivos de hardware o software encargados de revisar el tráfico de red con el propósito de detectar y responder a posibles ataques o intrusiones.

La respuesta usualmente consiste en descartar los paquetes involucrados en el ataque o modificarlos (*scrubbing*) de tal manera que se anule su propósito. Es claro que este comportamiento los clasifica como dispositivos proactivos debido a su reacción automática a situaciones anómalas.

9. Otro de los servidores fundamentales que se recomienda es también implementar un servidor DNS que son muy fundamentales, ya que utiliza una



base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

10. Es muy fundamental el implementar también un servidor de correo para poder controlar los correos de todos los usuarios principalmente de los que van a utilizar nuestro sistema.
11. Como última recomendación que les damos es que utilicen el Navegador Firefox (Cualquier versión), Ya que este navegador respeta todos los estándares WEB



CAPITULO VI

Bibliografía

1. Burton Harvey, Simon Robinson, Julian Templeman y Karli Watson , (2000), "C# Programming with the Public Beta", Wrox Press
2. Simon Robinson, Burt Harvey, Craig McQueen, Christian Nagel, Morgan Skinner, Jay Glynn, Karli Watson, Ollie Cornes, Jerod Moemeka,(2001),"Professional C#", Wrox Press .



3. JesseLiberty,(2001),“Programming C#”, O’Reilly
4. Eric Gunnerson(200),“A programmer’s introduction to C#”, Apress.
5. Jorge Serrano (2009) Manual Avanzado de Visual Basic 2008, editorial Anaya Multimedia, edición 2.
6. Jorge Serrano (2010) Manual Avanzado de Visual Basic 2010, editorial Anaya Multimedia, edición 1.

Net Grafía

1. Guía de referencia del lenguaje C#:

(<http://www.msdn.microsoft.com/net/ecma>)

2. Compilador de C# (incluido en el .NET SDK):

(<http://www.msdn.microsoft.com/code/sample.asp?url=/msdn-files/027/000/976/msdncompositedoc.xml>)

3. Versión beta de Visual Studio .NET:

(<http://msdn.microsoft.com/vstudio/nextgen/beta.asp>)

4. Portal de C# en español:

(<http://tdg.lsi.us.es/~csharp>)

5. Web de Microsoft:

www.msdn.microsoft.com



GLOSARIO DE TÉRMINOS TÉCNICOS

ANSI: American National Standards Institute - Instituto Nacional de Normas de Estados Unidos.

Aplicación: Cualquier programa que corra en un sistema operativo y que haga una función específica para un usuario. Por ejemplo, procesadores de palabras, bases de datos, agendas electrónicas, etc.

DDL: Lenguaje de definición de datos (Sistemas gestores de bases de datos).

GNU: Las letras GNU se refieren en inglés a No es UNIX (Not UNIX); es un sistema de programas compatible con UNIX, desarrollado por el Free Software Foundation (FSF). El proyecto GNU empezó en 1983 por Richard Stallman del Massachusetts Institute of Technology. La filosofía detrás de GNU es producir programas que no sean propietarios. Cualquiera puede descargar, modificar y redistribuir programas GNU. La única restricción es que no le pueden limitar la redistribución. Linux utiliza muchos programas GNU, y en el pasado los sistemas GNU usaban el kernel de Linux. Esta estrecha relación entre Linux y GNU hace que muchas personas piensen, erradamente, que son lo mismo. Pero son bastante distintos, y GNU está desarrollando su propio kernel llamado HURD para reemplazar el kernel de Linux en los sistemas GNU.

GPL: Acrónimo en inglés de General Public License (Licencia Pública General). Esta licencia regula los derechos de autor de los programas de software libre (free software) promovido por el Free Software Foundation (FSF) en el marco de la iniciativa GNU. Permite la



distribución de copias de programas (e incluso cobrar por ello), así como modificar el código fuente de los mismos o utilizarlo en otros programas.

IIS: Microsoft Internet Information Services. Servicios de Información de Internet de Microsoft. IIS es un conjunto de servicios basados en Internet, para maquinas con Windows. Compite con Apache en el área de servidores web.

INTERFAZ: Una conexión e interacción entre hardware, software y usuario, es decir como la plataforma o medio de comunicación entre usuario o programa.

MySQL: My SQL es uno de los Sistemas Gestores de Bases de Datos más populares. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características altamente funcionales, ha dado lugar a un sistema de administración de base de datos de alta velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido. Lo mantiene la empresa sueca MySQL AB bajo la licencia GPL (GNU PublicLicense); puede utilizarse gratuitamente y su código fuente está disponible.

USUARIO: Cualquier individuo que interactúa con la computadora a nivel de aplicación. Los programadores, operadores y otro personal técnico no son considerados usuarios cuando trabajan con la computadora a nivel profesional.

ODBC: Open DatabaseConnectivity. Estándar de acceso a Bases de Datos desarrollado por Microsoft cuyo objetivo es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene



los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS.

Perl: PracticalExtraction and ReportLanguage -- Lenguaje de programación muy utilizado para la elaboración de aplicaciones CGI, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc, o a herramientas locales como WAIS. Perl es un lenguaje para manipular textos, archivos y procesos, proporciona una forma fácil y legible para realizar trabajos que normalmente se realizarían en C o en un shell. Perl nació y se ha difundido bajo el sistema operativo UNIX, aunque existe para otras plataformas. Perl fue desarrollado por Larry Wall, y está distribuido libremente bajo la filosofía GNU.

Página Web: Resultado en hipertexto o hipermedia que proporciona un navegador del WWW después de obtener la información solicitada. Su contenido puede ir desde un texto corto a un voluminoso conjunto de textos, gráficos estáticos o en movimiento, sonido, etc. Algunas veces el citado término es utilizado incorrectamente en orden de designar el contenido global de un sitio web, cuando en ese caso debería decirse "Web site".

Windows CommunicationFoundation o WCF (también conocido como **Indigo**): Es la nueva plataforma de mensajería que forma parte de la API de la Plataforma .NET 3.0 (antes conocida como WinFX, y que no son más que extensiones para la version 2.0). Se encuentra basada en la Plataforma .NET 2.0 y de forma predeterminada se incluye en el Sistema Operativo Microsoft Windows Vista.

Windows PresentationFoundation (WPF): Es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando las mejores características de las aplicaciones Windows y de las aplicaciones web.



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

ANEXOS

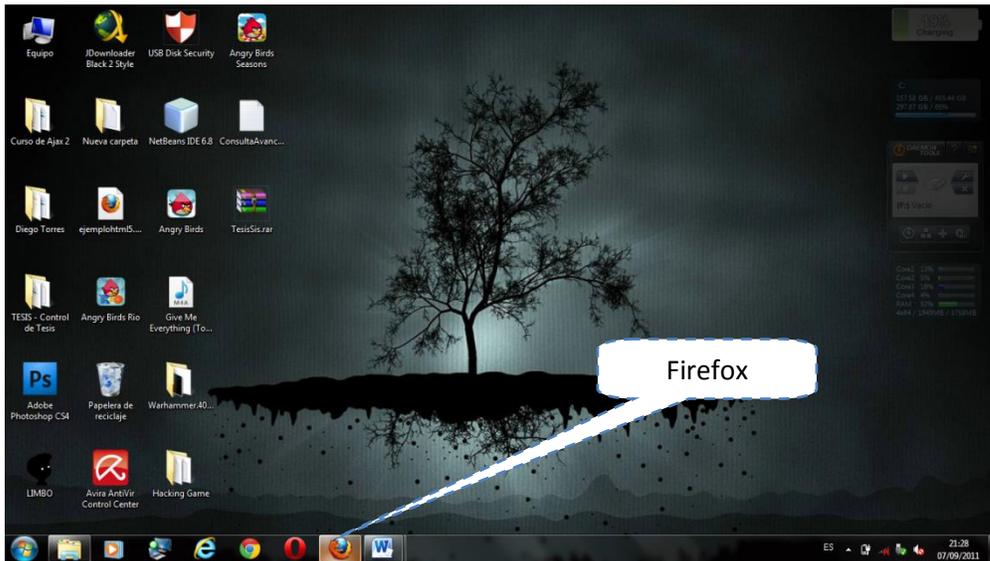
MANUAL DE USUARIO

INGRESO AL SISTEMA

Para ingresar el sistema se da doble click en el escritorio en el icono que representa a los navegadores en este caso Firefox



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

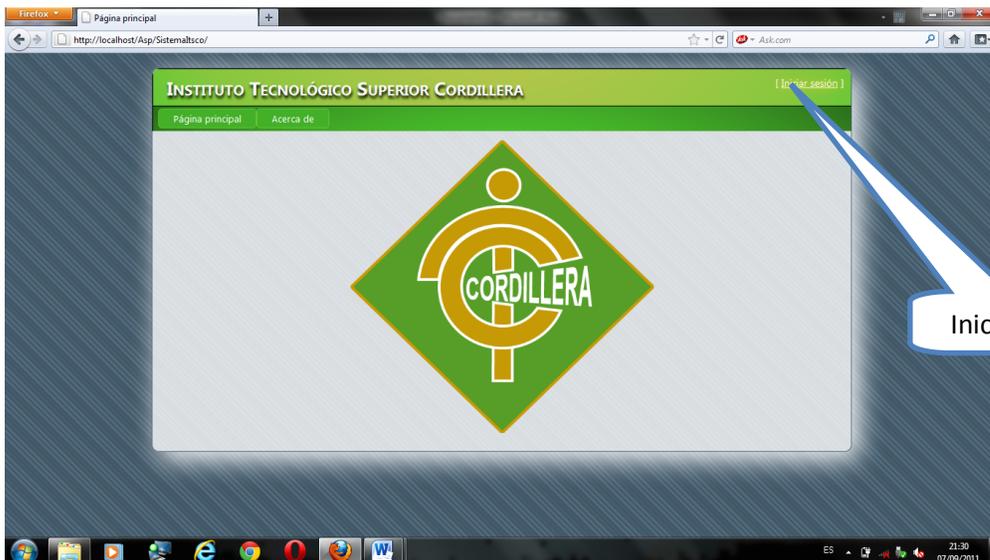


Nombre: Ingreso al Sistema.

Gráfico: N.- 24.

Fuente: Víctor Chiza.

Al ingresar al sistema se presenta la siguiente pantalla:



Nombre: Pantalla Principal.

Gráfico: N.- 25.



Fuente: Víctor Chiza.

En esta pantalla hacemos click en el link **Iniciar Sesión** y a continuación nos mostrara la siguiente pantalla.

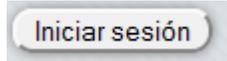
Esta pantalla es la de inicio del sistema, aquí nos pedirá ingresar el nombre de usuario y la contraseña, si son correctos presenta el menú del sistema, caso contrario no nos permitirá ingresar.

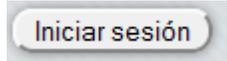
Nombre: Pantalla de Inicio de Sesión.

Gráfico: N.- 26.

Fuente: Víctor Chiza.

Es importante ingresar la contraseña tal como se escribió la primera vez, es decir, si contiene características en minúsculas, tiene que ser en minúsculas y viceversa, es recomendable usar una contraseña fácil de recordar y además solo en minúsculas.



Una vez ingresado los datos en sus respectivos campos presionamos  si el password es incorrecto se muestra el siguiente mensaje

El intento de conexión no fue correcto. Inténtelo de nuevo.

Información de cuenta

Nombre de usuario:

Contraseña:

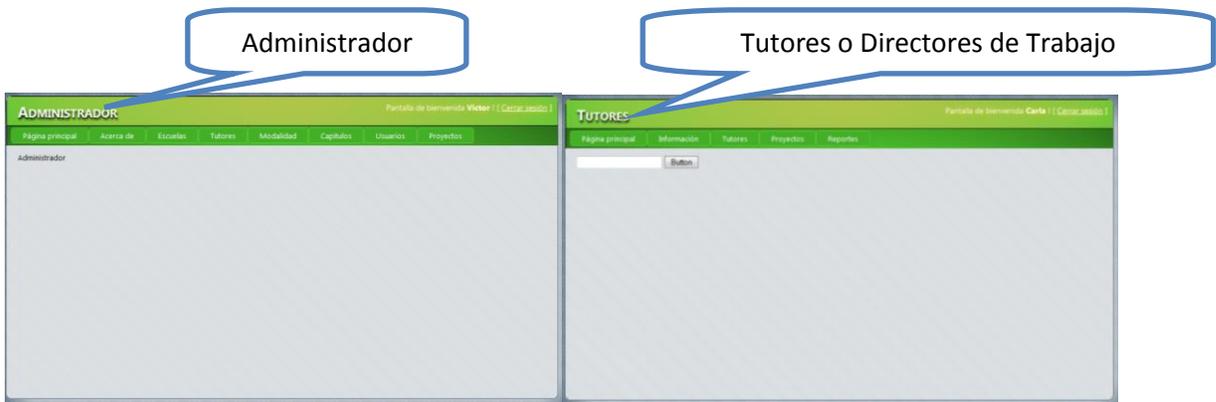
Mantenerme conectado

Nombre: Mensajes.

Gráfico: N.- 27.

Fuente: Víctor Chiza.

Una vez dentro, es sistemas muestra la pantalla principal del sistema dependiendo del rol que cumpla el usuario nos presentara los siguientes formularios.



Nombre: Formularios de Usuarios.

Gráfico: N.- 28.

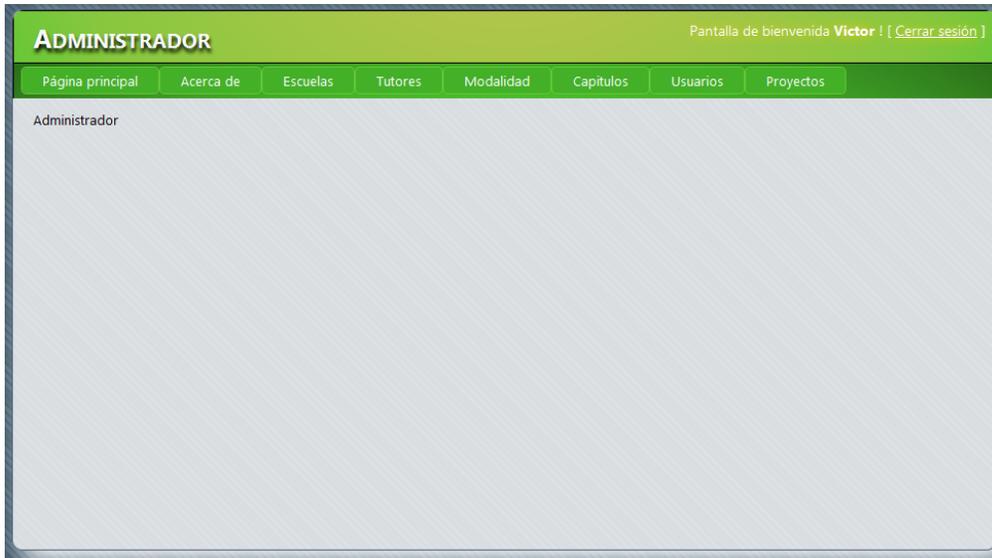
Fuente: Víctor Chiza.

Y así con los demás roles.

Pantalla principal



Está pantalla muestra un menú con características propias para el Administrador.



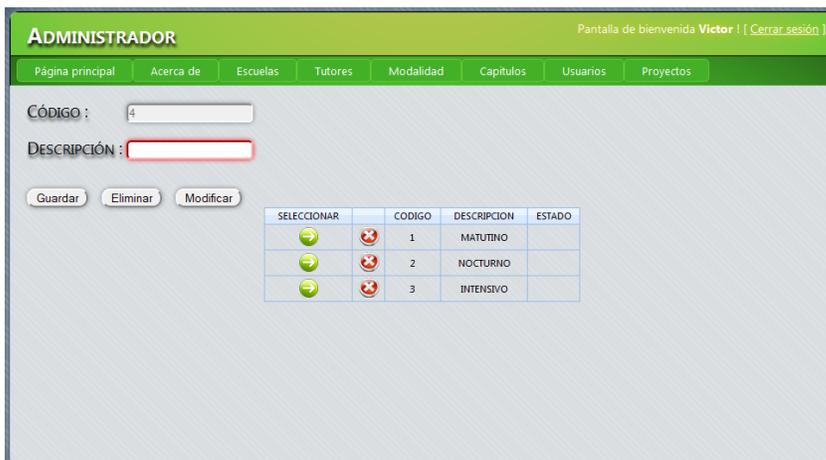
Nombre: Formulario de Administrador.

Gráfico: N.- 29.

Fuente: Víctor Chiza.

Pantalla Mantenimiento:

En estos tipos de formularios podremos ingresar información dependiendo del formulario en este caso podremos ingresar, modificar, eliminar modalidades.





Nombre: Formularios de Mantenimientos.

Gráfico: N.- 30.

Fuente: Víctor Chiza.

Pantalla Seguridades:

Esta pantalla podremos crear los diferentes tipos de usuarios que utilizaran el sistema.

ADMINISTRADOR Pantalla de bienvenida **Victor** | [Cerrar sesión]

Página principal | Acerca de | Escuelas | Tutores | Modalidad | Capítulos | Usuarios | Proyectos

CREAR UNA NUEVA CUENTA

Use el formulario siguiente para crear una cuenta nueva.
La contraseña del Usuario Será Enviada al Correo Ingresado.

Información de cuenta

Nombre de usuario:

Seleccione El Rol que Desempeña:

Seleccione La Escuela a la que Pertenece:

Correo electrónico:

Nombre: Formularios para Crear Cuentas.

Gráfico: N.- 31.

Fuente: Víctor Chiza.



Pantalla Procesos:

En esta pantalla el Director de Trabajo debe buscar al tutoriado que vaya a presentar el avance respectivo al capítulo de la fecha impuesta por el Administrador para ello podrá hacerlo por nombres, apellidos, cedula.

Una vez encontrado el alumno procederá a escoger los ítems del capítulo habilitado para dicha fecha y podrá calificarlo de forma individual por ítems.

The screenshot shows the 'TUTORES' web application interface. At the top, there is a navigation bar with 'Página principal', 'Información', 'Tutores', 'Proyectos', and 'Reportes'. Below this, there are two main sections: 'Datos del Alumno' and 'Datos del Proyecto'. The 'Datos del Alumno' section includes a search box for 'Buscar Alumno' with checkboxes for 'Nombres', 'Apellidos', and 'Cédula', and two input fields for 'Alumno'. The 'Datos del Proyecto' section includes a search box for 'Buscar Alumno', a dropdown for 'Capítulo' (set to 'CAP_II'), a dropdown for 'Ítems' (set to 'Planteamiento'), and date fields for 'Fecha Inicial' (2011/9/1) and 'Fecha Final' (2011/9/15). Below these sections is a table with columns for 'Presentados', 'No Presentados', and 'Inactivos'. The table has 8 columns: 'SELECCIONAR', 'CODIGO', 'NOMBRE', 'APELLIDO', 'ITEM', 'OBSERVACION', 'cap_nombre', and 'tem_descripcion'. The table contains three rows of data, each with a green arrow icon in the 'SELECCIONAR' column.

SELECCIONAR	CODIGO	NOMBRE	APELLIDO	ITEM	OBSERVACION	cap_nombre	tem_descripcion
→	1	Juan	Perez	Introduccion	Bien	CAP_I	Pedido Por Mensajes
→	3	Juan	Perez	Indice	Bien	CAP_I	Pedido Por Mensajes
→	4	Andrea	Acosta	Indice	Completo	CAP_I	COMUNITARIO

Nombre: Formulario de Procesos.

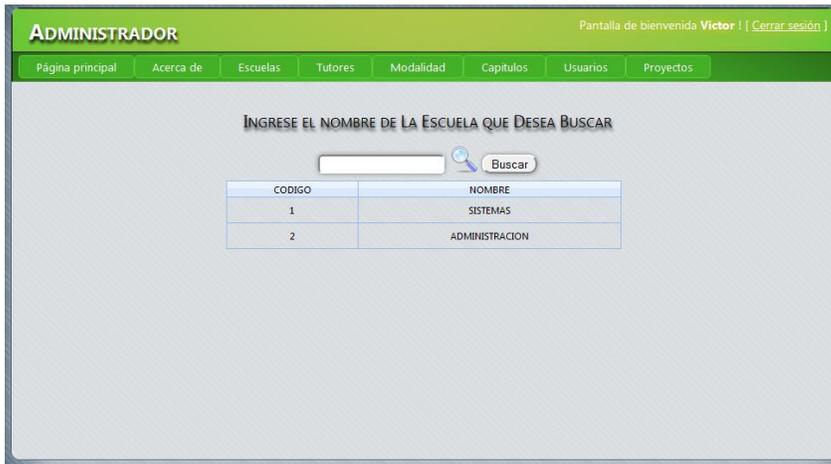
Gráfico: N.- 32.

Fuente: Víctor Chiza.

Pantalla Consultas:

Estas pantallas podremos hacer búsquedas dependiendo del rol por ejemplo el Administrador podrá buscar escuelas, alumno, tutores, lectores, avances.

En caso del Tutor o Director de Trabajo solo podrá buscar alumnos que dicho Director este tutorando, también podrá buscar avances.



Nombre: Formularios de Búsqueda.

Gráfico: N.- 33.

Fuente: Víctor Chiza.



MANUAL DE TÉCNICO

SCRIPT BASE DE DATOS

```
-- phpMyAdmin SQL Dump
-- version 3.2.0.1
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 12-09-2011 a las 03:27:29
-- Versión del servidor: 5.1.36
-- Versión de PHP: 5.3.0

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

---- Base de datos: `sistemaitscoprueba`--

-----

---- Estructura de tabla para la tabla `alumnos`--

CREATE TABLE IF NOT EXISTS `alumnos` (
  `ALU_CODIGO` int(11) NOT NULL AUTO_INCREMENT,
  `ALU_CEDULA` varchar(10) DEFAULT NULL,
  `ESC_CODIGO` int(11) DEFAULT NULL,
  `MOD_CODIGO` int(11) DEFAULT NULL,
  `ALU_NOMBRE` varchar(50) NOT NULL,
  `ALU_APELLIDO` varchar(50) NOT NULL,
  `ALU_ESTADO` varchar(1) NOT NULL COMMENT 'A="ACTIVO", I="Inactivo", B="Borrado"',
  PRIMARY KEY (`ALU_CODIGO`),
  KEY `FK_RELATIONSHIP_13` (`MOD_CODIGO`),
  KEY `FK_RELATIONSHIP_2` (`ESC_CODIGO`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;

-----

---- Estructura de tabla para la tabla `avances`--

CREATE TABLE IF NOT EXISTS `avances` (
  `AVA_CODIGO` int(11) NOT NULL AUTO_INCREMENT,
  `ITE_CODIGO` int(11) DEFAULT NULL,
  `FUN_CODIGO` int(11) DEFAULT NULL,
  `CAP_CODIGO` int(11) DEFAULT NULL,
  `TEM_CODIGO` int(11) DEFAULT NULL,
  `DT_CODIGO` int(11) DEFAULT NULL,
  `AVA_PORCENTAJE` int(11) NOT NULL,
  `AVA_OBSERVACION` varchar(100) DEFAULT NULL,
  `AVA_CALIFICACION` int(11) DEFAULT NULL,
  `AVA_ESTADO` varchar(2) NOT NULL,
  PRIMARY KEY (`AVA_CODIGO`),
  KEY `FK_RELATIONSHIP_18` (`ITE_CODIGO`),
  KEY `FK_RELATIONSHIP_19` (`TEM_CODIGO`),
  KEY `FK_RELATIONSHIP_23` (`FUN_CODIGO`),
  KEY `FK_RELATIONSHIP_24` (`CAP_CODIGO`),
  KEY `FK_RELATIONSHIP_28` (`DT_CODIGO`)
```



```
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----  
---- Estructura de tabla para la tabla `capitulos`--
```

```
CREATE TABLE IF NOT EXISTS `capitulos` (  
  `CAP_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `CAP_NOMBRE` varchar(50) NOT NULL,  
  `CAP_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`CAP_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
-----  
---- Estructura de tabla para la tabla `dtpar_periodos`--
```

```
CREATE TABLE IF NOT EXISTS `dtpar_periodos` (  
  `DT_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `CAP_CODIGO` int(11) DEFAULT NULL,  
  `PAR_CODIGO` int(11) DEFAULT NULL,  
  `DT_FECHA_INICIO` date DEFAULT NULL,  
  `DT_FECHA_FINAL` date DEFAULT NULL,  
  `DT_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`DT_CODIGO`),  
  KEY `FK_RELATIONSHIP_27` (`PAR_CODIGO`),  
  KEY `FK_RELATIONSHIP_29` (`CAP_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
-----  
---- Estructura de tabla para la tabla `escuelas`--
```

```
CREATE TABLE IF NOT EXISTS `escuelas` (  
  `ESC_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `ESC_NOMBRE` varchar(50) NOT NULL,  
  `ESC_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`ESC_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
-----  
---- Estructura de tabla para la tabla `funciones`--
```

```
CREATE TABLE IF NOT EXISTS `funciones` (  
  `FUN_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `USU_CODIGO` int(11) DEFAULT NULL,  
  `ROL_CODIGO` int(11) DEFAULT NULL,  
  `ESC_CODIGO` int(11) DEFAULT NULL,  
  `FUN_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`FUN_CODIGO`),  
  KEY `FK_RELATIONSHIP_14` (`ROL_CODIGO`),  
  KEY `FK_RELATIONSHIP_22` (`USU_CODIGO`),  
  KEY `FK_RELATIONSHIP_30` (`ESC_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```



---- Estructura de tabla para la tabla `items`--

```
CREATE TABLE IF NOT EXISTS `items` (  
  `ITE_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `CAP_CODIGO` int(11) DEFAULT NULL,  
  `ITE_NOMBRE` varchar(100) NOT NULL,  
  `ITE_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`ITE_CODIGO`),  
  KEY `FK_RELATIONSHIP_16` (`CAP_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

---- Estructura de tabla para la tabla `modalidad`--

```
CREATE TABLE IF NOT EXISTS `modalidad` (  
  `MOD_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `MOD_DESCRIPCION` varchar(50) NOT NULL,  
  `MOD_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`MOD_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

--

-- Estructura de tabla para la tabla `parametros`--

```
CREATE TABLE IF NOT EXISTS `parametros` (  
  `PAR_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `PAR_DESCRIPCION` varchar(50) NOT NULL,  
  `PAR_FECHA_INICIO` date NOT NULL,  
  `PAR_FECHA_FINAL` date NOT NULL,  
  `PAR_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`PAR_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

---- Estructura de tabla para la tabla `perfiles`--

```
CREATE TABLE IF NOT EXISTS `perfiles` (  
  `PER_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `PER_DESCRIPCION` varchar(50) NOT NULL,  
  `PER_ESTADO` varchar(1) NOT NULL,  
  PRIMARY KEY (`PER_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

---- Estructura de tabla para la tabla `perfilusuario`--

```
CREATE TABLE IF NOT EXISTS `perfilusuario` (  
  `P_U_CODIGO` int(11) NOT NULL,
```



```
`P_U_ESTADO` varchar(10) NOT NULL,  
`PER_CODIGO` int(11) DEFAULT NULL,  
`USU_CODIGO` int(11) DEFAULT NULL,  
PRIMARY KEY (`P_U_CODIGO`),  
KEY `FK_RELATIONSHIP_20` (`PER_CODIGO`),  
KEY `FK_RELATIONSHIP_21` (`USU_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

---- Estructura de tabla para la tabla `roles`--

```
CREATE TABLE IF NOT EXISTS `roles` (  
  `ROL_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `ROL_DESCRIPCION` varchar(50) NOT NULL,  
  `ROL_ESTADO` varchar(10) NOT NULL,  
  PRIMARY KEY (`ROL_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

---- Estructura de tabla para la tabla `temas`--

```
CREATE TABLE IF NOT EXISTS `temas` (  
  `TEM_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `PER_CODIGO` int(11) DEFAULT NULL,  
  `FUN_CODIGO` int(11) DEFAULT NULL,  
  `ALU_CODIGO` int(11) DEFAULT NULL,  
  `TEM_DESCRIPCION` varchar(500) NOT NULL,  
  `TEM_CALIFICACION` varchar(2) NOT NULL,  
  `TEM_ESTADO` varchar(10) NOT NULL,  
  PRIMARY KEY (`TEM_CODIGO`),  
  KEY `FK_RELATIONSHIP_15` (`ALU_CODIGO`),  
  KEY `FK_RELATIONSHIP_17` (`FUN_CODIGO`),  
  KEY `FK_RELATIONSHIP_4` (`PER_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

---- Estructura de tabla para la tabla `tematutor`--

```
CREATE TABLE IF NOT EXISTS `tematutor` (  
  `T_TU_CODIGO` int(11) NOT NULL,  
  `TEM_CODIGO` int(11) DEFAULT NULL,  
  `P_U_CODIGO` int(11) DEFAULT NULL,  
  `T_TU_ESTADO` varchar(10) NOT NULL,  
  PRIMARY KEY (`T_TU_CODIGO`),  
  KEY `FK_RELATIONSHIP_25` (`TEM_CODIGO`),  
  KEY `FK_RELATIONSHIP_26` (`P_U_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

---- Estructura de tabla para la tabla `usuarios`--



```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `USU_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `USU_CEDULA` varchar(10) NOT NULL,  
  `USU_CLAVE` varchar(50) NOT NULL,  
  `USU_NOMBRE` varchar(50) NOT NULL,  
  `USU_APELLIDO` varchar(50) NOT NULL,  
  `USU_TELEFONO` varchar(10) NOT NULL,  
  `USU_CORREO` varchar(50) NOT NULL,  
  `USU_ESTADO` varchar(10) NOT NULL,  
  PRIMARY KEY (`USU_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

CÓDIGO FUENTE

Login.aspx.cs

```
using System;  
using System.Web.UI.WebControls;  
using Logica_de_Negocios;  
  
publicpartialclassAccount_Login : System.Web.UI.Page  
{  
  protectedvoid Page_Load(object sender, EventArgs e)  
  {  
    //RegisterHyperLink.NavigateUrl = "Register.aspx?ReturnUrl=" +  
    HttpUtility.UrlEncode(Request.QueryString["ReturnUrl"]);  
  }  
  
  publicstring rol { get; set; }  
  
  protectedvoid LoginUser_Authenticate(object sender, AuthenticateEventArgs e)  
  {  
    //System.Web.Security.FormsAuthentication.SignOut();  
  
    var usu = Logica_Usuarios.ObtenerUsuarios(LoginUser.UserName, LoginUser.Password);  
    GridView gr = newGridView();  
    gr.DataSource = usu;  
    gr.DataBind();  
  
    if (usu.Rows.Count > 0)  
    {  
      switch (gr.Rows[0].Cells[1].Text)  
      {  
        case"ADMINISTRADOR": //Administrador  
          Session.Add("Administrador", LoginUser.UserName);  
          e.Authenticated = true;  
          Session.Add("Administrador", LoginUser.UserName);  
          this.LoginUser.DestinationPageUrl = "~/Formularios/Administrador/InicioAdministrador.aspx";  
          break;  
  
        case"TUTOR": //Usuarios  
  
          Session.Add("Tutores", gr.Rows[0].Cells[4].Text);  
          e.Authenticated = true;  
          this.LoginUser.DestinationPageUrl = "~/Formularios/Tutores/CuentaTutor/InicioTutor.aspx";
```



```
break;
default:
break;
}
}
}
}
```

Login.aspx

```
<%@PageTitle="Iniciar
sesión" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="Login.aspx.cs" Inherits="Account_Login"%>

<asp:ContentID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
<linkhref="~/Styles/Style.css" rel="stylesheet" type="text/css"/>
</asp:Content>
<asp:ContentID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
<h1class="Sombras">
    Iniciar sesión
</h1>
<p>
    Especifique su nombre de usuario y contraseña.
</p>
<asp:LoginID="LoginUser" runat="server" EnableViewState="false" RenderOuterTable="false"
OnAuthenticate="LoginUser_Authenticate">
<LayoutTemplate>
<spanclass="failureNotification">
<asp:LiteralID="FailureText" runat="server"></asp:Literal>
</span>
<asp:ValidationSummaryID="LoginUserValidationSummary" runat="server" CssClass="failureNotifica
tion"
ValidationGroup="LoginUserValidationGroup"/>
<divclass="accountInfo">
<fieldsetclass="login">
<legend>Información de cuenta</legend>
<p>
<asp:LabelID="UserNameLabel" runat="server" AssociatedControlID="UserName">Nombre de
usuario:</asp:Label>
<asp:TextBoxID="UserName" runat="server" CssClass="textEntry" requiredautofocus></asp:TextBox>
<asp:RequiredFieldValidatorID="UserNameRequired" runat="server" ControlToValidate="UserName"
CssClass="failureNotification" ErrorMessage="El nombre de usuario es obligatorio."
ToolTip="El nombre de usuario es
obligatorio." ValidationGroup="LoginUserValidationGroup">*</asp:RequiredFieldValidator>
</p>
<p>
<asp:LabelID="PasswordLabel" runat="server" AssociatedControlID="Password">Contraseña:</asp:La
bel>
<asp:TextBoxID="Password" runat="server" CssClass="passwordEntry" TextMode="Password"
required></asp:TextBox>
<asp:RequiredFieldValidatorID="PasswordRequired" runat="server" ControlToValidate="Password"
CssClass="failureNotification" ErrorMessage="La contraseña es obligatoria." ToolTip="La
contraseña es obligatoria."
ValidationGroup="LoginUserValidationGroup">*</asp:RequiredFieldValidator>
</p>
<p>
<asp:CheckBoxID="RememberMe" runat="server"/>
```



```
<asp:LabelID="RememberMeLabel"runat="server"AssociatedControlID="RememberMe"CssClass="inline
">Mantenerme conectado</asp:Label>
</p>
</fieldset>
<divalign="right"class="imglogo">
<asp:ImageID="img"runat="server"ImageUrl="~/Imagenes/Login.png"/></div>
<pclass="submitButton">
<asp:ButtonID="LoginButton"runat="server"CommandName="Login"CssClass="boton"
Text="Iniciar sesión"ValidationGroup="LoginUserValidationGroup"/>
</p>
</div>
</LayoutTemplate>
</asp:Login>
</asp:Content>
```

ConexionMysql.cs

```
using System;
using System.Data;
using MySql.Data.MySqlClient;

namespace Acceso_Datos
{
    publicclassConexionMysql
    {
        //ORIGINAL
        //string connectionString = "Server=localhost;Database=SistemaItsco;Uid=root;Pwd='';";

        //PRUEBA
        string connectionString = "Server=localhost;Database=sistemaitscoprueba;Uid=root;Pwd='';";
        //HOSTING
        //string connectionString =
        "Server=mysql.azok.org;Database=u882501597_prueba;Uid=u882501597_admin;Pwd=victor123;";
        //SERVER2008
        //string connectionString = "Server=localhost;Database=proyectos;Uid=root;Pwd='admin';";

        publicDataTable ObtenerEscuelas()
        {
            string queryString = "SELECT esc_codigo As CODIGO , esc_nombre As NOMBRE " +
            "FROM escuelas ORDER BY esc_codigo";
            DataSet ds = newSystem.Data.DataSet();
            MySqlDataAdapter da = newMySqlDataAdapter();
            using (MySqlConnection cnn = newMySqlConnection(connectionString))
            {
                cnn.Open();

                MySqlCommand cmm = newMySqlCommand();
                cmm.CommandText = queryString;
                cmm.Connection = cnn;
                da.SelectCommand = cmm;

                da.Fill(ds);
                cnn.Close();
            }
            return ds.Tables[0];
        }

        publicDataTable ObtenerMax()
        {
```



```
string queryString = "SELECT MAX(esc_codigo) FROM escuelas";
DataSet ds = newSystem.Data.DataSet();
MySqlDataAdapter da = newMySqlDataAdapter();
using (MySqlConnection cnn = newMySqlConnection(connectionString))
{
    cnn.Open();

    MySqlCommand cmm = newMySqlCommand();
        cmm.CommandText = queryString;
        cmm.Connection = cnn;
        da.SelectCommand = cmm;

    da.Fill(ds);
    cnn.Close();
}
return ds.Tables[0];
}

publicDataTable ObtenerEscuelaspornombre(string nom)
{
    string queryString = "SELECT esc_codigo As CODIGO,esc_nombre As NOMBRE " +
    " FROM escuelas " +
    " WHERE esc_nombre LIKE '%" + nom + "%' ORDER BY esc_codigo ";
    DataSet ds = newSystem.Data.DataSet();
    MySqlDataAdapter da = newMySqlDataAdapter();

    using (MySqlConnection cnn = newMySqlConnection(connectionString))
    {
        cnn.Open();
        MySqlCommand cmm = newMySqlCommand();
            cmm.CommandText = queryString;
            cmm.Connection = cnn;
            da.SelectCommand = cmm;

        da.Fill(ds);
        cnn.Close();
    }
    return ds.Tables[0];
}

publicbool GuardarEscuela(string nomesc)
{
    try
    {
        string queryString = "INSERT INTO escuelas (esc_codigo,esc_nombre) " +
        " values (0 ,'" + nomesc + "')";

        DataSet ds = newSystem.Data.DataSet();
        MySqlDataAdapter da = newMySqlDataAdapter();

        using (MySqlConnection cnn = newMySqlConnection(connectionString))
        {
            cnn.Open();

            MySqlCommand cmm = newMySqlCommand();
                cmm.CommandText = queryString;
                cmm.Connection = cnn;
                da.SelectCommand = cmm;
        }
    }
}
```



```
da.Fill(ds);
cnn.Close();
    }
returntrue;
    }
catch (Exception)
    {
returnfalse;
    }
}

publicbool ActualizarEscuela(string codesc1, string nomesc1)
    {
try
    {
string queryString = "UPDATE escuelas " +
" SET esc_nombre='" + nomesc1 + "' " +
" WHERE esc_codigo='" + codesc1 + "'";
DataSet ds = newSystem.Data.DataSet();
MySqlDataAdapter da = newMySqlDataAdapter();

using (MySqlConnection cnn = newMySqlConnection(connectionString))
    {
cnn.Open();

MySqlCommand cmm = newMySqlCommand();
        cmm.CommandText = queryString;
        cmm.Connection = cnn;
        da.SelectCommand = cmm;

da.Fill(ds);
cnn.Close();
    }
returntrue;
    }
catch (Exception)
    {
returnfalse;
    }
}

publicbool EliminarEscuela(string codesc1)
    {
try
    {
string queryString = "DELETE FROM escuelas " +
" WHERE esc_codigo='" + codesc1 + "'";
DataSet ds = newSystem.Data.DataSet();
MySqlDataAdapter da = newMySqlDataAdapter();

using (MySqlConnection cnn = newMySqlConnection(connectionString))
    {
cnn.Open();

MySqlCommand cmm = newMySqlCommand();
        cmm.CommandText = queryString;
        cmm.Connection = cnn;
        da.SelectCommand = cmm;
```



```
da.Fill(ds);
cnn.Close();
    }
returntrue;
    }
catch (Exception)
{
returnfalse;
}
}

//*****TUTORES*****
publicbool GuardarTutores(string codigo, string rolcod, string esccod, string nom, string
ape, string tel, string correo, string clave)
{
try
{
string queryString = "INSERT " +
" INTO tutores
(tut_codigo,rol_codigo,esc_codigo,tut_nombre,tut_apellido,tut_telefono,tut_correo,tut_clave)
" +
" values ('" + codigo + "','" + rolcod + "','" + esccod + "','" + nom + "','" + ape + "','"
+ tel + "','" + correo + "','" + clave + "')";
DataSet ds = newSystem.Data.DataSet();
MySqlDataAdapter da = newMySqlDataAdapter();

using (MySqlConnection cnn = newMySqlConnection(connectionString))
{
cnn.Open();

MySqlCommand cmm = newMySqlCommand();
cmm.CommandText = queryString;
cmm.Connection = cnn;
da.SelectCommand = cmm;

da.Fill(ds);
cnn.Close();
}

returntrue;
}
catch (Exception)
{
returnfalse;
}
}

publicDataTable ObtenerTutores()
{

string queryString = "SELECT u.usu_codigo As CODIGO,u.usu_cedula AS CEDULA,e.esc_nombre
ESCUELA,u.usu_nombre As NOMBRE,u.usu_apellido As APELLIDO,u.usu_telefono As TELEFONO ,
u.usu_correo As CORREO " +
" FROM usuarios u , escuelas e ,funciones f ,roles r " +
" WHERE u.usu_codigo=f.usu_codigo " +
" AND f.rol_codigo=r.rol_codigo " +
" AND e.esc_codigo=f.esc_codigo";
DataSet ds = newSystem.Data.DataSet();
MySqlDataAdapter da = newMySqlDataAdapter();
```



```
using (SqlConnection cnn = newSqlConnection(connectionString))
    {
    cnn.Open();
    MySqlCommand cmm = newMySqlCommand();
        cmm.CommandText = queryString;
        cmm.Connection = cnn;
        da.SelectCommand = cmm;
    da.Fill(ds);
    cnn.Close();
    }
return ds.Tables[0];
}

}

ModificarDatosTutor.aspx.cs
using System;
using System.Collections;
using System.Web.UI.WebControls;

publicpartialclassFormularios_Tutores_CuentaTutor_ModificarDatosTutor : System.Web.UI.Page
{
publicstring password { get; set; }

protectedvoid Page_Load(object sender, EventArgs e)
    {
    IEnumerator sesiones = Application.Keys.GetEnumerator();
    string usuario;
    string password = Session["Tutores"].ToString();

    //if (!Context.Session.IsNewSession)
    //{
    if (!IsPostBack)
    {
    GridView drgDatos = newGridView();
        drgDatos.DataSource =
    Logica_de_Negocios.Logica_Tutores.ObtenerDatosdelUsuario(password);
    drgDatos.DataBind();

    string codigo, nombre, apellido, telefono, correo, pass, cedula;

    codigo = drgDatos.Rows[0].Cells[0].Text;
    nombre = drgDatos.Rows[0].Cells[3].Text;
    apellido = drgDatos.Rows[0].Cells[4].Text;
    telefono = drgDatos.Rows[0].Cells[5].Text;
    correo = drgDatos.Rows[0].Cells[6].Text;
    pass = drgDatos.Rows[0].Cells[7].Text;
    cedula = drgDatos.Rows[0].Cells[8].Text;
    if (codigo == "&nbsp;")
        {
        codigo = "";
        }
    if (nombre == "&nbsp;")
        {
        nombre = "";
        }
    }
}
```



```
    }
    if (apellido == "&nbsp;")
    {
        apellido = "";
    }
    if (telefono == "&nbsp;")
    {
        telefono = "";
    }
    if (correo == "&nbsp;")
    {
        correo = "";
    }
    if (cedula == "&nbsp;")
    {
        cedula = "";
    }

        txtCodigo.Text = codigo;
        txtEscuela.Text = drgDatos.Rows[0].Cells[2].Text;
txtNombre.Text = nombre;
        txtApellido.Text = apellido;
        txtTelefono.Text = telefono;
        txtCorreo.Text = correo;
txtPass.Text = pass;
        txtCedula.Text = cedula;

grdEscuelas.DataSource = Logica_de_Negocios.Logica_Tutores.ObtenerDatosdelUsuario(password);
grdEscuelas.DataBind();

    }
}

protectedvoid btnGuardar_Click(object sender, EventArgs e)
{

    Logica_de_Negocios.Logica_Usuarios.ModificarDatosDeUsuarios(Convert.ToInt32(txtCodigo.Text),
txtPass.Text, txtCedula.Text, txtNombre.Text, txtApellido.Text, txtTelefono.Text,
txtCorreo.Text);
    if
(Logica_de_Negocios.Logica_Usuarios.ModificarDatosDeUsuarios(Convert.ToInt32(txtCodigo.Text)
, txtPass.Text, txtCedula.Text, txtNombre.Text, txtApellido.Text, txtTelefono.Text,
txtCorreo.Text) == true)
    {
        Response.Write("<script>alert('Datos Modificados Con Éxito');</script>");
        grdEscuelas.DataSource =
Logica_de_Negocios.Logica_Tutores.ObtenerDatosdelUsuario(txtCorreo.Text);
grdEscuelas.DataBind();
    }
    else
    {
        Response.Write("<script>alert('Los Datos No Fueron Modificados');</script>");
    }
}
}

Avances.aspx.cs
```



```
using System;
using System.Collections;
using System.Web.UI.WebControls;
using System.Windows.Forms;

public partial class Formularios_Tutores_CuentaTutor_Avances : System.Web.UI.Page
{
    public string cedula { get; set; }

    protected void Page_Load(object sender, EventArgs e)
    {
        IEnumerator sesiones = Application.Keys.GetEnumerator();
        string usuario;
        string password = "";
        for (int i = 0; i < Application.Count; i++)
        {
            sesiones.MoveNext();
            usuario = sesiones.Current.ToString();
            password = Application[sesiones.Current.ToString()].ToString();
            cedula = password;
        }

        if (!IsPostBack)
        {
            CargarCapituloPorFecha();
            grdAvances.DataSource =
                Logica_de_Negocios.Logica_Avances.ObtenerAvancesPorTutor(cedula);
            grdAvances.DataBind();
        }

        public void CargarCapituloPorFecha()
        {
            GridView grd = new GridView();

            grd.DataSource =
                Logica_de_Negocios.Logica_Capitulos.CapituloPorFecha(DateTime.Now.Year + "/" +
                DateTime.Now.Month + "/" + DateTime.Now.Day);
            grd.DataBind();
            DateTime Inicial = new DateTime();
            DateTime Final = new DateTime();
            Inicial = Convert.ToDateTime(grd.Rows[0].Cells[2].Text);
            Final = Convert.ToDateTime(grd.Rows[0].Cells[3].Text);
            txtCapitulo.Text = grd.Rows[0].Cells[1].Text;
            lblF_Inicial.Text = Inicial.Year + "/" + Inicial.Month + "/" + Inicial.Day;
            lblF_Final.Text = Final.Year + "/" + Final.Month + "/" + Final.Day;

            cboItems.DataSource =
                Logica_de_Negocios.Logica_Capitulos.ItemsPorcapitulo(Convert.ToInt32(grd.Rows[0].Cells[4].Text));
            cboItems.DataValueField = "ITEMS";
            cboItems.DataBind();
        }

        protected void txtBuscar_TextChanged(object sender, EventArgs e)
    }
}
```



```
{
if (this.chkNombres.Checked == true)
{
    grdAvances.DataSource =
Logica_de_Negocios.Logica_Alumnos.BuscarAlumnosdeUsuario(cedula, txtBuscar.Text);
grdAvances.DataBind();
}
else
{
if (this.chkApellido.Checked == true)
{
    grdAvances.DataSource =
Logica_de_Negocios.Logica_Alumnos.BuscarAlumnosdeUsuarioPorApellido(cedula, txtBuscar.Text);
grdAvances.DataBind();
}
else
{
if (this.chkCedula.Checked == true)
{
    grdAvances.DataSource =
Logica_de_Negocios.Logica_Alumnos.BuscarAlumnosdeUsuarioPorCedula(cedula, txtBuscar.Text);
grdAvances.DataBind();
}
}
}
}

protectedvoid chkNombres_CheckedChanged(object sender, EventArgs e)
{
if (this.chkNombres.Checked == true)
{
    chkApellido.Checked = false;
    chkCedula.Checked = false;
this.txtBuscar.Enabled = true;
this.txtBuscar.Text = string.Empty;
}
else
{
this.txtBuscar.Enabled = false;
this.txtBuscar.Text = string.Empty;
}
}

protectedvoid chkApellido_CheckedChanged(object sender, EventArgs e)
{
if (this.chkApellido.Checked == true)
{
    chkNombres.Checked = false;
    chkCedula.Checked = false;
this.txtBuscar.Enabled = true;
this.txtBuscar.Text = string.Empty;
}
else
{
this.txtBuscar.Enabled = false;
this.txtBuscar.Text = string.Empty;
}
}
```



```
    }  
  
protectedvoid chkCedula_CheckedChanged(object sender, EventArgs e)  
{  
    if (this.chkCedula.Checked == true)  
    {  
        chkApellido.Checked = false;  
        chkNombres.Checked = false;  
        this.txtBuscar.Enabled = true;  
        this.txtBuscar.Text = string.Empty;  
    }  
    else  
    {  
        this.txtBuscar.Enabled = false;  
        this.txtBuscar.Text = string.Empty;  
    }  
}
```

Avances.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Avances.aspx.cs"Inherits="Formularios_Tutores_CuentaTutor_Avances"  
MasterPageFile="~/../Tutores.master"%>
```

```
<%@RegisterAssembly="Infragistics2.Web.v11.1, Version=11.1.20111.1006, Culture=neutral,  
PublicKeyToken=7dd5c3163f2cd0cb"
```

```
Namespace="Infragistics.Web.UI.LayoutControls"TagPrefix="ig"%>
```

```
<%@RegisterAssembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,  
PublicKeyToken=31bf3856ad364e35"
```

```
Namespace="System.Web.UI.DataVisualization.Charting"TagPrefix="asp"%>
```

```
<asp:ContentID="HeaderContent"runat="server"ContentPlaceHolderID="HeadContent">
```

```
<linkhref="~/Styles/Style.css"rel="stylesheet"type="text/css"/>
```

```
</asp:Content>
```

```
<asp:ContentID="BodyContent"runat="server"ContentPlaceHolderID="MainContent">
```

```
<asp:ScriptManagerID="ScriptManager1"runat="server">
```

```
</asp:ScriptManager>
```

```
<div>
```

```
<tablestyle="width: 100%;">
```

```
<tr>
```

```
<td>
```

```
<divstyle="width: 100%;">
```

```
<fieldset>
```

```
<legend>Datos del Alumno</legend>
```

```
<p>
```

```
<asp:LabelID="Label12"runat="server"Text="Buscar Alumno :
```

```
"CssClass="Sombras"></asp:Label><br/>
```

```
<asp:TextBoxID="txtBuscar"runat="server"CssClass="Cajas"AutoPostBack="True"OnTextChanged="txtBuscar_TextChanged"
```

```
autofocusEnabled="False"></asp:TextBox>
```

```
<asp:CheckBoxID="chkNombres"runat="server"Text="Nombres"AutoPostBack="True"OnCheckedChanged="chkNombres_CheckedChanged"/><asp:CheckBox
```

```
ID="chkApellido"runat="server"Text="Apellidos"OnCheckedChanged="chkApellido_CheckedChanged"AutoPostBack="true"/>
```

```
<asp:CheckBoxID="chkCedula"runat="server"Text="Cédula"OnCheckedChanged="chkCedula_CheckedChanged"
```

```
AutoPostBack="true"/>
```

```
<br/>
```




```
<asp:CommandFieldButtonType="Image"HeaderText="SELECCIONAR"SelectImageUrl="~/Imagenes/Proyec  
to/Editar.png"  
ShowSelectButton="True"/>  
</Columns>  
</asp:GridView>  
</div>  
</Template>  
</ig:ContentTabItem>  
<ig:ContentTabItemrunat="server"Text="No Presentados">  
</ig:ContentTabItem>  
<ig:ContentTabItemrunat="server"Text="Inactivos">  
</ig:ContentTabItem>  
</Tabs>  
</ig:WebTab>  
</div>  
</asp:Content>
```

Register.aspx.cs

```
using System;
```

```
using System.Web.UI.WebControls;
```

```
publicpartialclassAccount_Register : System.Web.UI.Page
```

```
{
```

```
protectedvoid Page_Load(object sender, EventArgs e)
```

```
{
```

```
if (!IsPostBack)
```

```
{
```

```
    cboRol.DataSource = Logica_de_Negocios.Logica_Roles.ObtenerRoles();
```

```
    cboRol.DataValueField = "DESCRIPCION";
```

```
    cboRol.DataBind();
```

```
        txtConfirmaPass.Enabled = false;
```

```
        txtConfirmaPass.Visible = false;
```

```
        lblcofirmypass.Visible = false;
```

```
        Label2.Text = string.Empty;
```

```
    cboEscuelas.DataSource = Logica_de_Negocios.Logica_Escuelas.ObtenerEscuelas();
```

```
    cboEscuelas.DataValueField = "NOMBRE";
```

```
    cboEscuelas.DataBind();
```

```
}
```

```
  }
```

```
protectedvoid CreateUserButton_Click(object sender, EventArgs e)
```

```
{
```

```
    GridView grdCorreo = newGridView();
```

```
        grdCorreo.DataSource =
```

```
        Logica_de_Negocios.Logica_Usuarios.ObtenerUsuariosRepetidos(this.Email.Text);
```

```
    grdCorreo.DataBind();
```

```
    if (grdCorreo.Rows.Count > 0)
```

```
    {
```

```
        //Response.Write("<script>alert('El Usuario con Este Correo ya Existes Compruebe Sus  
Datos');</script>");
```

```
        Mensajes("Usuarios", "'El Usuario con Este Correo ya Existes Compruebe Sus Datos", "error");
```

```
        Email.Text = string.Empty;
```

```
        Email.Focus();
```

```
    }
```



```
else
{
Response.Write("<script>alert('Los Datos de Su Cuenta Seran Enviados a su Correo
Electronico');</script>");

if (cboRol.SelectedValue.ToString() == "ADMINISTRADOR")
{
}
else
{
if (cboRol.SelectedValue.ToString() == "TUTOR")
{
GridView gridRoles = newGridView();
gridRoles.DataSource =
Logica_de_Negocios.Logica_Roles.ObtenerRolesporNombre(cboRol.SelectedValue);
gridRoles.DataBind();

GridView gridEscuelas = newGridView();
gridEscuelas.DataSource =
Logica_de_Negocios.Logica_Escuelas.ObtenerEscuelasporNombre(cboEscuelas.SelectedValue);
gridEscuelas.DataBind();

string rol, codtut, nombre, clave, codesc, correo;
codesc = gridEscuelas.Rows[0].Cells[0].Text;
rol = gridRoles.Rows[0].Cells[0].Text;
nombre = UserName.Text.Trim();
clave = GenerarClave();

correo = Email.Text.Trim();

Response.Write(codesc + " " + rol + " " + nombre + " " + clave + " " + correo);
Logica_de_Negocios.Logica_Usuarios.GuardarUsuarios(clave, nombre,
correo, "A");

protectedvoid cboRol_SelectedIndexChanged(object sender, EventArgs e)
{
if (cboRol.SelectedValue.ToString() == "ADMINISTRADOR")
{
txtConfirmaPass.Enabled = true;
txtConfirmaPass.Visible = true;
lblcofirmpass.Visible = true;
}
else
{
if (cboRol.SelectedValue.ToString() == "TUTOR")
{
txtConfirmaPass.Enabled = false;
txtConfirmaPass.Visible = false;
lblcofirmpass.Visible = false;
GridView gridRoles = newGridView();

gridRoles.DataSource =
Logica_de_Negocios.Logica_Roles.ObtenerRolesporNombre(cboRol.SelectedValue);
gridRoles.DataBind();
}
}
}
}
```



```
    }

    public string GenerarClave()
    {
        string pass = "";
        int numero = 0, numlt = 0;

        Random rnd = new Random();
        Random rnd1 = new Random();
        string letra = "";
        int ascii;

        while (pass.Length < 8)
        {
            numero = rnd.Next(9);

            numlt = rnd1.Next(122);
            if ((numlt >= 65 && numlt <= 90) || (numlt >= 97 && numlt <= 122))
            {
                ascii = numlt;

                letra = letra + Convert.ToChar(ascii) + numero;
            }
            else
            {
                numlt = rnd1.Next(122);
            }

            pass = letra;
        }
        return pass;
    }

    private void Mensajes(string titulo, string mensaje, string tipmensaje)
    {
        string s = "<script lenguaje=\"javascript\">" + "javascript:showDialog('" + titulo + "', '" +
        mensaje + "', '" + tipmensaje + "');" + "</script>";
        RegisterStartupScript("mensaje", s);
    }

    private void MensajesTiempo(string titulo, string mensaje, string tipmensaje, int tiempo)
    {
        string s = "<script lenguaje=\"javascript\">" + "javascript:showDialog('" + titulo + "', '" +
        mensaje + "', '" + tipmensaje + "', 4);" + "</script>";
        RegisterStartupScript("mensaje", s);
    }

    //Tipos de Mensajes(error,warning,success,prompt)
}
```



Recurso Económico

DETALLE	COSTO
COMPUTADORAS	3.000
PAPEL BOND	100
IMPRESORA	400
SISTEMA DE TINTA	200
VARIOS	300
TOTAL	4.000

Recurso Económico Tabla N°21

Recurso Humano

Nombre	Descripción
VICTOR CHIZA	ALUMNO
ING. JAIME BASANTES	TUTOR

Recurso Económico Tabla N°22