

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“GESTIÓN ACADÉMICA VÍA WEB, APLICANDO
ERGONOMÍA Y USABILIDAD, PARA LA CARRERA DE CIENCIAS
DE LA EDUCACIÓN”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS

Postulante : Edgar Joaquin Arteaga Gutierrez
Tutor : Lic. Mario Loayza Molina
Revisor : Lic. Franz Cuevas Quiroz Mg. Sc.

La Paz – Bolivia

2008

DEDICATORIA

Dedico mi Proyecto de Grado a:

A Dios por que ha sido lo más importante en mi vida, a quien debo todos mis logros porque siempre ha estado a mi lado, guiándome por el buen camino. A mi madre Esther y a mi padre Juan, a quienes quiero muchísimo y admiro porque siempre me han apoyado.

A mis hermanos y amigos, que confiaron y confían en mí.

Gracias a mi abuelo Isaac que siempre me ha cuidado y guiado desde donde está.

AGRADECIMIENTOS

A Dios, por cuidarme siempre, guiándome por el buen camino, a mis padres Juan y Esther por darme su apoyo incondicional, a mis hermanos Marcelo y Bryan por ayudarme siempre, gracias por ser parte de mi vida.

Al Lic. Mario Loayza Molina y a el Lic. Franz Cuevas Quiroz, por su comprensión y disposición que me dieron para el desarrollo de este proyecto, gracias a su asesoramiento pude presentar y defender este proyecto.

Al Lic. Orlando Huanca Rodríguez y a Lic. Maria Angelica Carbajal, por confiar y dar todo su apoyo y comprensión, para que este proyecto se pueda ver la luz y ser de utilidad.

A mis amigos, que fueron un gran apoyo en todo el transcurso de mi vida en la universidad, con lo cuales compartí alegrías y tristezas, a los cuales se que nunca voy a olvidar y espero seguir contando con ellos en la vida profesional, gracias.

Edgar Joaquin Arteaga Guttierrez

RESUMEN

De un tiempo a esta parte las tecnologías de comunicaciones están evolucionando de una manera vertiginosa en nuestro País, cambiando el modo de ver las aplicaciones Web, donde la primicia ya no es poner la información en línea solamente, sino poder contar con aplicaciones Web tengan un alto grado de ergonomía y usabilidad, mecanismos avanzados de seguridad, con el fin de aprovechar y salvaguardar el recurso más importante para una institución, como lo es la información.

Por otra parte podemos ver que la administración académica es fundamental, para cualquier institución de educación superior, los estudiantes las autoridades y el plantel docente, requieren información, para la buena toma de decisiones, todas estas tareas podrían ser optimizadas mediante la utilización de aplicaciones Web.

La carrera de ciencias de la educación de la UMSA, no cuenta con un sistema confiable para la gestión académica, dados los actuales procedimientos utilizados en el manejo de la información de la carrera, se encuentran diversos problemas, originados por la falta de automatización de dichos procesos.

Por lo anteriormente mencionado se ha desarrollado e implementado un nuevo Sistema de Gestión Académica en la Carrera de Ciencias de la Educación, para tal efecto se han utilizado metodologías orientadas a objetos, modelos de diseño aplicando ingeniería de la usabilidad Web y ergonomía Web, enfocándose en un diseño centrado en el usuario.

De los resultados obtenidos mediante una evaluación cualitativa se puede concluir que el sistema es de fácil aprendizaje, pero en lo que respecta al manejo se encuentra en el límite permitido por la tecnología, considerando que se está trabajando en un entorno web, en el cual se debe guardar un equilibrio entre la funcionalidad y la ligereza de la aplicación

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1 - MARCO REFERENCIAL	2
1.1 ANTECEDENTES	2
1.2 DEFINICIÓN DEL PROBLEMA.....	3
1.2.1 PLANTEAMIENTO DEL PROBLEMA	3
1.2.2 SITUACIÓN PROBLEMÁTICA	3
1.3 OBJETIVOS	4
1.3.1 OBJETIVO GENERAL.....	4
1.3.2 OBJETIVOS ESPECÍFICOS.....	4
1.4 JUSTIFICACIÓN	5
1.5 MÉTODOS Y TÉCNICAS	6
1.6 ALCANCES Y APORTES	7
CAPÍTULO 2 - MARCO TEÓRICO.....	10
2.1 GESTIÓN Y ADMINISTRACIÓN ACADÉMICA DE LA CARRERA DE CIENCIAS DE LA EDUCACIÓN	10
2.2 PROCEDIMIENTOS DE LA CARRERA.....	11
2.3 PLATAFORMA TECNOLÓGICA.....	12
2.3.1 APLICACIÓN WEB.....	12
2.3.2 BASE DE DATOS.....	13
2.3.3 SISTEMAS DE MANEJO DE BASE DE DATOS (DBMS).....	13
2.4 METODOLOGÍA DE DESARROLLO DEL SOFTWARE	13
2.4.1 METODOLOGÍA ORIENTADA A OBJETOS.....	14
2.4.2 METODOLOGÍA OOHDM	15
2.4.3 ERGONOMÍA	19
2.4.5 ERGONOMÍA Y USABILIDAD, DISEÑO CENTRADO EN EL USUARIO	20
2.4.6 USABILIDAD.....	22
2.5 CALIDAD DE SOFTWARE	26
2.6 GRUPOS DE DISCUSIÓN DIRIGIDO (FOCUS GROUP)	26
2.7 SEGURIDAD DE LA INFORMACIÓN	27

2.7.1	ISO/IEC 17799:2005 TECNOLOGÍA DE LA INFORMACIÓN. CÓDIGO DE BUENAS PRÁCTICAS PARA LA GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN.....	28
2.8	HERRAMIENTAS Y TECNOLOGÍAS	29
2.8.1	ENTERPRICE ARCHITECT	29
2.7.1	SERVIDOR DE BASE DE DATOS SQL SERVER 2000	29
2.7.2	PLATAFORMA MICROSOFT .NET	31
2.7.3	VISUAL ESTUDIO .NET 2005.....	32
2.7.4	ASP .NET.....	32
2.7.5	VISUAL BASIC .NET	33
2.7.6	AJAX.....	33
CAPÍTULOS III - DESARROLLO DEL SOFTWARE		34
3.1	SITUACIÓN ACTUAL.....	34
3.2	MODELO DE REQUISITOS	34
3.2.1	MODELO DEL DOMINIO DEL PROBLEMA	34
3.2.2	MODELO DE NEGOCIO	36
3.2.3	CASOS DE USO	38
3.3	MODELO DE ANÁLISIS	45
3.3.1	ARQUITECTURA DE CLASES.....	45
3.3.2	IDENTIFICACIÓN DE CLASES SEGÚN ESTEREOTIPOS	46
3.4	MODELO DE DISEÑO.....	47
3.4.1	DISEÑO DEL MODELO CONCEPTUAL	47
3.4.2	DISEÑO NAVEGACIONAL.....	49
3.4.3	DISEÑO INTERFAZ ABSTRACTA	51
CAPÍTULO IV – IMPLEMENTACIÓN Y PRUEBAS		53
4.1	MODELO IMPLEMENTACIÓN	53
4.2	REPRESENTACIÓN DE LOS MÓDULOS DEL SISTEMA COMPLETO.....	53
4.3	REQUERIMIENTOS MÍNIMOS.....	54
4.4	INTERFAZ DE USUARIO	54
4.5	MODELO FÍSICO DE LA BASE DE DATOS	65
4.6	SEGURIDAD.....	66
4.7	VALIDACIÓN Y MANEJO DE ERRORES	67
4.8	MODELO DE PRUEBAS	68

4.8.1 PRUEBA DE CAJA NEGRA	68
4.8.2 PRUEBA DE CAJA BLANCA.....	72
CAPÍTULO V – CALIDAD DE LA APLICACIÓN WEB.....	73
5.1 CALIDAD DE SOFTWARE	73
5.2 EVALUACIÓN DE USABILIDAD (FOCUS GROUP).....	76
5.3 ANÁLISIS COSTO/BENEFICIO.....	79
CONCLUSIONES.....	81
RECOMENDACIONES.....	81
BIBLIOGRAFÍA.....	83



LISTA DE FIGURAS

Figura 3.1: Diagrama de clases con asociaciones y roles entre clases identificadas	35
Figura 3.2: Actores identificados para el Sistema Académico	37
Figura 3.3: Diagrama de Casos de uso de Alto nivel	38
Figura 3.4: Diagrama de caso de uso del administrador del sitio	43
Figura 3.5: Diagrama de caso de uso del docente	44
Figura 3.6: Diagrama de caso de uso del estudiante	44
Figura 3.7: Diagrama del Modelo Conceptual del Sistema Académico	48
Figura 3.8: Diagrama navegacional para el usuario Administrador	49
Figura 3.9: Diagrama navegacional para el usuario Docente	50
Figura 3.10: Diagrama navegacional para el usuario Estudiante	50
Figura 3.11: ADV Administrador	51
Figura 3.12: ADV Docente	52
Figura 3.13: ADV Docente	52
Figura 4.1: Diagrama de módulos del Sistema Académico	53
Figura 4.2 , Autenticación del Sistema de Información	55
Figura 4.3, Funciones y transacciones del rol Administrador	56
Figura 4.4 Administración de materias por carrera	57
Figura 4.5 Llenado de prerrequisitos	58
Figura 4.6 Búsqueda de docentes y registro	59
Figura 4.7 Habilitación de materias y paralelos para la inscripción	59
Figura 4.8 Pensum grafico de materias aprobadas, inscritas y convalidadas	60
Figura 4.9 Inscripción de materias por el administrador	60
Figura 4.10 Reporte de Alumnos Egresados por gestión	61
Figura 4.11 Listado de alumnos por Materia	61
Figura 4.12 Boletas de Inscripción por Gestión	62
Figura 4.13 Actas de notas	62
Figura 4.14 Reporte de Historiales Académicos	63
Figura 4.15 Pantalla de Ingreso del Estudiante	63
Figura 4.16 Reporte de Materias inscritas	64
Figura 4.17: Modelo físico de Base de Datos del Sistema Académico	65
Figura 4.18: Usuario para la conexión a la Base de Datos.	66
Figura 4.19: Validación de entrada de datos	67
Figura 4.20: Diagrama de caja negra	68
Figura 4.21: Grado de la prueba de caja negra.	68
Figura 4.22: Camino que se sigue en el caso de ingresar datos incorrectos	70
Figura 4.23: Grafo completo para registrar un nuevo proyecto	71
Figura 4.24: Diagrama de prueba de caja blanca.	72

LISTA DE TABLAS

Tabla 3.1: Clases Identificadas	34
Tabla 3.2: Descripción Actor Administrador	37
Tabla 3.3: Descripción Actor Docente	37
Tabla 3.4: Descripción Actor Estudiante	38
Tabla 3.5: Caso de uso Validar usuario	39
Tabla 3.6: Caso de uso Registrar usuario	39
Tabla 3.7: Caso de uso Planificación Académica	40
Tabla 3.8: Caso de uso Reportes	41
Tabla 3.9: Caso de uso inscripciones	41
Tabla 3.10: Caso de uso Seguimiento Académico	42
Tabla 4.1: Valores de las condiciones de entrada	70
Tabla 4.2: Valores de las condiciones de entrada	71
Tabla 5.1, Cálculo de puntos de función	74
Tabla 5.2, Valoración de ajustes de Complejidad	75
Tabla 5.3 Descripción de Costos	79



INTRODUCCIÓN

De un tiempo a esta parte las tecnologías de comunicaciones están evolucionando de una manera vertiginosa en nuestro País, cambiando el modo de ver las aplicaciones Web, donde la primicia ya no es poner la información en línea solamente, sino poder contar con aplicaciones Web tengan un alto grado de ergonomía y usabilidad, mecanismos avanzados de seguridad, con el fin de aprovechar y salvaguardar el recurso más importante para una institución, como lo es la información.

Por otra parte podemos ver que la administración académica es fundamental, para cualquier institución de educación superior, debido a que en la misma se realizan procesos como inscripciones, adición y/o retiro de materias, convalidaciones, emisión de actas y pre-actas, etc., los estudiantes necesitan tener información acerca de su estado académico, las autoridades y el plantel docente, requieren información, para la buena toma de decisiones, todas estas tareas podrían ser optimizadas mediante la utilización de aplicaciones Web.

Para contribuir al logro de los objetivos de la Carrera de Ciencias de la Educación se ve necesario el replanteamiento de los actuales procesos de gestión académica. El replanteamiento que se propone en este proyecto es la automatización de dichos procesos utilizando las ventajas obvias del uso de la Red Internet como base de comunicaciones, ventajas que serán cada vez más valiosas conforme Internet esté disponible para más y más personas en el medio actual. Es por esta razón que el presente proyecto apunta hacia la automatización del Sistema de Gestión Académica de la Carrera de Ciencias de la Educación para prestar un mejor servicio a sus estudiantes y docentes.



En el presente trabajo se pretende mostrar la utilización de las metodologías de ergonomía y usabilidad para el diseño de una aplicación Web, apuntando hacia la automatización del sistema de gestión académica, de la Carrera de Ciencias de la Educación, con el fin de poder prestar un mejor servicio a sus estudiantes y docentes.

CAPÍTULO 1 - MARCO REFERENCIAL

1.1 ANTECEDENTES

La carrera de Ciencias de la Educación, como las demás carreras del sistema universitario boliviano, ha entrado al mundo de la Web, con el desarrollo de un Sistema Académico, el mismo que fue diseñado en el año 2003, elaborado por el Ing. Marco Salazar, pero el Modelo Informacional no llegó a satisfacer en su totalidad las necesidades de los usuarios debido a la magnitud de requerimientos por parte de la carrera y las particularidades que tiene en los procesos académicos, ya que la mayor parte de los procesos que se desarrollan actualmente con la información académica deben hacerse de forma manual, no cuenta con mecanismos de seguridad avanzados y se tuvo problemas durante la inscripción debido problemas en el algoritmo de generación de materias a cursar, produciendo errores en la inscripción.

Como otros antecedentes al proyecto se tienen: el "Sistema Inteligente de Gestión Académica" (SIGA) de la carrera de Informática elaborado por el Lic. Franz Cuevas y estudiantes de la carrera; el "Sistema Integrado Académico" (SIA) desarrollado por la empresa Élite Data System, que a la fecha está en funcionamiento en la carrera de Informática. El "Sistema Académico para Programas de Post Grado: Seguimiento y consultas mediante Internet para la Universidad NUR", el cual fue desarrollado por el postulante Fabián Medina Gemio. En la facultad de Odontología se desarrolló el proyecto de grado de la carrera de Informática titulado "Sistema de Administración Académica Vía Web Facultad de Odontología". En la Unidad de Postgrado en Informática existen antecedentes de un proyecto de grado de la Carrera de Informática con el tema "Sistema de la Administración Académica de Postgrado en Informática vía Web" elaborado por la postulante Silvana Vidal.

1.2 DEFINICIÓN DEL PROBLEMA

1.2.1 PLANTEAMIENTO DEL PROBLEMA

La carrera de ciencias de la educación de la UMSA, no cuenta con un sistema confiable para la gestión académica, dados los actuales procedimientos utilizados en el manejo de la información de la carrera, se encuentran diversos problemas, originados por la falta de automatización de dichos procesos.

1.2.2 SITUACIÓN PROBLEMÁTICA

- El sistema actual no se adapta a los requerimientos de información, que se tiene de parte de las autoridades de la carrera, como ser estadística de inscritos, promedios generales por materia, rendimiento de alumnos por materia, estadísticas de egresados.
- No son explotados al máximo los recursos informáticos que se poseen, teniendo deficiencias en su funcionamiento.
- Dificultad en los procesos de inscripción debido a que el sistema actual no es flexible en cuantos a ciertas políticas que tiene la carrera.
- Dificultad en la toma de decisiones debido a la falta de disponibilidad de información histórica y estadística
- No se tiene cuenta con un control eficiente sobre el ingreso de alumnos nuevos y egreso, ya que este trabajo se lo realiza de forma manual.
- El sistema actual no cuenta con mecanismos óptimos de seguridad, siendo un riesgo para la integridad y fiabilidad de la información.
- Record académico e historial académico: los estudiantes tienen que esperar para verificar sus notas, muchas veces no son atendidos rápidamente debida al trabajo arduo que tienen los kardixtas.
- No se tiene el registro de notas que son modificadas por causas de error al transcribir o por resolución.
- Registro de notas por los docentes: presentan sus notas en actas impresas para luego ser transcritas, lo cual demora mucho tiempo.

Con esto se plantea el siguiente problema:

¿El sistema de gestión académica vía Web, será capaz de mejorar los procesos de inscripciones y consultas, proporcionando a los usuarios la información la requerida?

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar e implementar un sistema de gestión académica vía Web, para la carrera de ciencias de la educación, garantizando la usabilidad de la aplicación final con alcance a los diferentes estamentos y sectores que componen la organización de la Carrera.

1.3.2 OBJETIVOS ESPECÍFICOS

- Implementar modulo de validación de usuarios, para el control de seguridad el cual permitirá que el sistema pueda ser utilizado y accesado por usuarios autorizados, según las políticas de la institución, mediante la utilización de roles.
- Implementar modulo de consultas, para que los alumnos puedan realizar consultas sobre su situación académica (record académico, materias a cursar, boleta de inscripción, notas, horarios, ficha del alumno y estructura del plan de estudios.
- Mejorar el grado de usabilidad y mantenimiento de la aplicación, mediante la utilización de interfaces ergonómicas y arquitectura de tres capas.
- Implementar módulos de inscripción y retiro/adición óptimos, utilizando la estructura de datos Hashtables (diccionarios de datos), en los procesos de selección de materias a llevar, para reducir el tiempo de ejecución en las consultas a la base de datos,
- Generar reportes estadísticos que permitan a la institución la obtención de información estadística que pueda ser utilizada para la evaluación de la institución, sobre la base de toda los datos cargados en el sistema agrupados en: estadística de inscritos, promedios generales por materia, rendimiento de alumnos por materia, estadísticas de egresados.

1.4 JUSTIFICACIÓN

TÉCNICA

La importancia de este trabajo radica en la necesidad de aplicar métodos de diseño para aplicaciones Web, utilizando tecnologías actualmente existentes para la gestión de la base de datos, y así también la utilización de paginas ASP.NET 2.0 (Active Server Pages), permitiendo al sistema comunicación remota entre cliente y un servidor Web.

OPERATIVA

El sistema tendrá el efecto de reducir y descongestionar de gran manera el accionar operativo, que se presenta en épocas de inscripciones y consultas de notas y materias a llevar.

Además podemos contar con la capacitación necesaria para interactuar con el sistema a través de diferentes seminarios, manuales e información técnica del mismo, con el fin de garantizar una buena implantación de los objetivos propuestos.

SOCIAL

Los beneficiarios del sistema principalmente son los alumnos que requieren de la información sobre su situación académica con la universidad, lo cual posibilita el ahorro de tiempo y esfuerzo en el acceso a esta información. También se beneficia el personal de la Universidad, que cuenta con información oportuna y transparente sobre los procesos académicos.

ECONÓMICA

Con la implementación del sistema se reducirán los costos de papelería, comunicaciones y publicaciones.

Reducción de las horas extras de trabajo, por parte del personal encargado de la administración académica.

El aprovechamiento de los equipos computacionales y acceso a Internet con el que actualmente se cuenta, lo que incide en el nivel de depreciación de los equipos.

El valor agregado al conjunto de activos fijos en la Carrera de Ciencias de la Educación

1.5 MÉTODOS Y TÉCNICAS

Las etapas del presente trabajo se realizarán de manera sistemática y programada, es por esta razón para un buen estudio de la metodología a seguir para el desarrollo del Sistema debemos utilizar el método científico, el cual nos permitirá realizar una observación profunda de cada uno de los acontecimientos a seguir, la medición y experimentación correcta a partir de:

- Las fuentes de Observación
- Los fenómenos observables
- Resultado de datos y observaciones
- La base para la investigación a partir de los datos que nos proporcionen

Para la realización de este perfil del presente trabajo se ha utilizado la metodología del Marco lógico.

La metodología a utilizar para el desarrollo del sistema será la OOHDM (Metodología de diseño de Hipermedia Orientada a Objetos) y comprenderá los siguientes aspectos.

- Diseño conceptual
- Diseño de navegación
- Diseño de interfaz abstracta
- Implementación

En lo que se refiere al diseño conceptual y de navegación, van separados porque tratan de temas diferentes:

- Modelo conceptual y diseño, reflejan objetos y comportamientos
- Diseño de navegación tiene en cuenta perfiles y tareas de los usuarios

Se construirá un modelo del dominio de la aplicación usando principios de modelado orientado a objetos, utilizando UML, las clases conceptuales se crean usando agregación y jerarquías de generalización o especialización obteniendo la semántica del dominio neutral como resultado de esquemas de clases e instancias creadas a partir de Subsistemas, Clases y Relaciones.

El modelo conceptual no refleja el hecho de que la aplicación se implementara e el entorno World Wide Web (el modelo de aplicación principal se construirá durante el diseño de navegación)

Las clases del modelo conceptual se mapearán a nodos en el modelo de navegación usando un mecanismo de vistas, las relaciones se usaran para definir los enlaces entre nodos, hay otros enlaces que no se corresponden con las relaciones del modelo conceptual que implica:

- Definir objetos de la interfaz que percibirá el usuario
- La manera en que se mostraran los objetos de navegación
- Los objetos de la interfaz que activaran la navegación
- La manera en que se sincronizan los objetos multimedia de la interfaz
- Transformaciones de la interfaz

1.6 ALCANCES Y APORTES

Inscripción a cursar materias: el sistema inscribe a los alumnos para cursar en las asignaturas habilitadas por la Facultad aplicando los controles establecidos tales como el cumplimiento de prerrequisitos definidos en fechas y horarios establecidos.

Egresados: el sistema automáticamente pasa al alumno en condición de egresado cuando el mismo cumplió con todo lo establecido en su plan de estudios.

Convalidaciones: en muchas ocasiones la institución recibe a alumnos de otras universidades en varios términos como por ejemplo trasladado, por convenio, etc.

Eventualmente el alumno ya pudo haber cursado algunas materias y éstas ser reconocidas por la institución. El sistema prevé que las aprobaciones de esta forma puedan registrarse como convalidaciones.

También es posible que la institución tenga sistemas de convalidaciones automáticas entre carreras. Esto se tiene dentro del sistema con el nombre de equivalencias entre asignaturas. La equivalencia puede establecerse como simple: una asignatura es equivalente a otra, o como múltiple: una asignatura es equivalente a una serie de asignaturas.

Consistencia e integridad de información: en todo momento el sistema chequea que la información a ser modificada no tenga efectos en la consistencia de la información. Se controla por ejemplo: que un alumno no tenga nunca dos aprobaciones, que un alumno no se inscriba a una materia que ya aprobó, que no se pueda modificar una nota cuando la asignatura es prerrequisito de otra, etc.

Otros controles asociados son los denominados técnicamente “de integridad referencial”. Por ejemplo, no se puede borrar un alumno que tiene notas dentro del sistema, o que tenga inscripciones u otra información relacionada.

Seguridad en el acceso a la información: es posible definir, según las políticas de la institución, el acceso restringido a la información. Opcionalmente se tiene la posibilidad de definir roles de acceso en el Sistema.

Reportes estadísticos: este módulo del sistema permite a la institución la obtención de información estadística sobre la base de todos los datos cargados en el sistema.

Esta información puede ser utilizada para la evaluación de la institución en cuanto a la gestión académica y también para hacer análisis y obtener conclusiones sobre el principal estamento que constituye el estudiantado.

Los reportes generados tienen formato de planilla y también en algunos casos pueden generarse reportes gráficos comparativos.

Los reportes generados están agrupados en: estadística de inscritos, promedios generales por materia, rendimiento de alumnos por materia, estadísticas de egresados.

Consultas y procesos vía Web: con esta facilidad el alumno puede realizar consultas a informaciones mantenidas en el sistema como ser: record académico, materias inscritas, notas finales, ficha del alumno, estructura del plan de estudios (lista de materias con prerrequisitos).

Con esta posibilidad los datos incluidos en el sistema académico quedan disponibles, eventualmente, para su consulta, con un navegador de páginas Web, desde cualquier lugar que tenga acceso a la red Internet.

La aplicación de técnicas de estructuras de datos, para realizar la optimización de los procesos de los módulos de inscripción del sistema y realización de consultas.

La utilización de una arquitectura de sistema de tres capas pensada para un entorno multiusuario, facilitando la escalabilidad y mantenibilidad de la aplicación Web.

Utilizando interfaces ergonómicas necesarias para garantizar la usabilidad de la aplicación final.

La Carrera de Ciencias de la Educación se convierte en pionera al implementar un sistema de estas características, dando así un ejemplo para las otras Carreras de la Facultad de humanidades de la UMSA.

La Aplicación de Gestión Académica aumentará la confianza y prestigio de la Carrera de Ciencias de la Educación, en sus procesos de administración y control.

CAPÍTULO 2 - MARCO TEÓRICO

2.1 GESTIÓN Y ADMINISTRACIÓN ACADÉMICA DE LA CARRERA DE CIENCIAS DE LA EDUCACIÓN

El plan de estudios es la estructura de información neural del sistema. En base a él giran gran parte de los controles que se realiza. El plan de estudios refleja la estructura curricular establecida para cada carrera.

El Sistema de Registro e Historial Académico es el instrumento de aplicación obligatoria en la Carrera de Ciencias de la Educación, que permite recopilar, almacenar, procesar y difundir la información relativa a la permanencia de los docentes y estudiantes de los cursos.

La información de los docentes incluye hojas de vida, participación en la docencia del postgrado, participación en la vida universitaria, participación en proyectos de investigación e interacción social, evaluación y reportes de publicaciones y sanciones disciplinarias, entre otros.

Los datos personales de alumnos que son mantenidos son los siguientes: documento de identidad, teléfono, dirección, estado civil, fecha y lugar de nacimiento, colegio de procedencia, datos de los padres.

El ingreso completo de esta información permite luego a la unidad académica mediante del módulo estadístico, tener informaciones interesantes sobre la conformación del alumnado de la Universidad.

El sistema de Registro e Historial académico es operado en la Oficina de Kardex de Postgrado, en el cual se mantiene registro y el archivo de los documentos que respaldan la información de la Carrera de Ciencias de la Educación.

El Sistema de Seguimiento Académico es el conjunto de normas y procedimientos para monitorear la ejecución del Programa de Actividades Académicas, cuya aplicación

es obligatoria, el Director de Carrera debe verificar el cumplimiento de los objetivos y metas del Programa de Actividades Académicas, analizando los resultados, los recursos empleados y los tiempos utilizados en cada actividad; además, deben recomendar propuestas de medidas correctivas, en caso de existir variaciones en la ejecución, con respecto a lo programado.

El Sistema de Evaluación de la Gestión Académica es el conjunto de normas y procedimientos para evaluar los resultados de la gestión académica en base a la información obtenida de la verificación del Sistema de Seguimiento Académico, a los indicadores de eficacia y eficiencia y otros aspectos que permitan medir el cumplimiento de las actividades académicas de la carrera. La evaluación de los resultados permitirá generar medidas correctivas, cuya aplicación puede originar la realización de ajustes en los planes de estudio. El responsable de la operación de este sistema es el Consejo de Carrera.

2.2 PROCEDIMIENTOS DE LA CARRERA

Registro de nuevos estudiantes.- Una vez que los estudiantes aprobaron el curso prefacultativo o el examen de dispensación, la oficina de kardex se encarga de recibir la documentación de los estudiantes y registrarlos en el sistema.

Inscripción a cursar materias.- Se inscribe a los alumnos para cursar en las asignaturas habilitadas por la Carrera aplicando los controles establecidos tales como el cumplimiento de prerrequisitos definidos, matriculación en la carrera a la que inscribe, permanencia en la carrera, cantidad máxima de materias que puede cursar, etc.

La habilitación de las asignaturas es establecida por la Carrera y se encuentra relacionada a la estructura del plan de estudios.

Evaluaciones parciales.- Son las pruebas parciales, laboratorios, trabajos prácticos, etc. Estas definiciones se establecen en forma particular por cada cátedra lo que permite tener flexibilidad en cuanto a las ponderaciones para la obtención de la habilitación a los exámenes finales.

Registro de notas.- Los docentes deben enviar sus notas de manera manuscrita en los listados de actas de notas de cada materia, que son entregados por la dirección de

carrera, una vez recibidos, los mismos son transcritos a un sistema en FOXPRO, para realizar la impresión de certificados de notas e historiales académicos.

Egresados.- El alumno pasa en condición de egresado cuando el mismo cumplió con todo lo establecido en su plan de estudios.

Convalidaciones.- En muchas ocasiones la institución recibe a alumnos de otras universidades en varios términos como por ejemplo trasladado, por convenio, etc. Eventualmente el alumno ya pudo haber cursado algunas materias y éstas ser reconocidas por la institución. Se registran estas aprobaciones como convalidaciones.

2.3 PLATAFORMA TECNOLÓGICA

2.3.1 APLICACIÓN WEB

En ingeniería de software, una aplicación Web (Web Apps) es aquella que permite a los usuarios acceder a una gran variedad de recursos a través de Internet o de una intranet. La popularidad de estas aplicaciones ha crecido, debido a los múltiples servicios que se ofrecen y a las nuevas oportunidades que presenta para las organizaciones. Las aplicaciones Web son accedidas a través de un navegador Web como cliente ligero, sin necesidad que el usuario tenga que instalar aplicaciones o software específico.

Los atributos y las características que debe tener una aplicación Web, según Pressman son las siguientes:

- Intensiva de Red. Porque reside en la red y proporciona servicios que satisfacen las necesidades de los clientes.
- Evolución continua. A diferencia de las aplicaciones convencionales, las aplicaciones Web están en constante evolución, por esta razón se debe crear una estructura inicial que permita crecer.
- Inmediatez. Las aplicaciones Web deben ser desarrolladas en tiempos muy cortos.
- Seguridad. Dado que una aplicación Web está disponible en la red, es necesario restringir el acceso para usuarios autorizados. Además se deben tomar ciertas medidas para proteger el contenido.
- Estética. La interfaz que se le presente al usuario debe ser amigable, de tal manera que el usuario se encuentre a gusto mientras va revisando el contenido.

2.3.2 BASE DE DATOS

Base de datos se puede definir "Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones". (Conference des Statisticiens Européens, 1977).

Es una colección de archivos interrelacionados, son creados con sistema manejador de Base de Datos. El contenido de una base de datos engloba a la información concerniente de una organización, de tal forma que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

2.3.3 SISTEMAS DE MANEJO DE BASE DE DATOS (DBMS)

El DBMS es considerado la parte medular de la base de datos que permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de reportes.

Los objetivos de un DBMS incluyen:

- Asegurarse de que la base de datos pueda ser compartida entre los usuarios de una diversidad de aplicaciones.
- Asegurarse que todos los datos requeridos para las aplicaciones actuales y futuras estén fácilmente disponibles.
- Permitir que los usuarios construyan su vista personal de los datos sin preocuparse de la forma en que estén físicamente guardados los datos.
- Manejar datos que sean preciso y consistentes.

2.4 METODOLOGÍA DE DESARROLLO DEL SOFTWARE

Para el desarrollo del software, se utilizará dos metodologías que contemplan el ciclo de vida del software, por un lado la metodología orientada a objetos para la captura de requisitos, el análisis y parte diseño, y por otro lado será complementado con la metodología OOHDM que nos proporciona el modelo navegacional y el diseño de interfaces para la Web.

2.4.1 METODOLOGÍA ORIENTADA A OBJETOS

La metodología orientada a objetos propuesta por Alfredo Weitzenfeld, en su libro “Ingeniería de Software Orientada a Objetos con UML y Java”, esta se apoya en el Lenguaje de Modelado Unificado (UML por la siglas de Unified Modeling Language) que es el estándar mas importante a nivel mundial para el modelado de sistemas orientado a objetos. Esta metodología se basa en la obtención de los siguientes modelos:

- a) **Modelo de Requisitos**, tiene como meta definir y delimitar la funcionalidad del sistema de software, sirve como base de negociación entre el desarrollador y el cliente, y debe reflejarse los deseos de este
- b) **Modelo de Análisis**, tiene como objetivo construir una arquitectura capaz de resolver el problema bajo condiciones ideales en base a la aprobación previa del cliente, significa desarrollar una estructura lógica del sistema, la cual debe ser estable y extensible.
- c) **Modelo de Diseño**, el propósito de este es extender la arquitectura de análisis, el diseño se concentra en dos aspectos principales: *diseño de objetos* que incluye la selección de la estructura de datos para satisfacer los objetivos de rendimiento y espacio, y también la descripción de operaciones y atributos de los objetos; y por otro lado tenemos *diseño de sistemas* que define las decisiones estratégicas sobre cómo organizar la funcionalidad del sistema en tomo al ambiente de implementación, es necesario identificar las restricciones técnicas bajo las cuales se tiene que construir el sistema.
- d) **Modelo de Implementación**, toma el resultado del modelo de diseño para generar el código final del sistema. Esta traducción debe ser relativamente sencilla y directa.
- e) **Modelo de Pruebas**, es el responsable de revisar la calidad del sistema. Este consiste en la *validación del sistema* o prueba de especificación y la *verificación* o prueba de resultado.

2.4.2 METODOLOGÍA OOHDH

OOHDM (Object Oriented Hypermedia Design Method) es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones de la Web. En sus comienzos no contemplaba la fase de captura y definición de requisitos, pero actualmente propone el uso de *user interaction diagrams* (UIDs). Esta propuesta parte de los casos de uso que considera una técnica muy difundida, ampliamente aceptada y fácilmente entendible por los usuarios y clientes no expertos, pero que resulta ambigua para el equipo de desarrollo en fases posteriores del ciclo de vida. Igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos web, teniendo un claro y amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema.

Partiendo de estas dos premisas, OOHDH propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos los analistas elaboran los UIDs. Estos UIDs son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz. El proceso de transformación de un caso de uso a un UIDs es descrito detalladamente en la propuesta, y se basa en detectar la interacción necesaria para la realización del caso de uso. OOHDH centra el desarrollo de un sistema de información web entorno del modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema, pero los casos de uso resultan demasiado ambiguos para ello. Así, propone refinar el proceso de desarrollo descrito en UML de forma que de los casos de uso se generen los UIDs que concreten más la definición de los requisitos para partir de ellos, obtener el diagrama conceptual.

Esta metodología utiliza notación estándar para representar los modelos (UML) y está compuesta por las siguientes fases:

- Especificación de requisitos.
- Diseño conceptual.
- Diseño navegacional.
- Diseño de Interfaz abstracta.
- Implementación.

a) Especificación de requisitos

Esta es una fase que forma parte del proceso de desarrollo de cualquier sistema, donde se realiza la identificación de requisitos. La identificación de requerimientos que el sistema

debe satisfacer, es una tarea compleja, debido a que a veces el usuario no expresa lo que realmente necesita, y se hace necesario en varias oportunidades, revisarlas con el usuario, pero para realizar esta tarea, los requerimientos deben estar representados de tal manera, que el usuario pueda entenderlas en el momento de la revisión. Por esta razón, utilizar técnicas estructuradas es una buena práctica para abordar el problema.

El desarrollo de aplicaciones Web, agrupa una serie de características que lo hacen diferente a otros sistemas, la más importante es la estructura navegacional, que obliga un desarrollo preciso de este aspecto, o sea una interfaz adecuada, para que el usuario no se pierda en el espacio navegacional del sistema.

Para esta fase se comienza por elaborar casos de uso, los cuales describen como el usuario va a comunicarse o utilizar el sistema (a esto se denomina detección de acciones que el usuario va a realizar con la aplicación).

b) Diseño conceptual

El diseño conceptual es muy importante en la metodología OOHDM, porque a partir de este modelo se comenzará a construir la estructura de la aplicación Web. En esta fase se realiza el análisis del dominio del problema y se identifican los conceptos involucrados en el sistema, dicho de otra forma, se elabora el esquema conceptual, mediante diagrama de clases (UML), que está representado por las clases abstractas y objetos, las relaciones y colaboraciones que existen entre los objetos.

En la metodología OOHDM, el modelo conceptual está constituido por clases, relaciones y subsistemas. Las clases son representadas como en el modelo orientado a objeto tradicional, en las relaciones se utiliza agregación y generalización/especialización, pero los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Las clases que se logran obtener son similares a las del modelado orientado a objetos, donde una clase es una descripción de un conjunto de objetos con las mismas propiedades que tienen significado en el sistema. Las relaciones representan unión entre las clases, pudiendo ser relaciones de asociación, agregación, generalización o

especialización. Los atributos de las clases pueden ser diferentes tipos y representan propiedades de las clases.

c) **Diseño navegacional**

Una vez obtenido el esquema conceptual se procede a la fase de diseño navegacional, la navegación es un aspecto muy relevante, porque aquí se va a definir la estructura de navegación que tendrá la aplicación Web. Un modelo navegacional es construido como una vista sobre el diseño conceptual, en esta fase se admite la construcción de diferentes modelos, esto debido a los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual.

El modelo navegacional también se describe utilizando el enfoque orientado a objetos, es decir, que este modelo está compuesto por objetos, creados a partir de los objetos conceptuales, a estos objetos se les llama clases navegacionales.

El diseño de la navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de los contextos navegacionales.

Las clases navegacionales, son los nodos, enlaces y estructuras de acceso que determinan los objetos que se presentarán a los usuarios.

- Los nodos, son contenedores básicos de información, representados por un conjunto de clases identificadas en la fase del diseño conceptual, además tienen definidos atributos (incluidos los enlaces) y un conjunto de métodos (comportamiento) cuando se navega por la aplicación, la figura 2.2 muestra la descripción gráfica de un nodo.

Los contextos navegacionales, organizan el espacio o estructura navegacional en conjuntos convenientes, asignándoles un orden particular, definiendo como rutas que ayudan al usuario a revisar el contenido de la aplicación y evitando de esta manera que se pierda.

Los contextos navegacionales están compuestos por conjuntos de nodos, enlaces de clases de contexto y otros contextos navegacionales.

d) Diseño interfaz abstracta

Después de definir las estructuras navegacionales, se procede a describir la interfaz que tendrá la aplicación. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad del sistema, que transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

Los modelos que van a realizarse en esta fase permiten elaborar diferentes interfaces, que son independientes a las plataformas tecnológicas que se utilizarán para la implementación.

El modelo de interfaz Vista de Datos Abstracta (Abstract Data View - ADVs), especifica la organización y el comportamiento de la interfaz. Los ADVs son objetos que tienen un estado y una interfaz, donde la interfaz puede ser ejercida a través de mensajes. Los ADVs son abstractos porque solo representan interfaz y estado, y no la aplicación.

Un ADV comprende un conjunto de atributos (y objetos de interfaz anidado) que define sus propiedades de percepción, y un conjunto de eventos que puede manejar, como eventos generados por el usuario. Los ADVs pueden ser fácilmente implementados en ambientes orientados a objetos para la Web que pueden traducirse a documentos HTML.

e) Implementación

Durante esta fase el Webmaster implementa el diseño. Los anteriores modelos se elaboraron independientemente de la plataforma de implementación, pero ahora se toma en cuenta el entorno en el que va a correr la aplicación y cuáles son las herramientas que se utilizarán.

Lo primero que se debe realizar es, definir los ítems de información que forman parte del dominio. Luego se debe decidir cómo serán organizados los ítems, de acuerdo a los perfiles de usuario y a sus roles; la interfaz que se le presentará al usuario, cuál será el comportamiento del sitio y qué información deberá ser almacenada.

2.4.3 ERGONOMÍA

Es el campo de conocimientos multidisciplinarios que estudia las características, necesidades, capacidades y habilidades de los seres humanos, analizando aquellos aspectos que afectan al entorno artificial construido por el hombre relacionado directamente con los actos y gestos involucrados en toda actividad de éste. En todas las aplicaciones su objetivo es común: se trata de adaptar los productos, las tareas, las herramientas, los espacios y el entorno en general a la capacidad y necesidades de las personas, de manera que mejore la eficiencia, seguridad y bienestar de los consumidores, usuarios o trabajadores.

2.4.4 ERGONOMÍA EN LA INGENIERÍA DE SISTEMAS

El desarrollo del conocimiento sobre el Factor Humano y tecnológico permite a los ingenieros de sistemas proyectar herramientas, máquinas, equipos, procedimientos, etc, con unas elevadas prestaciones. Sin embargo, a veces, se pasan por alto los límites de actuación y las capacidades/posibilidades de respuesta de las personas, que son el factor fundamental y crítico de **cualquier** sistema, esencialmente en su dinámica, y en su fiabilidad, ya que, sea por **fallos** debidos a errores humanos en el uso, o a fallos cometidos en la etapa de concepción del sistema, estos se salen de programa por cuestiones que siempre son responsabilidad de las personas.

El problema básico de todo proyecto de ingeniería de sistemas estriba en encontrar el quantum de restricciones que configuran el árbol funcional que debe aportar este nuevo subsistema al sistema Persona-Máquina (P-M), y es ahí donde la Ergonomía como ciencia pluridisciplinar aporta los datos básicos para conformar esta relación de forma armoniosa.

El handicap inicial, y que acostumbra a viciar de origen los proyectos de ingeniería de sistemas, estriba en que mientras que para los objetos poseemos métodos de intervención que nos permiten delimitar las superficies, las durezas, los vértices, las tolerancias y el grado de fiabilidad, etc..., para el Factor Humano estos límites son mucho más difusos, cambiantes, e incluso, a veces, contradictorios.

Si queremos conseguir sistemas de ingeniería de calidad, debemos recurrir necesariamente al uso sistemático de la Ergonomía, lo que nos lleva a considerar y diseñar con una serie de premisas "extras", y que podemos cifrar, en:

No siempre es posible determinar las condiciones iniciales de un fenómeno. Salvo en condiciones excepcionales, raramente podemos prever con total exactitud las respuestas de las personas, pues pueden aparecer reacciones impredecibles (miedo, apatía, nerviosismo, fatiga...) e incluso aberrantes, que no estén implicadas, necesariamente, en las condiciones previas.

No siempre una causa produce el mismo efecto. Ya que procesos imprevistos alteran los resultados; lo que funciona y es fiable con una persona puede no serlo con otras. Las consecuencias finales de las acciones humanas en ocasiones no son predecibles. Pequeñas causas pueden degenerar en efectos desastrosos, y errores importantes, incluso en el proyecto de concepción, puede que sólo provoquen efectos mínimos, debido, por ejemplo, a la pericia de los usuarios.

Como primera aproximación, podríamos definir la Ergonomía como la ciencia aplicada que trata de alcanzar el mayor equilibrio posible entre las necesidades/posibilidades del usuario y las prestaciones/requerimientos de los sistemas, con el fin de ayudar al equipo de proyectos de ingeniería a crear sistemas seguros, eficaces y confortables. [P Mondelo y E Torada, 2008]

2.4.5 ERGONOMÍA Y USABILIDAD, DISEÑO CENTRADO EN EL USUARIO

La mayor parte de los sistemas informáticos se diseñan usando una aproximación vertical, de arriba abajo, y basándose en una descomposición funcional. El ciclo de vida asociado con este tipo de metodología comienza con el análisis de la aplicación y continua secuencialmente por las fases de especificación de requerimientos, diseño, prototipado, implementación y evaluación.

Al progresar el desarrollo de una fase a la siguiente secuencialmente, este método recibe el nombre de *método de catarata*. La existencia inicial de especificaciones formales que describan minuciosamente lo que se pretende conseguir, asegura en gran modo la corrección y resulta como en cualquier actividad de desarrollo o proyecto, esencial para garantizar la conclusión de un objetivo, la calidad de la aplicación desarrollada y el auditado del proceso. Pero en la práctica la implementación o el desarrollo de sistemas complejos no pueden realizarse estrictamente en modo secuencial, de principio a fin, debiendo procederse en muchos casos por tanteos, borradores o bosquejos sucesivos que permitan corregir las desviaciones observadas.

Cuando se adopta la iteración como una necesidad práctica, se está trabajando con una metodología en espiral. Esta aproximación implica un movimiento circular repetitivo a través de todo el proceso, como estrategia de autocorrección.

De la propia experiencia en el desarrollo de aplicaciones hipertexto se deriva, que una cierta flexibilización en esta metodología es necesaria, sobre todo si se opta además por la solución simultánea de problemas concretos. Con este enfoque, en realidad se pueden considerar flujos alternativos de actividades complementarias: actividades creativas y concretas de síntesis se alternarán en el proceso con actividades estructuradas y abstractas de análisis. En el proceso que describimos, es necesaria con frecuencia una aproximación más compleja, ya que las secciones de una aplicación no son entidades aisladas sino que en muchas ocasiones constituyen fracciones lógicas o comparten un modelo de información. [Art20]

En las aplicaciones convencionales el énfasis se centra sobre el procedimiento, pero no hay en general una distinción clara entre éste y los datos procesados. Se ha decidido utilizar un paradigma de diseño orientado a objetos porque sus mecanismos de abstracción y composición permiten una descripción concisa de espacios complejos de información. Las metodologías de Aproximaciones Orientadas a Objetos se han utilizado en el ámbito hipertexto desde hace tiempo, si bien en la mayoría de los casos los objetos han sido utilizados tan sólo como artefactos para la implementación. Al contrario de aquellas, la metodología presentada en este trabajo se orienta a los aspectos de diseño, presentando la interacción con el sistema desde el dominio del comportamiento. Así esta metodología se integra en el ciclo de vida en estrella. Esta aproximación permite:

- En primer lugar, una separación clara de los patrones de navegación y de la interfaz de usuario respecto de las particularidades de la implementación. Esta separación favorecerá la reutilización y el mantenimiento del modelo de diseño.
- Además en esta metodología el desarrollo de la interactividad del usuario puede comenzar, teóricamente, con casi cualquier actividad del mismo y continuar con casi cualquier otra. Las actividades están altamente interconectadas, pero pasando por un centro que es la evaluación. Por esto se considera que el ciclo de vida está centrado en la evaluación: el resultado de cada actividad es evaluado antes de comenzar con otra nueva.

2.4.6 USABILIDAD

En el concepto tradicional de calidad de un sistema software, la usabilidad se presenta como un atributo del

software al que se presta especial atención en determinados proyectos software, de forma similar a como se trata la seguridad, por ejemplo. Este enfoque provoca que un gran número de sistemas software construidos tengan un nivel de usabilidad deficiente, cuando un mayor cuidado por este tema hubiera producido un sistema de mayor calidad percibida por el cliente sin un aumento excesivo en el coste de desarrollo.

Sin embargo, la creciente demanda de software más usable está cambiando este panorama. En las versiones más recientes de los estándares de ISO sobre calidad del software, ya se puede observar cómo la calidad de un sistema se distingue entre calidad inherente del software y **calidad de uso** [ISO98a] e [ISO00].

Delimitación del Concepto de Usabilidad

Usabilidad se define en el estándar ISO 9241 como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” [ISO98b], y en el estándar ISO 14598-1 [ISO98a] se define calidad de uso de forma análoga. Como se puede comprobar, en esta definición se liga la usabilidad de un sistema a usuarios, necesidades y condiciones específicas. Por tanto, la usabilidad del sistema no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios concretos que vayan a utilizar el sistema.

La Ingeniería de Usabilidad se puede definir como una aproximación al desarrollo de sistemas en la que se especifican niveles cuantitativos de usabilidad a priori, y el sistema se construye para alcanzar dichos niveles, que se conocen como métricas [Preece94]. La Ingeniería de Usabilidad proporciona un modo práctico de asegurar que el software desarrollado alcanza un cierto nivel de usabilidad y está basada en la evaluación mediante test de usabilidad con usuarios.

La usabilidad no puede definirse como un atributo simple de un sistema, pues implicará aspectos distintos dependiendo del tipo de sistema a construir. Por ejemplo, un sistema para ser colocado en una pantalla táctil de la sala de un museo, debe llevar un software que requiera un entrenamiento mínimo con el sistema por parte del usuario, debido a que será utilizado por la mayoría de usuarios una sola vez en toda su vida. Aspectos de

usabilidad como la eficiencia (número de tareas que se pueden realizar por hora) no son relevantes para este tipo de sistemas, mientras de que la facilidad de aprendizaje es vital para el éxito del sistema. Por otra parte, el software que utiliza una teleoperadora para recoger los mensajes a enviar a un buscapersonas, puede requerir de un periodo de formación apreciable, pero interesa fundamentalmente que pueda realizar cada operación en el menor tiempo posible (eficiencia), para reducir el tiempo de espera de los clientes que utilizan el servicio.

Estos distintos aspectos de la usabilidad se denominan atributos de usabilidad.

Atributos de Usabilidad

La usabilidad es una cualidad demasiado abstracta como para ser medida directamente. Para poder estudiarla se descompone habitualmente en los siguientes cinco atributos básicos [Nielsen93]:

- **Facilidad de aprendizaje:** Cuán fácil es aprender la funcionalidad básica del sistema, como para ser capaz de realizar correctamente la tarea que desea realizar el usuario. Se mide normalmente por el tiempo empleado con el sistema hasta ser capaz de realizar ciertas tareas en menos de un tiempo dado (el tiempo empleado habitualmente por los usuarios expertos). Este atributo es muy importante para usuarios noveles.
- **Eficiencia:** El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez. Nótese que eficiencia del software en cuanto su velocidad de proceso no implica necesariamente eficiencia del usuario en el sentido en el que aquí se ha descrito.
- **Recuerdo en el tiempo:** Para usuarios intermitentes (que no utilizan el sistema regularmente) es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez. Este atributo refleja el recuerdo acerca de cómo funciona el sistema que mantiene el usuario, cuando vuelve a utilizarlo tras un periodo de no utilización.
- **Tasa de errores:** Este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario, y pueden verse

como un fracaso en la transmisión al usuario del modo de hacer las cosas con el sistema.

- **Satisfacción:** Éste es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.

Algunos de estos atributos no contribuyen a la usabilidad del sistema en la misma dirección, pudiendo ocurrir que el aumento de uno de ellos tenga como efecto la disminución de otro. Por ejemplo, esto puede ocurrir con la facilidad de aprendizaje y la eficiencia. Es preciso realizar el diseño del sistema cuidadosamente si se desea tanto una alta facilidad de aprendizaje como una alta eficiencia; siendo el uso de aceleradores (combinaciones de teclas que ejecutan operaciones de uso habitual) la solución más común para conjugar ambos atributos de usabilidad.

La usabilidad del sistema no es una simple adición del valor de estos atributos, sino que se define para cada sistema como un nivel a alcanzar para algunos de ellos.

Estos cinco atributos pueden descomponerse a su vez para conseguir una mayor precisión en los aspectos de usabilidad en los que se quiere poner mayor énfasis. Por ejemplo, *rendimiento en uso normal* y *uso de opciones avanzadas* son ambos subatributos de eficiencia, mientras que *primera impresión* es un subatributo de satisfacción.

Relación entre la Usabilidad y la Interfaz Gráfica de Usuario

En el desarrollo de software se identifica a menudo la usabilidad con las características de los elementos de una interfaz gráfica de usuario basada en ventanas, como puede ser su color, su disposición o el diseño gráfico de los iconos y animaciones. Sin embargo, la usabilidad no sólo tiene que ver con la interfaz gráfica de usuario.

La usabilidad de un sistema está ligada principalmente a la interacción del mismo, al modo en que se realizan las operaciones con el sistema. Esta interacción no está definida en la interfaz gráfica, sino que está imbricada en el código que implementa la funcionalidad del sistema. La interfaz gráfica de usuario es la parte visible de tal interacción. Es cierto que la interfaz gráfica es una parte importante del sistema, y un buen diseño de la misma puede hacer que un sistema aumente su nivel de usabilidad, pero un sistema con un diseño de la interacción pobre no puede mejorar su nivel de usabilidad tan solo cambiando la interfaz gráfica.

En cuanto a usabilidad, la parte más crítica es la lógica del sistema (el concepto en base al cual funciona). Por tanto, la interacción debe diseñarse junto con la lógica de negocio, para asegurarnos de que la lógica del sistema es usable. No es posible diseñar la lógica

de negocio independientemente de la interacción y luego unirlos. Esta relación entre diseño de la interacción y el diseño software tradicional obliga a modificar este último para acomodar al primero.

Por otra parte, la usabilidad se ocupa también de lo que es el entorno del sistema software propiamente dicho.

Por ejemplo, se ocupa del sistema de ayuda, de la documentación de usuario, y del procedimiento de instalación.

¿Por qué Aplicar Ingeniería de Usabilidad?

La principal razón por la cual aplicar la Ingeniería de Usabilidad cuando se desarrolla un sistema software, es la obtención de un sistema que hace al usuario más productivo, aumentando su eficiencia y satisfacción al utilizarlo.

La usabilidad es un tema crítico para la aceptación de un sistema: Si el sistema no es percibido como una herramienta que ayuda al usuario a realizar sus tareas, se dificulta la aceptación del sistema. Puede ocurrir que el sistema no llegue a usarse en absoluto, o que se use con escasa eficiencia. Si las tareas del usuario no son respaldadas convenientemente por el sistema, entonces no se está respondiendo adecuadamente a las necesidades del usuario, y el equipo de desarrollo se está alejando del objetivo principal de la construcción de un sistema software.

De cara a la organización de desarrollo de software también es importante invertir en la usabilidad de los sistemas que construye. Puede ocurrir que se detecten errores graves de usabilidad en un momento cercano al final estimado del proyecto, y entonces el tiempo de desarrollo puede crecer en demasía para corregir tales problemas. Por otra parte, hay que tener en cuenta los elevados costes del servicio de atención al usuario que pueden derivarse de un sistema con deficiencias graves de usabilidad. En un mercado altamente competitivo también puede ser una seria amenaza el lanzamiento por parte de un competidor de un sistema con un nivel alto de usabilidad, y frente a tal posibilidad es muy importante invertir en usabilidad para poder mantener o ampliar la cuota de mercado.

2.5 CALIDAD DE SOFTWARE

La calidad del software es una meta muy importante para un desarrollador, y sobre todo para las compañías que se dedican al desarrollo de software, porque un software de calidad representa un ahorro en los costes y una mejora de en los productos. Por esta razón, la calidad del software se ha mejorado significativamente en los últimos quince años [SOM05], este importante cambio se debe a que las compañías han adoptado nuevas técnicas y tecnologías para el desarrollo, y además porque se están adoptando técnicas y estándares para ofrecer productos de calidad.

Sin embargo, medir la calidad en un producto software, no es una tarea fácil, porque no se puede comparar o medir la calidad como en la manufactura de productos. La utilización de estándares y técnicas permiten de alguna una forma de garantizar la calidad en el software.

Definir que es calidad de software, de manera precisa y universal, es decir, que pueda ser aplicada a cualquier producto software es complicado, pero vamos ha utilizar la definición de Pressman:

“La calidad es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.”

2.6 GRUPOS DE DISCUSIÓN DIRIGIDO (FOCUS GROUP)

El *focus group* o grupo de discusión dirigido es una técnica de recogida de datos donde se reúnen de 6 a 9 usuarios para discutir aspectos relacionados con el sistema. Un evaluador experto en usabilidad y/o accesibilidad (dependiendo de lo que deseemos evaluar) hace las veces de moderador. Éste preparará previamente la lista de aspectos a discutir y recoger la información que necesita de la discusión.

Esto puede permitir capturar reacciones espontáneas del usuario e ideas que evolucionan en el proceso dinámico del grupo.

El método puede ser aplicado en cualquiera de las fases de ciclo de vida del proceso de software, aun así en las que más se ha aplicado (y por tanto de la que se tiene mayor experiencia) es en la fase de lanzamiento.

Procedimiento

Los procedimientos generales para dirigir un focus group son:

- Localizar usuarios representativos (típicamente 6 a 9 por focus group) que quieran participar.
- Seleccionar un moderador.
- Preparar una lista de temas a ser discutidos y objetivos a asumir por los temas propuestos.
- Controlar la discusión sin inhibir el flujo libre de ideas y comentarios.
- Asegurar que todos los participantes contribuyen a la discusión.
- Procurar que no haya un participante que domine la discusión.
- Conservar la discusión que discorra libremente y no estructurada, pero procura que siga un esquema planeado.
- Escribir un resumen de las opiniones que han prevalecido y comentarios críticos de la sesión incluyendo cuotas representativas.

Aspectos a considerar

Al dirigir un *focus group* hay una serie de aspectos importantes a tener en cuenta:

- Tener **más de un grupo principal**, puesto que *el resultado de una sola sesión puede no ser representativo* y una sola discusión pudo haberse centrado en un subconjunto de las ediciones o de los aspectos de menor importancia del sistema.
- Es preferible que el moderador tenga **dotes dinamizadores y comunicativos**.

No es tan simple como preparar las preguntas, porque el moderador necesita facilitar y dirigir la discusión en tiempo real.

- Debe dudarse siempre de los **datos** recogidos porque debido a su naturaleza desestructurada y de flujo libre tienden a tener una **validez cuestionable** y son muy **difíciles de analizar**.

2.7 SEGURIDAD DE LA INFORMACIÓN

La información es un activo que, como otros activos importantes del negocio, tiene valor para la organización y requiere en consecuencia una protección adecuada. Esto es muy importante en el creciente ambiente interconectado de negocios. Como resultado de esta creciente ínter conectividad, la información esta expuesta a un mayor rango de amenazas y vulnerabilidades. La información adopta diversas formas. Puede estar impresa o escrita en papel, almacenada electrónicamente, transmitida por el correo o por medios

electrónicos, mostrada en video o hablada en conversación. Debe protegerse adecuadamente cualquiera sea la forma tome o los medios por los que se comparta o almacene.

La seguridad de la información se consigue implantando un conjunto adecuado de controles como ser políticas, procedimientos, estructuras organizativas y funciones de software y hardware. Estos controles necesitan ser establecidos, implementados, monitoreados, revisados y mejorados donde sea necesario, para asegurar que se cumplan los objetivos específicos de seguridad y negocios de la organización.

Esta norma nos servirá como una guía práctica par desarrollar estándares organizacionales de seguridad y practicas efectivas de la gestión de seguridad.

2.7.1 ISO/IEC 17799:2005 TECNOLOGÍA DE LA INFORMACIÓN. CÓDIGO DE BUENAS PRÁCTICAS PARA LA GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN

Esta norma ofrece recomendaciones para realizar la gestión de la seguridad de la información que pueden utilizarse por los responsables de iniciar, implantar o mantener y mejorar seguridad en una organización. Persigue proporcionar una base común para desarrollar normas de seguridad dentro de las organizaciones y ser una práctica eficaz de la gestión de la seguridad.

La estándar ISO 17799, nos ayuda a definir cláusulas para la gestión de seguridad en una aplicación. La seguridad de la información se consigue implementando un conjunto adecuado de controles que pueden ser políticas, prácticas, procedimientos, estructuras organizativas y funciones de hardware y software. Estos controles necesitan ser establecidos, implementados, monitoreados, revisados y mejorados donde sea necesario, para asegurar que se cumplan los objetivos específicos de seguridad y negocios de la organización.

Entre los puntos que se hará énfasis, podemos mencionar: políticas de seguridad, organización de la seguridad de la información, seguridad de recursos humanos, seguridad física, control de acceso.

2.8 HERRAMIENTAS Y TECNOLOGÍAS

2.8.1 ENTERPRICE ARCHITECT

Enterprise Architect (EA) es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software. EA soporta la especificación UML, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto.

EA es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio.

Algunas de las características claves de Enterprise Architect son:

- ✓ Crear elementos del modelo UML para un amplio alcance de objetivos.
- ✓ Ubicar esos elementos en diagramas y paquetes.
- ✓ Crear conectores entre elementos.
- ✓ Documentar los elementos que ha creado.
- ✓ Generar código para el software que está construyendo.
- ✓ Realizar ingeniería reversa del código existente en varios lenguajes.

EA le ayuda a visualizar las aplicaciones soportando ingeniería inversa de un amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos. Ingrese frameworks completos desde código fuente o archivos Java .jar - o aún ensambladores binarios .Net! Importando frameworks y librerías de código.

EA soporta transformaciones de Arquitectura avanzada dirigida por Modelos (MDA) usando plantillas de transformaciones de desarrollo y fáciles de usar. Con transformaciones incorporadas para DDL, C#, Java, EJB y XSD, Se puede rápidamente desarrollar soluciones complejas desde los simples "modelos independientes de plataforma" (MIP) que son el objetivo en "modelos específicos de plataforma" (MEP). Un MIP se puede usar para generar y sincronizar múltiples MIP's - proveyendo un aumento de productividad significativo.

2.7.1 SERVIDOR DE BASE DE DATOS SQL SERVER 2000

SQL Server es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR), desarrollado por Microsoft, que permite, como su propio nombre indica, la gestión de un

entorno de bases de datos relacional. SQL Server abarca, tanto el área de diseño, como la de administración, proporcionando un interfaz bastante amigable con el usuario.

SQL Server utiliza una extensión al SQL estándar, que se denomina *Transact SQL*. Esto quiere decir que soporta el SQL de ANSI, pero además se le han añadido ciertas funciones adicionales, no contempladas en el estándar, y que son específicas para este producto, es decir, si ejecutamos una sentencia del conjunto adicional (*Transact SQL*) en otro SGBRD, éste no la entendería.

El *Transact SQL*, soporta la definición, modificación y eliminación de bases de datos, tablas, atributos, índices, etc., es decir, el lenguaje de definición de datos (DDL), así como la consulta, actualización y borrado de tuplas de tablas, es decir, el lenguaje de manipulación de datos (DML).

SQL Server 2000 está diseñado para trabajar con dos tipos de bases de datos:

- **OLTP (OnLine Transaction Processing)** Son bases de datos caracterizadas por mantener una gran cantidad de usuarios conectados concurrentemente realizando ingreso y/o modificación de datos. Por ejemplo: entrada de pedidos en línea, inventario, contabilidad o facturación.
- **OLAP (OnLine Analytical Processing)** Son bases de datos que almacenan grandes cantidades de datos que sirven para la toma de decisiones, como por ejemplo las aplicaciones de análisis de ventas.

SQL Server puede ejecutarse sobre redes basadas en Windows Server así como sistema de base de datos de escritorio en máquinas Windows XP, Windows NT Workstation, Windows Millenium y Windows 98.

Los entornos *Cliente/Servidor*, están implementados de tal forma que la información se guarde de forma centralizada en un computador central (servidor), siendo el servidor responsable del mantenimiento de la relación entre los datos, asegurarse del correcto almacenamiento de los datos, establecer restricciones que controlen la integridad de datos, etc.

Existen muchas formas de organizar la información pero una de las formas más efectivas de hacerlo está representada por las bases de datos relacionales, las cuales están basadas en la aplicación de la teoría matemática de los conjuntos al problema de la organización de los datos. En una base de datos relacional, los datos están organizados en tablas (llamadas relaciones en la teoría relacional) [J. Moratalla, 2001].

2.7.2 PLATAFORMA MICROSOFT .NET

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones. Esto quiere decir que no sólo nos brinda todas las herramientas y servicios que se necesitan para desarrollar modernas aplicaciones empresariales y de misión crítica, sino que también nos provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima. Los componentes principales de la plataforma .NET son:

- Un entorno de ejecución de aplicaciones, también llamado “Runtime”, que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.
- Un conjunto de bibliotecas de funcionalidades y controles reutilizables, con una enorme cantidad de componentes ya programados listos para ser consumidos por otras aplicaciones.
- Un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores y linkers, que permitirán el desarrollo de aplicaciones sobre la plataforma .NET.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones
- Documentación y guías de arquitectura, que describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones .NET.

El .NET Framework (traducido como “Marco de Trabajo”) es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder *desarrollar aplicaciones* como para poder *ejecutarlas* luego en entornos de prueba o producción. Está compuesto por:

- El entorno de ejecución de la plataforma .NET
- Las bibliotecas de funcionalidad reutilizable

.NET Framework SDK: esta versión contiene herramientas de desarrollo de línea de comandos (compiladores, depuradores, etc.), documentación de referencia, ejemplos y manuales para desarrolladores de aplicaciones. Normalmente ésta variante se instala en los entornos de desarrollo de aplicaciones, y es más útil a los programadores que a los usuarios finales. Para poder instalar la versión SDK (Software Development Kit) es necesario instalar previamente el Redistributable Package.

.NET Compact Framework: esta es una versión reducida del .NET Framework Redistributable, especialmente pensada para ser instalada en dispositivos móviles como Pocket PC's y SmartPhones.

El objetivo de esta plataforma .Net es simplificar el desarrollo proporcionando todas las herramientas y tecnologías necesarias para desarrollar aplicaciones Web.

2.7.3 VISUAL ESTUDIO .NET 2005

Ofrece un entorno de desarrollo de alto nivel para desarrollar aplicaciones que se ejecutan sobre el .Net Framework 2.0, proporciona las tecnologías fundamentales para simplificar la creación, implantación y constante evolución de aplicaciones y Servicios Web seguros, escalables y de alta disponibilidad. También permite crear una nueva generación de aplicaciones basadas en Windows con numerosas y nuevas características disponibles en el .Net Framework.

2.7.4 ASP .NET

ASP .NET es un lenguaje de programación generado en el Common Language Runtime (CLR). ASP .NET es un componente especializado del entorno de trabajo de .NET, que permite aplicaciones Web. ASP.NET facilita el desarrollo de aplicaciones en comparación con el ASP clásico, por lo que la productividad de los programadores mejora considerablemente. Esta plataforma permite dotar de funciones adicionales a una aplicación Web y escribir una menor cantidad de código. Entre las ventajas que ofrece esta tecnología podemos mencionar las siguientes:

- **Mejor rendimiento.** ASP .NET es un código compilado que trabaja en el servidor. Además, ofrece la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento.
- **Compatibilidad de herramientas.** ASP .NET se integra con un nuevo diseñador y herramientas en un entorno integrado de programación y desarrollo Web, proporcionando la edición WYSIWYG para implementación rápida.
- **Eficacia y flexibilidad.** ASP .Net se basa en CLR, entonces la eficacia y la flexibilidad de toda esta plataforma está disponible para los desarrolladores de aplicaciones Web.

2.7.5 VISUAL BASIC .NET

Visual Basic .Net, es uno de los lenguajes que nos proporciona Microsoft por medio de su plataforma .Net Framework, para la construcción de componentes y servicios. Con esta herramienta se pueden construir aplicaciones ASP.NET, además ofrece numerosas características nuevas y mejoradas; como herencia, interfaces, sobrecarga de operadores y constructores, que lo convierten en un eficaz lenguaje de programación orientada a objetos. Se pueden construir aplicaciones multiproceso y escalables, Visual Basic .Net incluye el control estructurado de excepciones, atributos personalizados y compatibilidad con CLS (Common Language Specification) especificación del Lenguaje Común.

Con Visual Basic .Net se pueden desarrollar dos tipos de aplicaciones: aplicaciones para Windows (WinForms) y aplicaciones Web (WebForms). El acceso a datos es usando *ADO.NET* el cual nos permite trabajar en modo desconectado.

2.7.6 AJAX

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript y XML asíncronos), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- **XHTML** (o **HTML**) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña a la información.
- Document Object Model (**DOM**) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto **XMLHttpRequest** para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto **iframe** en lugar del **XMLHttpRequest** para realizar dichos intercambios.
- **XML** es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo **HTML** pre-formateado, texto plano, **JSON** y hasta **EBML**.

Como el **DHTML**, **LAMP** o **SPA**, **AJAX** no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente [WKA07].

CAPÍTULOS III - DESARROLLO DEL SOFTWARE

3.1 SITUACIÓN ACTUAL

Una vez analizada la información obtenida, se ha establecido que el proceso de gestión académica en la carrera de Ciencias de la Educación se realiza en gran parte de forma manual, y no existen mecanismos específicos que apoyen en este proceso.

3.2 MODELO DE REQUISITOS

3.2.1 MODELO DEL DOMINIO DEL PROBLEMA

El modelo del dominio del problema define un *modelo de clases* común para todos los involucrados en el modelo de requisitos. Este modelo de clases consiste en los objetos del dominio del problema, o sea objetos que tienen una correspondencia directa en el área de la aplicación. El propósito principal del dominio del problema en el modelo de requisitos de esta metodología es formar una base común de entendimiento del desarrollo y no definir el sistema completo.

a) Selección de Clases

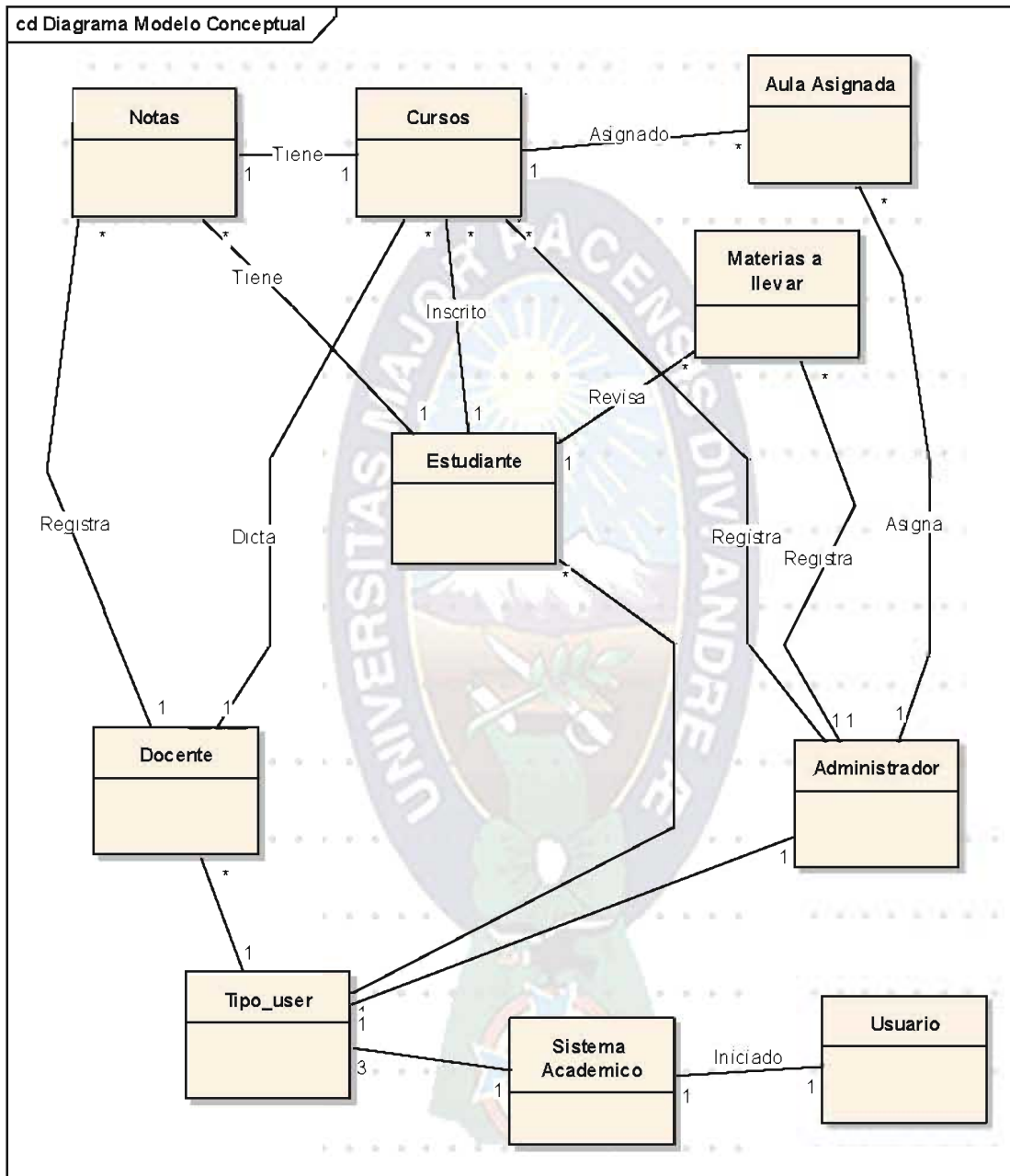
Tabla 3.1: Clases Identificadas

Clases identificadas	
Usuario	Cursos
Estudiante	Aula asignada
Docente	Materias a llevar
Administrador	Notas

Fuente: Elaboración Propia

b) Diagrama de Clases

Figura 3.1: Diagrama de clases con asociaciones y roles entre clases identificadas



Fuente: Elaboración Propia

c) **Diccionario de Clases**

El diccionario de clases o diccionario de datos, describe textualmente las clases identificadas durante el modelo del dominio del problema. Este diccionario sirve como un glosario de términos y se muestra a continuación.

✓ *Usuario:* Para poder utilizar el Sistema Académico se necesita un login y password asignado.

✓ *Tipo User:* Define el rol del usuario para poder utilizar el Sistema Académico. Los roles pueden ser tres Administrador, Docente y Estudiante.

✓ *Administrador:* El Administrador puede realizar tareas como el control y el registro de Docentes y Estudiantes, registra y habilita materias por semestre, asigna prerrequisitos, define fechas de inscripción, e inscribe alumnos directamente, genera números aleatorios, tiene acceso a reportes de estadísticas académicas.

✓ *Docente:* Puede realizar distintas tareas, como son la registro de notas, asignación de ponderación de los exámenes, ver listado de alumnos de su materia.

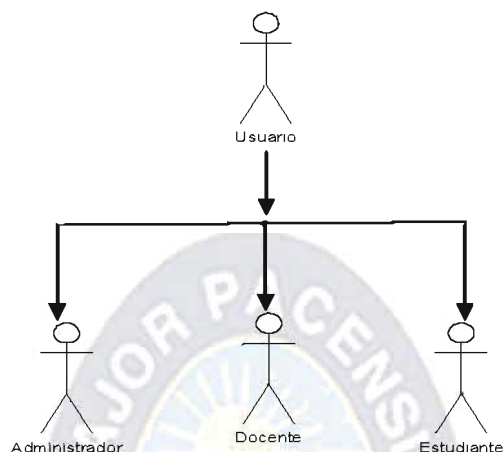
✓ *Estudiante:* Realiza distintas tareas como la consulta de record e historial académico, inscripción a materias según prerrequisitos, consulta de notas parciales.

3.2.2 **MODELO DE NEGOCIO**

ACTORES

En la terminología orientada objetos, se considera al actor una clase de usuario, mientras que los usuarios se consideran objetos o instancias de esa clase. Los actores modelan cualquier entidad externa que necesite intercambiar información con el sistema. Antes de identificar los casos de uso, se identifican los actores del sistema, esto es para que estos sean la herramienta principal que permita encontrar los casos de uso. Retomando la descripción del Sistema Académico, se puede identificar a varios tipos de usuarios como el Administrador, Docente y Estudiante, quienes están encargados de hacer consultas y otras operaciones en el Sistema Académico. También se puede identificar que la Base de Datos del sistema, encargado de proveer la información de los usuarios.

Figura 3.2: Actores identificados para el Sistema Académico



Fuente: Elaboración Propia

Tabla 3.2: Descripción Actor Administrador

Actor	Administrador
Casos de Uso	Validar Usuario, Registro de Usuario, Registro Materias y registro prerequisites, control de inscripciones
Tipo	Primario
Descripción	Es el actor principal el cual administra los usuarios del Sistema Académico.

Fuente: Elaboración Propia

Tabla 3.3: Descripción Actor Docente

Actor	Docente
Casos de Uso	Validar Usuario, Control y registro de Notas totales y parciales, listados de alumnos, llenado de actas de notas.
Tipo	Primario
Descripción	Es un actor principal, el cual de juega el papel de administrador de las materias que tiene asignadas.

Fuente: Elaboración Propia

Tabla 3.4: Descripción Actor Estudiante

<i>Actor</i>	<i>Estudiante</i>
<i>Casos de Uso</i>	Validar Usuario, Consulta historial académico, consulta de notas, inscripciones.
<i>Tipo</i>	Primario
<i>Descripción</i>	Es un actor principal, el cual realiza consultas permanentes al sistema.

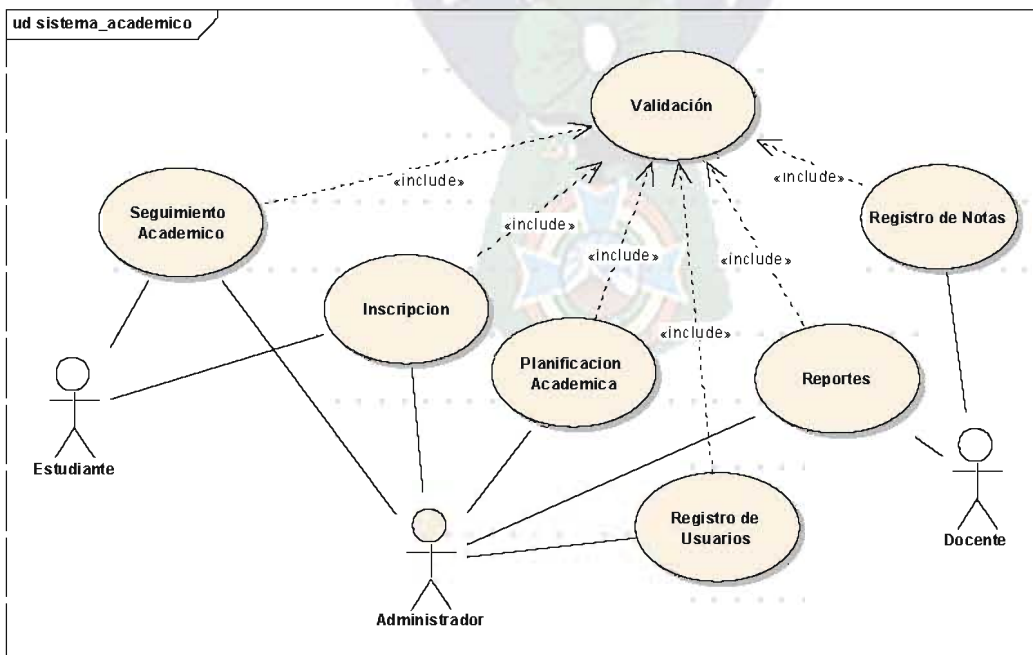
Fuente: Elaboración Propia

3.2.3 CASOS DE USO

Después de haber identificado a los actores del Sistema Académico, se establece la funcionalidad propia del mismo por medio de los casos de uso. Cada caso de uso se define una clase o forma particular de usar el Sistema Académico, mientras que cada ejecución del caso de uso se puede ver como una instancia del mismo. Cada caso de uso constituye un flujo completo de eventos, que especifican la interacción que toma lugar entre el actor y el Sistema Académico.

- a) Caso de Uso de Alto Nivel

Figura 3.3: Diagrama de Casos de uso de Alto nivel



b) Descripción Casos de Uso Extendido

i) Caso de uso: Validar Usuario

Tabla 3.5: Caso de uso Validar usuario

Caso de uso	Validar Usuario
Actores	Administrador, Docente, Estudiante
Tipo	Inclusión
Propósito	Validar a los usuarios ya registrados para el uso del Sistema Académico para participar en sus respectivos Cursos.
Resumen	Este caso de uso se inicia tanto por el administrador, docente y estudiante. Se valida mediante el login y password a verificarse por su respectivo registro de usuario, para que pueda utilizar el Sistema Académico con el rol asignado (Administrador, Docente, Estudiante).
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso de registro de usuarios.
Flujo Principal	Se presenta al usuario en la pantalla principal. Una vez validado el usuario presenta el menú de opciones según el tipo de rol de usuario.
Subflujos	Ninguno
Excepciones	E-1 No hubo validación login/password de usuario correctamente. No puede ingresar al Sistema Académico.

Fuente: Elaboración Propia

ii) Caso de uso: Registro de Usuario

Tabla 3.6: Caso de uso Registrar usuario

Caso de uso	Registro de usuario
Actores	Administrador
Tipo	Primario
Propósito	Registrar a los usuarios y proporcionarles login/password para el Sistema Académico.
Resumen	El Administrador registrara la inscripción asignaturas y paralelos de los estudiantes y la asignación de asignaturas para docentes, además de proporcionar login/password a estos, con la información del Sistema Académico de la Carrera.
Precondiciones	Haber pasado por el periodo de inscripción del semestre.
Flujo Principal	Se ejecuta el registro de estudiantes en base a la inscripción realizada para el semestre y se les asigna login/password a los usuarios Docente y Estudiante.
Subflujos	Ninguno
Excepciones	Ninguno

Fuente: Elaboración Propia

iii) **Caso de uso: Planificación Académica**

Tabla 3.7: Caso de uso Planificación Académica

Caso de uso	Planificación Académica
Actores	Administrador
Tipo	Primario
Propósito	Registrar la información necesaria, para el inicio de la nueva Gestión.
Resumen	El Administrador registrara las materias y paralelos habilitados, las aulas y horarios asignados, los Docentes que dictaran cátedra, Generación de Materias a llevar y números aleatorios, para realizar la inscripción de los alumnos.
Precondiciones	Haber ejecutado el caso de uso validar usuario.
Flujo Principal	Se ejecuta el caso de uso Planificación Académica, dependiendo de la opción seleccionada el Administrador ejecutara los subflujos.
Subflujos	<p><i>S1: Registro de Materias y Paralelos.-</i> Se ejecuta registro de materias y paralelos habilitados.</p> <p><i>S2: Asignación de Aulas y horarios.-</i> Se ejecuta Asignación de Aulas y Horarios para adicionar o modificar una asignación de aula ya existente, para luego guardar y generar el horario correspondiente.</p> <p><i>S3: Asignación de docentes.</i> Se escoge la materia y el paralelo, luego se busca que docente dictara esa asignatura.</p> <p><i>S4: Generar Materias a Llevar y Números Aleatorios.-</i> Se ejecuta Generar Inscripción, luego el sistema de manera interna generara el listado de materias a llevar para cada estudiante, en base a la tabla de prerequisites; también asignara números aleatorios para ver en qué fecha y hora podrán inscribirse los alumnos, en base al promedio obtenido en el anterior semestre.</p>
Excepciones	Ninguno

Fuente: Elaboración Propia

iv) Caso de uso: Reportes

Tabla 3.8: Caso de uso Reportes

Caso de uso	Reportes
Actores	Administrador, Docente
Tipo	Primario
Propósito	Obtener reportes de la información almacenada en la BD.
Resumen	El Administrador y el Docente, podrán sacar reportes de lista de alumnos inscritos por materia y gestión, índices de aprobación y abandono de los estudiantes, actas de notas, cantidad de egresados por Gestión.
Precondiciones	Haber ejecutado el caso de uso validar usuario.
Flujo Principal	Se ejecuta el caso de uso reportes, dependiendo de la opción seleccionada el Administrador ejecutara los subflujos.
Subflujos	S1: <i>Listados de Alumnos y Actas de Notas.</i> - Se ejecuta reportes, introduciendo los parámetros de gestión y código de materia. S2: <i>Índices.</i> - Se ejecuta reportes de índices, introduciendo los parámetros de gestión y periodo.
Excepciones	Ninguno

Fuente: Elaboración Propia

v) Caso de uso: Inscripciones

Tabla 3.9: Caso de uso inscripciones

Caso de uso	Inscripciones
Actores	Administrador, Estudiante
Tipo	Primario
Propósito	Realizar el registro respectivo de materias a cursar para el alumno.
Resumen	El estudiante ingresa al sistema en la fecha y hora asignada, y registra las materias habilitadas que desea tomar, seleccionando el paralelo deseado y finaliza su inscripción. El director de carrera recibe la solicitud de inscripción, retiro y/o adición del estudiante, realiza la inscripción, retiro y/o adición de materias; también tiene la posibilidad de habilitar una o varias materias para inscribir al estudiante, rompiendo prerrequisitos.
Precondiciones	Haber ejecutado el caso de uso validar usuario, el estudiante debe estar dentro de la fecha y hora asignada para su inscripción.
Flujo Principal	Se ejecuta el caso de uso inscripciones, dependiendo de la opción seleccionada el Administrador o el Estudiante, se ejecutarán los subflujos.
Subflujos	S1: <i>Inscripción Estudiante.</i> - Se ejecuta inscripción, registra las materias y los paralelos. S2: <i>Inscripción Administrador.</i> - Se ejecuta inscripción, busca al estudiante, adiciona o elimina las materias requeridas, pudiendo romper prerrequisitos.
Excepciones	Ninguno

Fuente: Elaboración Propia

i) **Caso de uso: Seguimiento Académico**

Tabla 3.10: Caso de uso Seguimiento Académico

Caso de uso	Seguimiento Académico
Actores	Administrador, Estudiante
Tipo	Primario
Propósito	Obtener reportes de record y/o historial académico.
Resumen	El Administrador, podrá responder a la solicitud del estudiante emitiendo el record y/o historial académico. El Estudiante, podrá ver directamente su record y/o historial académico, mediante el sistema.
Precondiciones	Haber ejecutado el caso de uso validar usuario.
Flujo Principal	Se ejecuta el caso de uso Seguimiento Académico, dependiendo de la opción seleccionada el Administrador o Estudiante, se ejecutaran los subflujos.
Subflujos	S1: <i>Record Académico</i> .- Se ejecuta seguimiento académico – record académico, e inmediatamente se despliega en pantalla la información del record académico del estudiante seleccionado. S2: <i>Historial Académico</i> .- Se ejecuta seguimiento académico – historial académico, e inmediatamente se despliega en pantalla la información del historial académico del estudiante seleccionado.
Excepciones	Ninguno

Fuente: Elaboración Propia

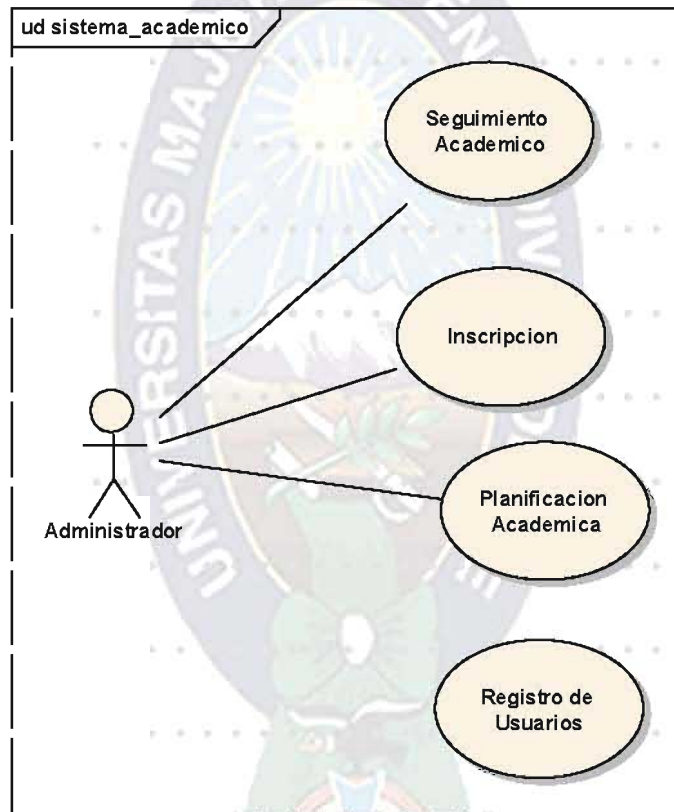
a) Diagramas de Casos uso

Del análisis del sitio actual y de la descripción realizada en los casos de uso extendidos, se obtienen los diferentes diagramas de casos de uso, que muestran el comportamiento del sistema, de acuerdo a los perfiles de usuarios.

i) Diagrama de Caso de uso administrador

Para el caso del usuario administrador, la figura muestra su correspondiente caso de uso:

Figura 3.4: Diagrama de caso de uso del administrador del sitio

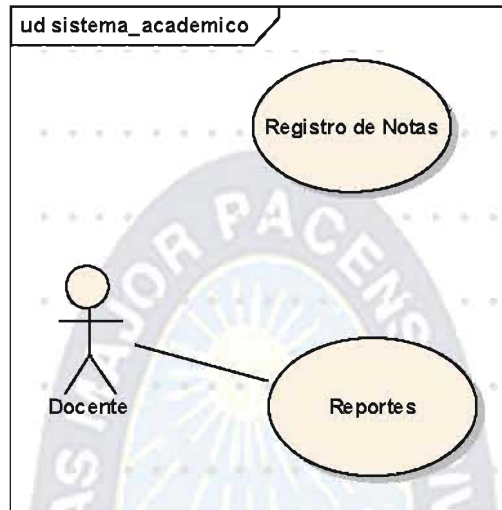


Fuente: Elaboración Propia

ii) Diagrama de Caso de uso docente

Para el caso del usuario docente, la figura muestra su correspondiente caso de uso:

Figura 3.5: Diagrama de caso de uso del docente

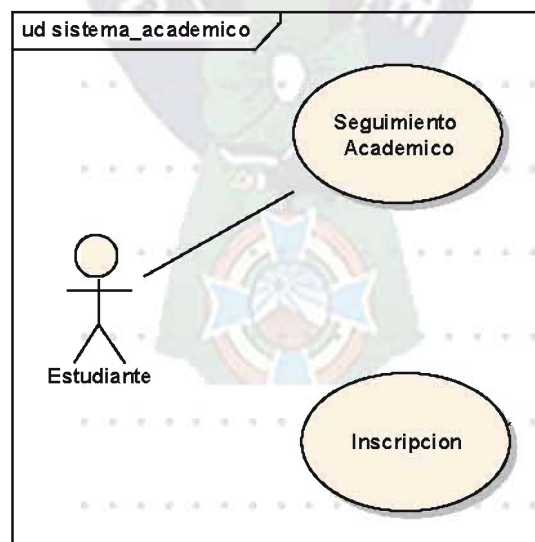


Fuente: Elaboración Propia

iii) Diagrama de Caso de uso estudiante

Para el caso del usuario estudiante, la figura muestra su correspondiente caso de uso:

Figura 3.6: Diagrama de caso de uso del estudiante



Fuente: Elaboración Propia

3.3 MODELO DE ANÁLISIS

3.3.1 ARQUITECTURA DE CLASES

El modelo de análisis tiene como objetivo generar una arquitectura de objetos que sirva como base para el diseño del sistema, dependiendo del tipo de aplicación existen diversas arquitecturas que se pueden utilizar.

Las arquitecturas se distinguen según la organización de los objetos de acuerdo a su funcionalidad. Esto es también conocido como la dimensión de la arquitectura. Por ejemplo, si existe un grupo de objetos para el manejo de la funcionalidad de la aplicación y otro para interactuar con las entidades externas de la aplicación, como el usuario y las bases de datos, entonces se considera que la arquitectura es de dos dimensiones. En general, una arquitectura puede incluir cualquier número de dimensiones, algo que depende del tipo de aplicación que se desee desarrollar.

En el caso de los sistemas de información una de las arquitecturas más utilizadas es la Modelo, Vista, Control (MVC Model, View, Control). Esta arquitectura se basa en tres dimensiones principales: *Modelo* correspondiente a la información, *Vista* correspondiente a la presentación o la interacción con el usuario y *Control* correspondiente al comportamiento.

a) Clases con estereotipos

El tipo de funcionalidad o la razón de ser de un objeto dentro de una arquitectura se conoce como su estereotipo. Siguiendo con la metodología de casos de uso, la arquitectura del sistema para el modelo de análisis se basará en tres estereotipos:

- El estereotipo entidad, para los objetos que guardan información sobre el estado interno del sistema. Estos objetos corresponden al dominio del problema.
- El estereotipo borde, para objetos que implementan las interfaces del sistema con el mundo externo, correspondientes a todos los actores incluyendo a aquellos que no son humanos.
- El estereotipo control, para objetos que implementan el comportamiento a control de la lógica de los casos de uso, especificando cuando y como el sistema cambia de estado. Los objetos control modelan la funcionalidad que no se asocia naturalmente con un solo objeto.

b) Clases para los Casos de uso

En cada caso de uso se identifican los objetos necesarios para su implementación. Los objetos se identifican según sus estereotipos de manera que correspondan con la funcionalidad ofrecida en cada uno.

La asignación de objetos a cada caso de uso se hace de acuerdo con los siguientes principios:

- La funcionalidad de los casos de uso que depende directamente de la interacción del sistema con el mundo externo se asigna a los objetos borde.
- La funcionalidad relacionada con el almacenamiento y manejo de información del dominio del problema se asigna a los objetos entidad.
- La funcionalidad específica a uno o varios casos de uso y que afecta a múltiples objetos a la vez, o que se relaciona naturalmente con ningún objeto borde o entidad, se asigna a los objetos control.

3.3.2 IDENTIFICACIÓN DE CLASES SEGÚN ESTEREOTIPOS

Para llevar a cabo la transición del modelo de requisitos al modelo de análisis se deben identificar los objetos necesarios para implementar todos los casos de uso.

a) CLASES SEGÚN CASOS DE USO

ii) Caso de uso: Validar usuario

El caso de uso Valida Usuario involucra como clases control Manejador Registro Usuario que controla la información de Registro Usuario, el Manejador de Registro de Asignaturas el cual administra el registro de asignaturas que tienen asignadas los Docentes y Estudiantes para el semestre; y Manejador Principal para administrar los aspectos generales del Sistema incluyendo la pantalla principal. Como clase borde tenemos a la Interfase Usuario.

iii) Caso de uso: Registrar usuario

El caso de uso Registrar Usuario involucra una clase control Manejador Registro Usuario que controla información de Registro Usuario, y como clases borde a la Pantalla Registro Usuarios, Pantalla Usuarios Registrados y la Pantalla de Información de Inscripción a asignaturas del semestre.

iv) Caso de uso: Seguimiento Académico

El caso de uso Seguimiento Académico involucra una clase control Seguimiento Académico que controla la información de Materias Aprobadas del alumno, como clases borde a la Pantalla Reporte Record Académico y al Historial Académico.

v) Caso de uso: Inscripción

El caso de uso Inscripción involucra una clase control inscripción, que controla la información de validación de fecha y hora de inscripción, como clases borde a la Pantalla Inscripción Estudiante, Pantalla de inscripción Administrador.

vi) Caso de uso: Planificación Académica

El caso de uso Planificación Académica involucra una clase control Planificación Académica que controla toda información de la Gestión Vigente, como clase borde a las pantallas de Registro de Materias y Paralelos, la de Asignación de Aulas y horarios, Asignación de Docentes, Generar Materias a llevar y Números Aleatorios.

vii) Caso de uso: Reportes

El caso de uso Reportes involucra una clase control Reportes que controla información estadística de los estudiantes, en todas las gestiones y periodos, como clase borde a las pantallas de Reporte de listados de notas, la de índices de aprobación y abandono.

3.4 MODELO DE DISEÑO

El modelo de diseño es un refinamiento y formalización adicional del análisis, donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado de este son especificaciones muy detalladas de todos los objetos, incluyendo sus operaciones y atributos. Este modelo se obtendrá en base a la metodología de OOHDM, que es muy utilizado para el desarrollo de aplicaciones vía Web.

3.4.1 DISEÑO DEL MODELO CONCEPTUAL

El modelo conceptual se utiliza para definir los objetos del dominio y las relaciones entre ellos, utilizando la notación de UML, donde estos objetos son representados mediante un diagrama de clases. La figura muestra el modelo conceptual de la aplicación.

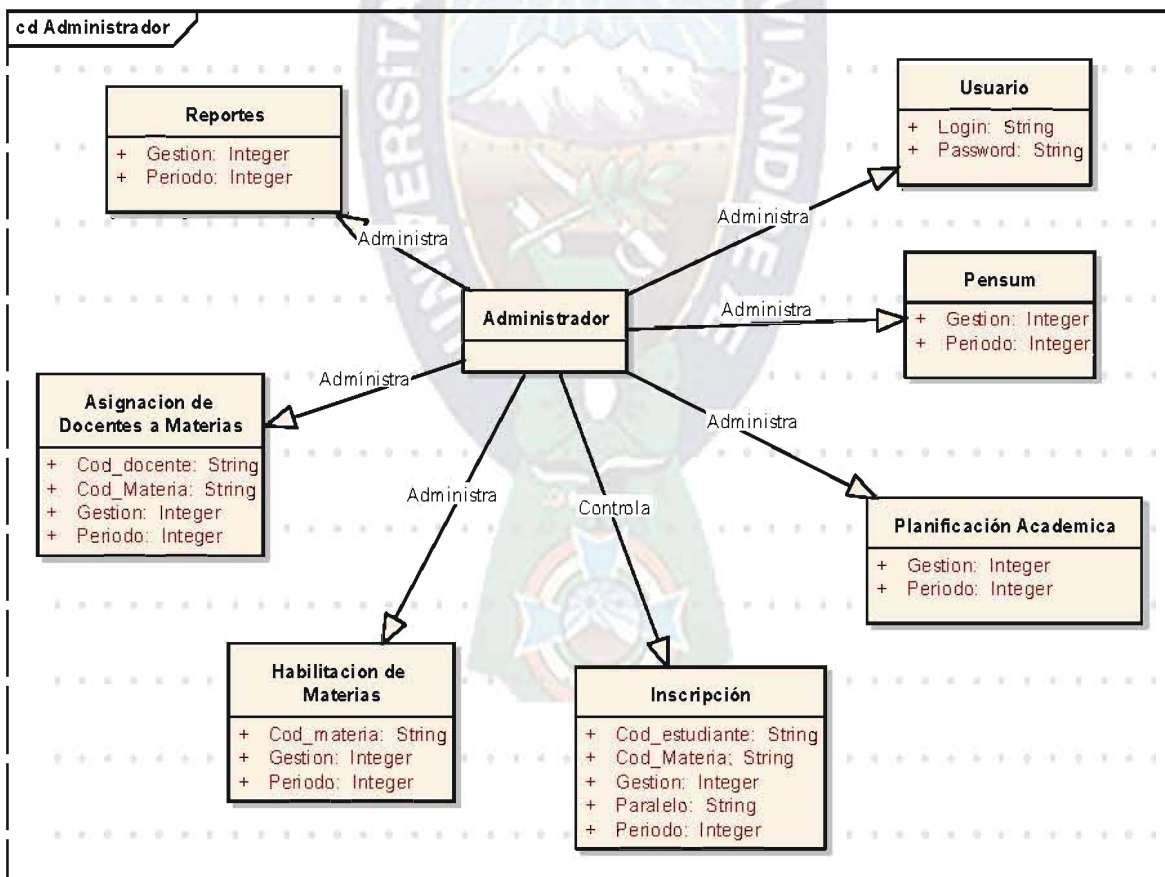
3.4.2 DISEÑO NAVEGACIONAL

El diseño navegacional nos permite elaborar vistas de la aplicación, representados por nodos, enlaces y estructuras de acceso. El diseño de estas vistas, son de acuerdo a los perfiles de los usuarios identificados.

Esquema de clases navegacionales

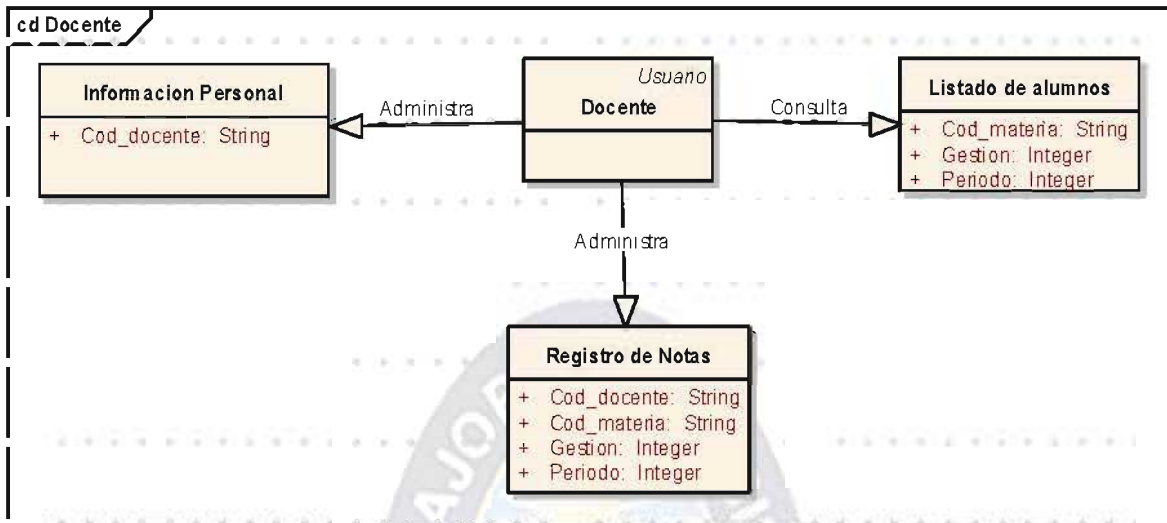
Un esquema de clases navegacionales, está compuesto por nodos (objetos obtenidos del modelo conceptual) y por enlaces (líneas que representan los enlaces entre los nodos). A continuación se presenta los esquemas de clases navegacionales, tomando en cuenta los diferentes perfiles de usuario identificados.

Figura 3.8: Diagrama navegacional para el usuario Administrador



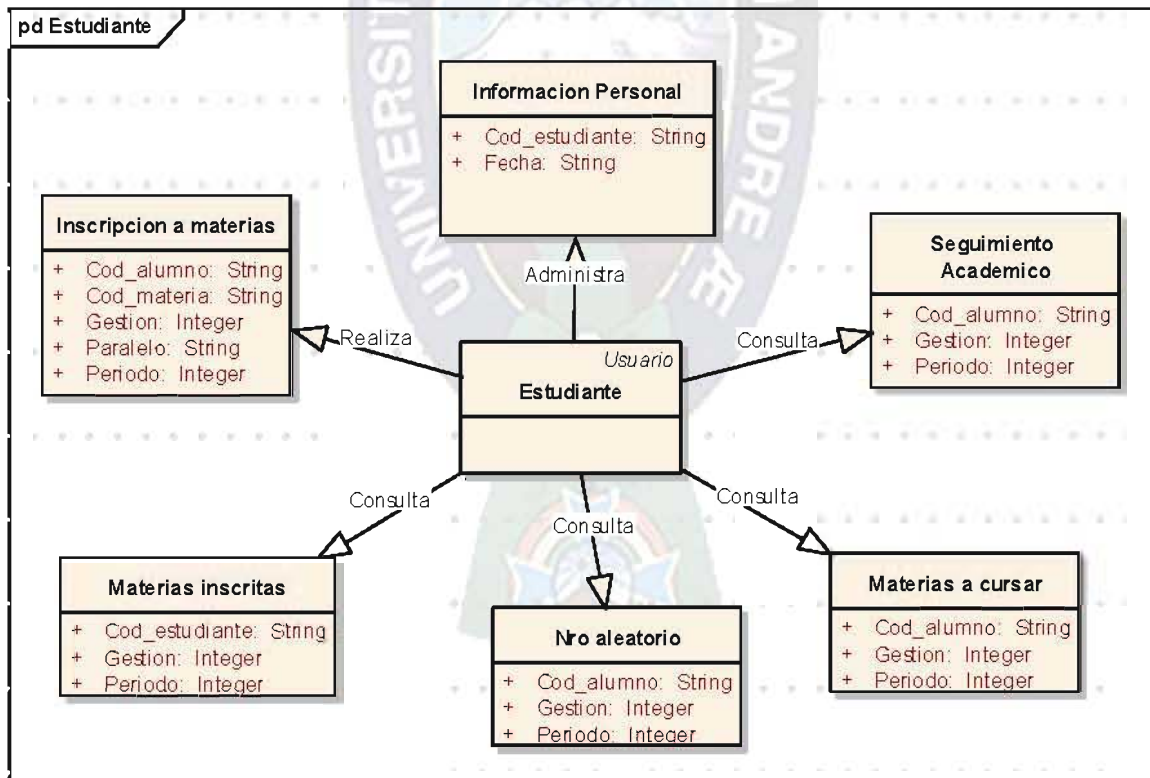
Fuente: Elaboración propia

Figura 3.9: Diagrama navegacional para el usuario Docente



Fuente: Elaboración propia

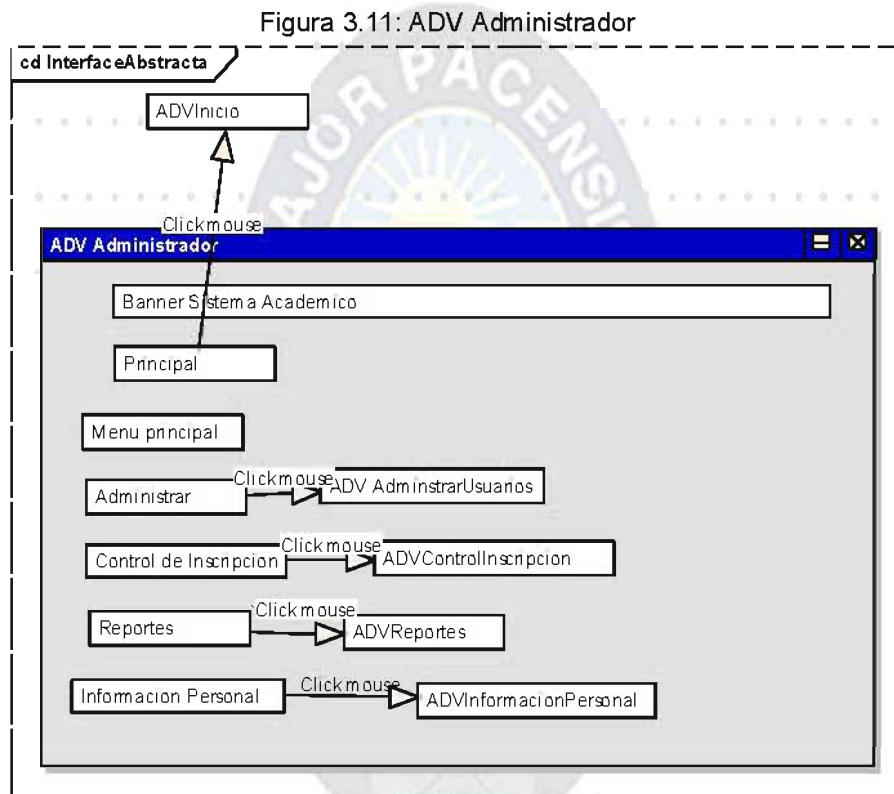
Figura 3.10: Diagrama navegacional para el usuario Estudiante



Fuente: Elaboración propia

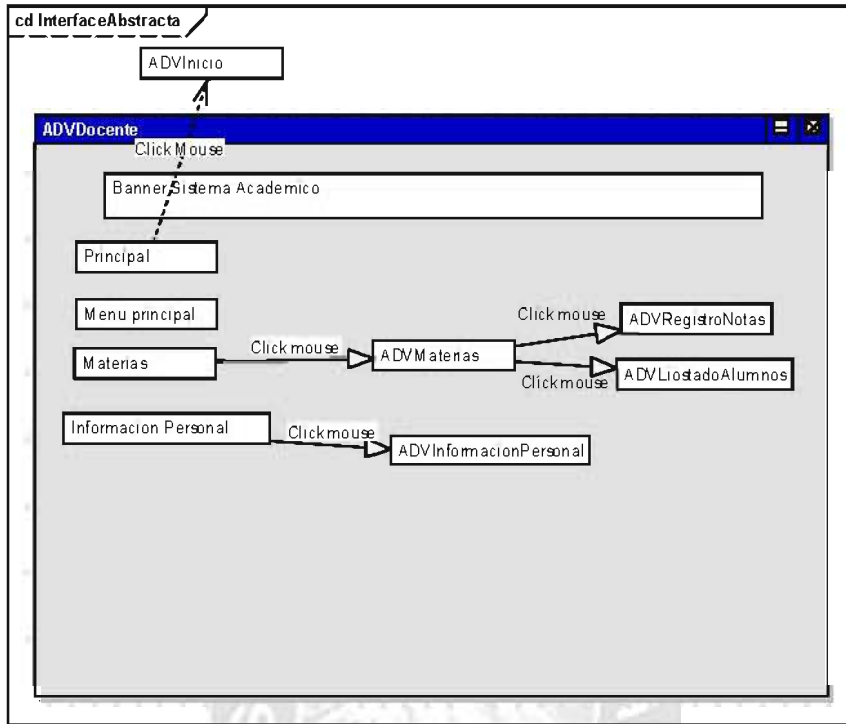
3.4.3 DISEÑO INTERFAZ ABSTRACTA

Después de haber definido la estructura de navegación de la aplicación, se debe proceder a especificar las interfaces de usuario. El diseño de una interfaz abstracta, nos ayuda a decidir que objetos serán mostrados. Para la definición los objetos de interfaz se utilizan enlaces, imágenes, etc.



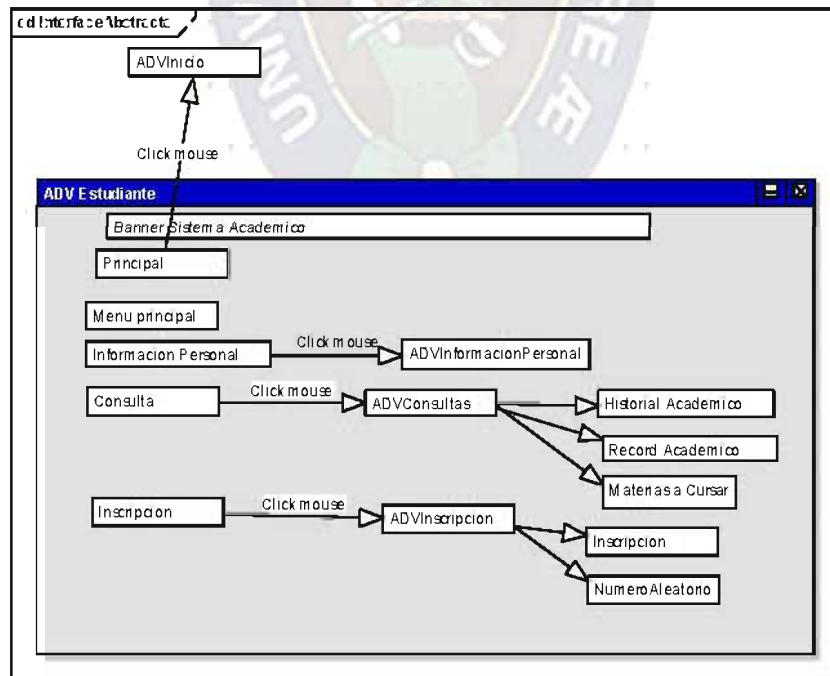
Fuente: Elaboración propia

Figura 3.12: ADV Docente



Fuente: Elaboración propia

Figura 3.13: ADV Docente



Fuente: Elaboración propia

CAPÍTULO IV – IMPLEMENTACIÓN Y PRUEBAS

4.1 MODELO IMPLEMENTACIÓN

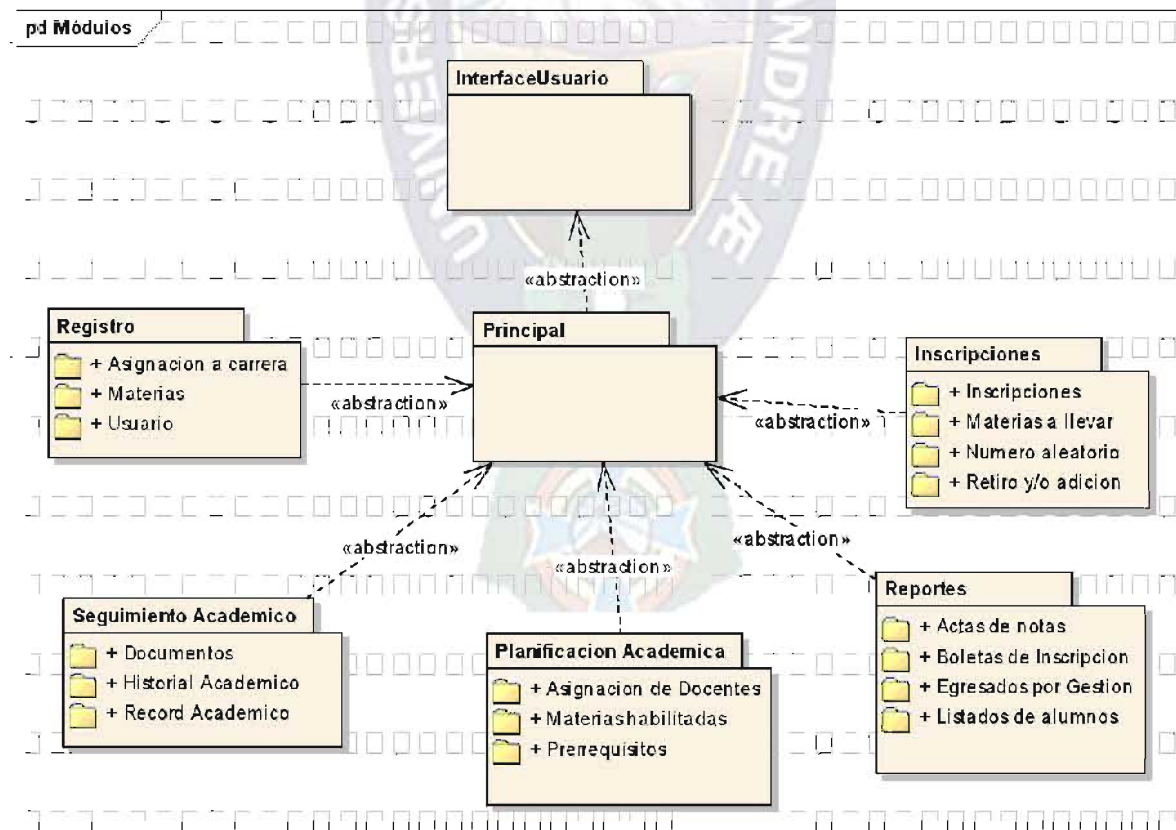
En el presente capítulo se muestra la implementación y desarrollo del Sistema de Gestión Académica vía Web para la carrera de Ciencias de la Educación de la UMSA.

El modelo de implementación toma el resultado del modelo de diseño para generar el código final. El modelo de implementación se adapta al lenguaje de programación y/o base de datos, según la especificación del diseño y las propiedades del lenguaje de implementación.

4.2 REPRESENTACIÓN DE LOS MÓDULOS DEL SISTEMA COMPLETO

Este diagrama sirve como apoyo visual para el proceso de programación.

Figura 4.1: Diagrama de módulos del Sistema Académico



Fuente: Elaboración propia

4.3 REQUERIMIENTOS MÍNIMOS

Los requerimientos mínimos que tiene que poseer el ordenador del cliente, en cuanto a hardware y software, son los siguientes:

- Un procesador Pentium 3 o superior.
- Un mínimo de 128 MB de Memoria RAM.
- Un navegador con soporte para javascript, para la utilización de AJAX.
- Lector de CD.
- Capacidad de almacenamiento de 300 MB.

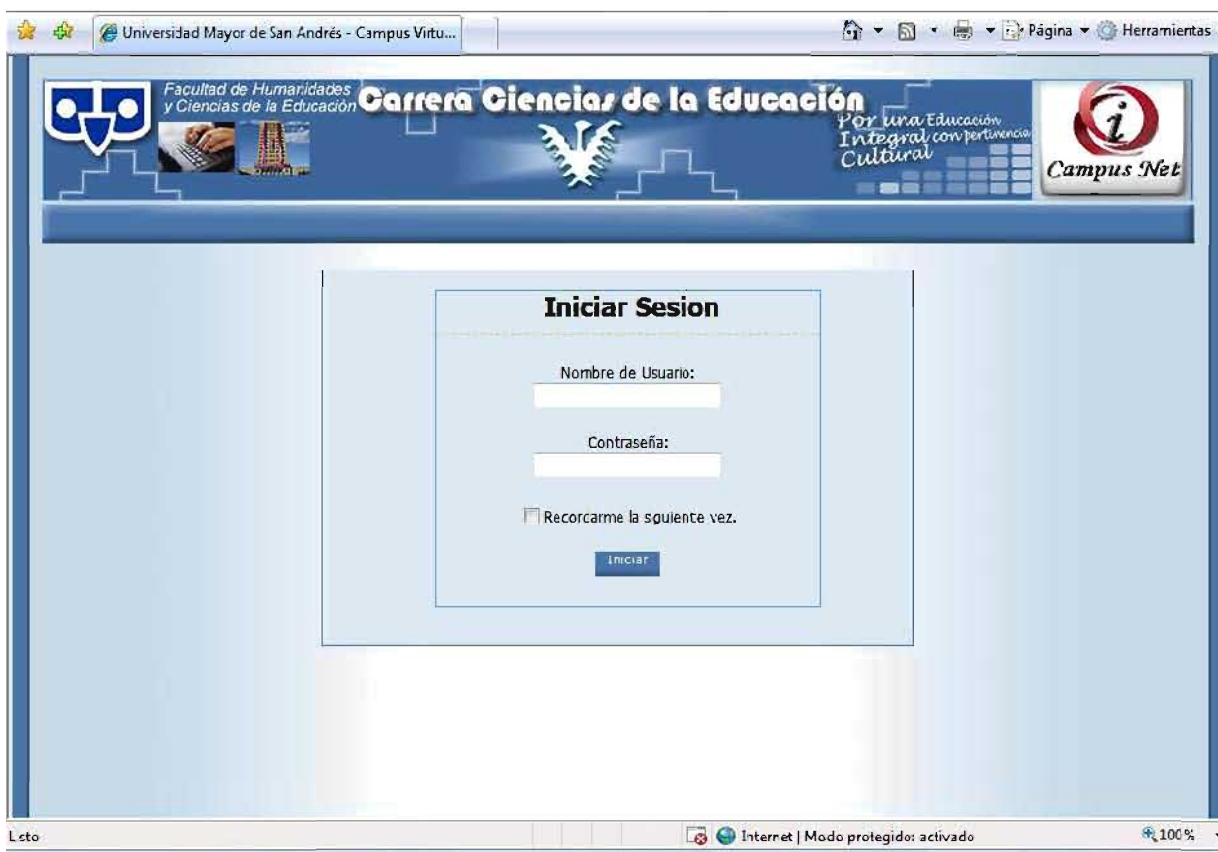
Los requerimientos mínimos que tiene que poseer el servidor, en cuanto a hardware y software, son los siguientes:

- Un procesador Pentium 4 o superior.
- Un mínimo de 1 Gb de Memoria RAM.
- Microsoft Net Framework 2.0
- Ajax Extension y Control Toolkit
- Un motor de Base de Datos SQLSERVER 2000.
- Un servidor de Páginas Web IIS (Internet Information Server)
- Lector de DVD.
- Capacidad de almacenamiento de 20 Gb.

4.4 INTERFAZ DE USUARIO

En la Fig. se observa el Portal de la Carrera de Ciencias de la Educación. A partir de este portal los visitantes tienen la posibilidad de navegar hacia diferentes lugares; Entre estas la de visitar el plan curricular y demás información pertinente.

Figura 4.2 , Autenticación del Sistema de Información



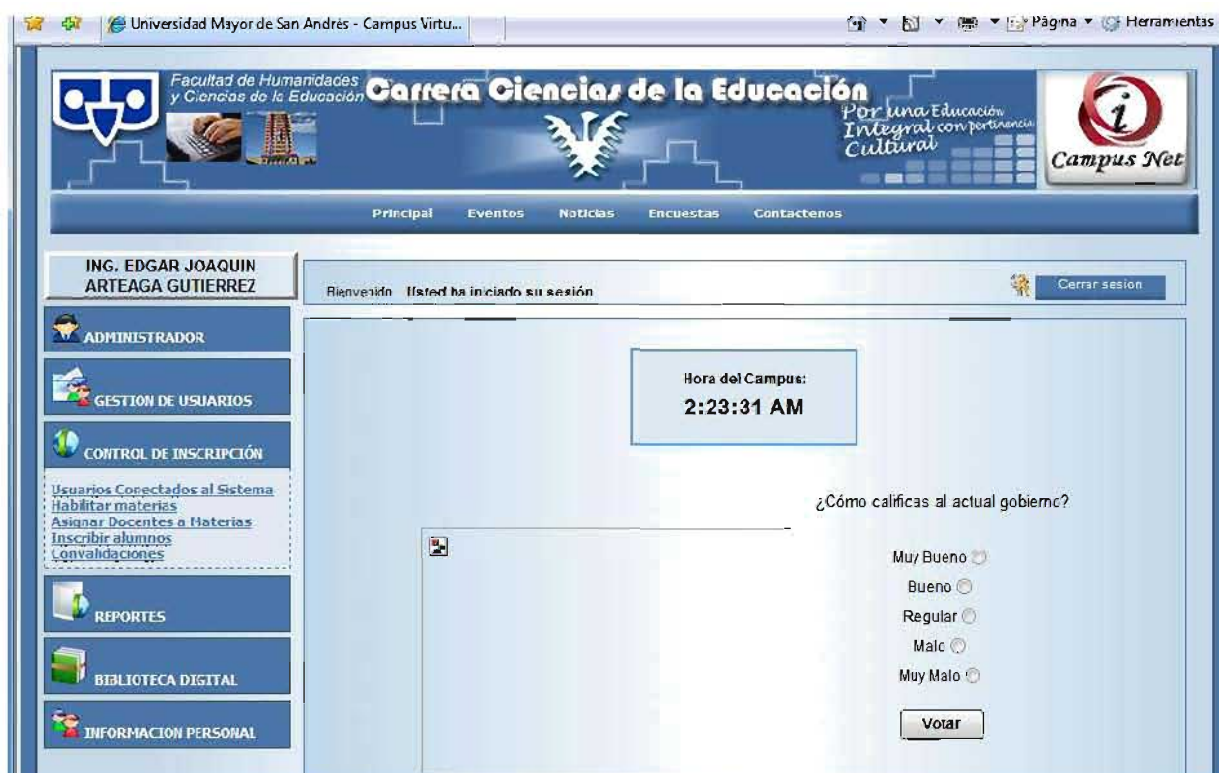
Fuente: Elaboración propia

Una vez dentro del sistema de información el visitante y ser reconocido como un usuario del sistema, tiene un conjunto de funciones y transacciones asignadas según su rol.

Cada vez que un usuario, cualquiera sea, ejecute alguna función o transacción en el sistema, el modulo de seguridad revisa los permisos que dicho usuario tenga, para permitir o negar la transacción.

Los usuarios reconocidos por el sistema de información son Estudiantes, Docente y Administrador.

Figura 4.3, Funciones y transacciones del rol Administrador



Fuente: Elaboración propia

Si las asignaturas que constituirán el plan curricular no existen en la Base de Datos, tienen que ser creadas una a una.

Figura 4.4 Administración de materias por carrera

ADMINISTRACIÓN DE MATERIAS POR CARRERA

[Volver Carrera<<](#)

CARRERA SELECCIONADA:

<u>SIGLA</u>	<u>DESCRIPCIÓN</u>	<u>SEM.</u>	<u>CRÉDITOS</u>	<u>HOR. TEOR.</u>	<u>HOR. PRACT.</u>	<u>TOTAL HORAS</u>		
NO CORREG	MATERIA NO CORREGIDA	0					Editar	Eliminar
NO VALIDO	MATERIA NO CONVALIDADA	0					Editar	Eliminar
CCE-110	INVESTIGACION EDUCATIVA I	1					Editar	Eliminar
CCE-120	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION I	1					Editar	Eliminar
CCE-150	FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION	1					Editar	Eliminar
CSO-107	SOCIOLOGIA GENERAL	1					Editar	Eliminar
HIS-134	HISTORIA DE LA EDUCACION BOLIVIANA	1					Editar	Eliminar
TL-100	TALLER DE LENGUAJE I	1					Editar	Eliminar
CCE-111	INVESTIGACION EDUCATIVA II	2					Editar	Eliminar
CCE-121	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION II						Editar	Eliminar
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Adicionar materia	

1 2 3 4 5 6 7 8 9 10

Fuente: Elaboración propia

Crear el Pensum, y luego, agregar una a una las asignaturas del plan curricular y sus prerrequisitos.

Figura 4.5 Llenado de prerequisites

LLENADO DE PREREQUISITOS

CARRERA: Ciencias de la Educación
NIVEL: Ciencias de la Educación
N° RESOLUCIÓN:
FECHA CREACIÓN:

LISTA DE MATERIAS

- 1 - CCE-110 - INVESTIGACION EDUCATIVA I
- 1 - CCE-120 - INTRODUCCION A LAS CIENCIAS DE LA EDUCACION
- 1 - CCE-150 - FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION
- 1 - CSO-107 - SOCIOLOGIA GENERAL
- 1 - HIS-134 - HISTORIA DE LA EDUCACION BOLIVIANA
- 1 - TL-100 - TALLER DE LENGUAJE I
- 2 - CCE-111 - INVESTIGACION EDUCATIVA II
- 2 - CCE-121 - INTRODUCCION A LAS CIENCIAS DE LA EDUCACION
- 2 - CSO-132 - SOCIOLOGIA DE LA EDUCACION
- 2 - HIS-135 - HISTORIA DE LA EDUCACION LATINOAMERICANA
- 2 - PSI-100 - FUNDAMENTOS PSICOLOGICOS DE LA EDUCACION
- 2 - TL-101 - TALLER DE LENGUAJE II
- 3 - CCE-100 - ESTADISTICA I
- 3 - CCE-112 - INVESTIGACION EDUCATIVA III
- 3 - CCE-130 - PEDAGOGIA GENERAL I
- 3 - CCE-170 - ANTROPOLOGIA Y EDUCACION I
- 3 - FIL-218 - INTRODUCCION A LA FILOSOFIA
- 3 - PSI-203 - PSICOLOGIA EVOLUTIVA I
- 4 - CCE-101 - ESTADISTICA II
- 4 - CCE-113 - INVESTIGACION EDUCATIVA IV
- 4 - CCE-131 - PEDAGOGIA GENERAL II
- 4 - CCE-171 - ANTROPOLOGIA Y EDUCACION II
- 4 - FIL-219 - FILOSOFIA DE LA EDUCACION
- 4 - PSI-204 - PSICOLOGIA EVOLUTIVA II

LISTA DE MATERIAS PREREQUISITOS

- 1 - CCE-120 - INTRODUCCION A LAS CIENCIAS DE LA EDUCACION
- 1 - CCE-150 - FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION
- 1 - CSO-107 - SOCIOLOGIA GENERAL
- 1 - HIS-134 - HISTORIA DE LA EDUCACION BOLIVIANA
- 1 - TL-100 - TALLER DE LENGUAJE I
- 2 - CCE-111 - INVESTIGACION EDUCATIVA II
- 2 - CCE-121 - INTRODUCCION A LAS CIENCIAS DE LA EDUCACION
- 2 - CSO-132 - SOCIOLOGIA DE LA EDUCACION
- 2 - HIS-135 - HISTORIA DE LA EDUCACION LATINOAMERICANA
- 2 - PSI-100 - FUNDAMENTOS PSICOLOGICOS DE LA EDUCACION
- 2 - TL-101 - TALLER DE LENGUAJE II
- 3 - CCE-100 - ESTADISTICA I
- 3 - CCE-112 - INVESTIGACION EDUCATIVA III
- 3 - CCE-130 - PEDAGOGIA GENERAL I
- 3 - CCE-170 - ANTROPOLOGIA Y EDUCACION I
- 3 - FIL-218 - INTRODUCCION A LA FILOSOFIA
- 3 - PSI-203 - PSICOLOGIA EVOLUTIVA I
- 4 - CCE-101 - ESTADISTICA II
- 4 - CCE-113 - INVESTIGACION EDUCATIVA IV
- 4 - CCE-131 - PEDAGOGIA GENERAL II
- 4 - CCE-171 - ANTROPOLOGIA Y EDUCACION II
- 4 - FIL-219 - FILOSOFIA DE LA EDUCACION
- 4 - PSI-204 - PSICOLOGIA EVOLUTIVA II

LISTA DE MATERIAS Y PREREQUISITOS

	<u>Sigla</u>	<u>Descripcion</u>	<u>Sem</u>	<u>Sigla Prerequisito</u>	<u>Descripcion Prerequisito</u>	<u>Sem Pre</u>
Eliminar	CCE-110	INVESTIGACION EDUCATIVA I	1	CCE-150	FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION	1
Eliminar	HIS-134	HISTORIA DE LA EDUCACION BOLIVIANA	1	TL-100	TALLER DE LENGUAJE I	1
Eliminar	TL-101	TALLER DE LENGUAJE II	2	HIS-135	HISTORIA DE LA EDUCACION LATINOAMERICANA	2
Eliminar	CCE-121	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION II	2	TL-101	TALLER DE LENGUAJE II	2
Eliminar	CCE-111	INVESTIGACION EDUCATIVA II	2	CCE-110	INVESTIGACION EDUCATIVA I	1
Eliminar	CCE-111	INVESTIGACION EDUCATIVA II	2	TL-100	TALLER DE LENGUAJE I	1
Eliminar	CCE-130	PEDAGOGIA GENERAL I	3	CCE-100	ESTADISTICA I	3
Eliminar	LIN-157	QUECHUA III	10	LIN-157	QUECHUA III	10

Fuente: Elaboración propia

- (1) Registrar a los docentes que dictarán las clases, esto si aún no están registrados.

Figura 4.6 Búsqueda de docentes y registro

BUSCADOR DE DOCENTES:

APELLIDO PATERNO: APELLIDO MATERNO: NOMBRE:

CARIET	AP PATERNO	AP MATERNO	NOMBRES	
	HUANCA	RODRÍGUEZ	ORLANDO VILTOR	<input type="button" value="Seleccionar"/>
	HURTADO	LOPEZ	MARGARETH T.	<input type="button" value="Seleccionar"/>

Fuente: Elaboración propia

Figura 4.7 Habilitación de materias y paralelos para la inscripción

HABILITAR MATERIAS Y PARALELOS

Unidad académica :

Gestión :

Periodo :

Carreras

Materias

- CCE-110 - INVESTIGACION EDUCATIVA I**
- CCE-120 - INTRODUCCION A LAS CIENCIAS DE LA EDUCACION I
- CCE-150 - FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION
- CSO-107 - SOCIOLOGÍA GENERAL

Paralelo :

Turno :

	SIGLA	NOMBRE	PARALELO	SEMESTRE	
<input type="button" value="Eliminar"/>	CCE-110	INVESTIGACION EDUCATIVA I	A	1	<input type="button" value="Asig. de horarios >>"/>
<input type="button" value="Eliminar"/>	CCE-120	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION I	A	1	<input type="button" value="Asig. de horarios >>"/>

Fuente: Elaboración propia

Figura 4.8 Pensum grafico de materias aprobadas, inscritas y convalidadas

INSCRIBIR ESTUDIANTE

BUSCADOR DE ESTUDIANTE:

<< [Buscar estudiante](#)

CCE-11D	CCE-111	CCE-112	CCE-113	CCE-114	CCE-11E	CCE-185	CCE-190
CCE-12J	CCE-121	CCE-13U	CCE-131	CCE-13U	CCE-181	CCE-186	CCE-191
CCE-15D	PSI-100	PSI-203	PSI-204	PSI-21	PSI-212	CCE-102	CCE-187
HIS-134	HIS-135	CCE-170	CCE-171	CCE-140	CCE-13E	CCE-188	CCE-193
CSO-107	CSO-132	CCE-100	CCE-101	CCE-116	CCE-117	CCE-189	CCE-194
TL-10C	TL-101	FL-213	FIL-219	CCE-136	CCE-11E	-	-

LIN-108	LIN-109	LIN-110
LIN-131	LIN-132	LIN-133
LIN-155	LIN-156	LIN-157

CCE-200	CCE-210	CCE-220	CCE-230	CCE-240	CCE-25C	CCE-260	CCE-270	CCE-280
CCE-300	CCE-310	CCE-320	CCE-330	CCE-340	CCE-35C	CCE-360	CCE-370	CCE-380
CCE-400	CCE-410	CCE-420	CCE-430	CCE-440	CCE-45C	CCE-460	CCE-470	CCE-480
CCE-600	CCE-610	CCE-620	CCE-630	CCE-640	CCE-65C	CCE-660	CCE-670	CCE-680

INFORMACION DEL ESTUDIANTE

CARNET: 6145732

TERCEROS NUNEZ NADESHDA LILIAN

Fuente: Elaboración propia

Figura 4.9 Inscripción de materias por el administrador

INFORMACION DEL ESTUDIANTE

CARNET: 6145732

TERCEROS NUNEZ NADESHDA LILIAN

CARRERA(S) INSCRITA(S)

UNIDAD ACADÉMICA	CARRERA	
Universidad Mayor de San Andres	Ciencias de la Educación	Ver inscripción >>

Carrera:

Ciencias de la Educación

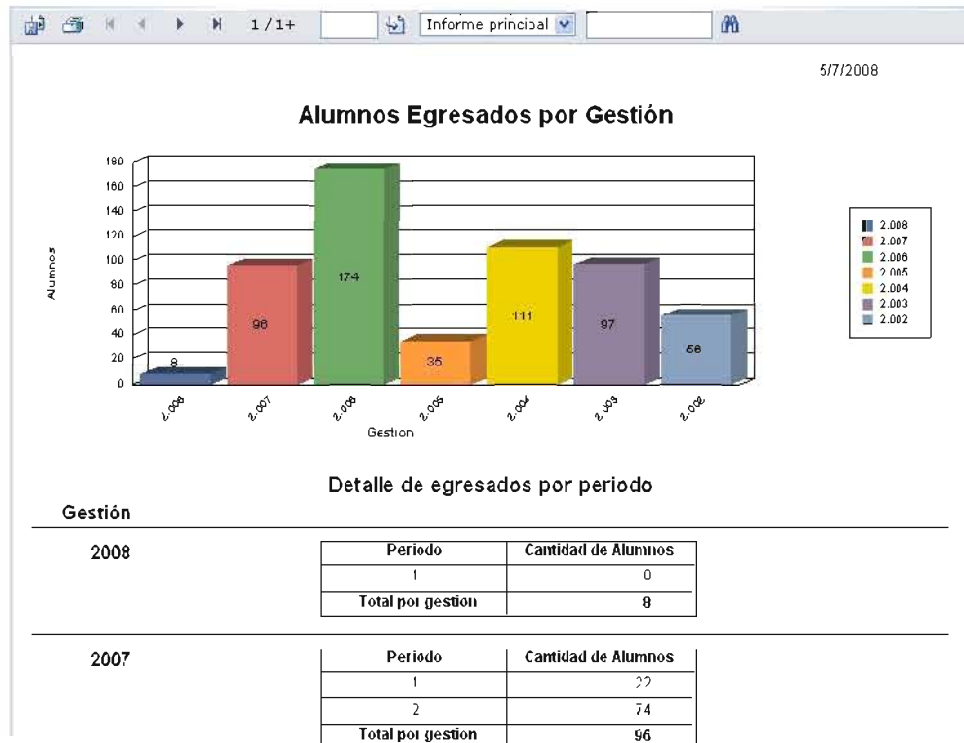
Materias:

Inscribir

	SIGLA	NOMBRE	PARALELO
Eliminar	CCE-187	PROBLEMAS DEL APRENDIZAJE	A
Eliminar	CCE-190	DISEÑO DE PROYECTOS EDUCATIVOS	B
Eliminar	CCE-193	COMUNICACION EDUCATIVA	A
Eliminar	CCE-194	EDUCACION POPULAR Y ALTERNATIVA	A
Eliminar	CCE-191	FUNDAMENTOS DE PSICOPEDAGOGIA	A

Fuente: Elaboración propia

Figura 4.10 Reporte de Alumnos Egresados por gestión



Fuente: Elaboración propia

Figura 4.11 Listado de alumnos por Materia

REPORTE LISTADO DE ESTUDIANTES POR MATERIA

CARRERA: Ciencias de la Educación
 GESTIÓN: 2008
 PERIODO: 1
 MATERIA: INVESTIGACION EDUCATIVA I
 PARALELO: A

CARGAR REPORTE

UNIVERSIDAD MAYOR DE SAN ANDRÉS 5/7/2008

LISTA DE ALUMNOS
 CCE-110 - INVESTIGACION EDUCATIVA I

CARRERA: Ciencias de la Educación PARALELO: A
 DOCENTE: Ph D Emilio Oros Mendez GESTION ACADEMICA: 1 - 2LU3
 SEMESTRE: 1

Nº	CARNET	NOMBRE
1	6726913	ACHA FLXO CI AUDIA ANDREA
2	8326851	APAZA DOLORE CARMEN
3	6368938	APAZA MAMANI WILMA VICTORIA
4	7037318	APAZA ZEGARRA JULIA
5	5348933	ARTEAGA GUTIERREZ EDGAR JOAQUIN

Fuente: Elaboración propia

Figura 4.12 Boletas de Inscripción por Gestión

Gestión: 2008
 Período: 1
 Generar Boletas

1 / 2+ Informe principal

Universidad Mayor de San Andrés
 Facultad de Humanidades
 Carrera de Ciencias de la Educación

INSCRIPCION - Periodo Académico : 2008

COD: AAG4283581 Gestión: 1 - 2008
 Nombre: Univ. Apaza Alvarez Guido Caleb CI: 4283581

SIGLA	MATERIA	PARALELO
TL-100	TALLER DE LENGUAJE I	A
LIN-108	AYMARAI	B
CCE-120	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION I	A

Numero de Materias Inscritas: 3

 JEFE DE CARRERA

 KARDIXTA

Fuente: Elaboración propia

Figura 4.13 Actas de notas

UNIVERSIDAD MAYOR DE SAN ANDRES

ACTA DE NOTAS

CCE-110 - INVESTIGACION EDUCATIVA I

FACULTAD: HUMANIDADES PARALELO: A
 CARRERA: CIENCIAS DE LA EDUCACION GESTION: 1 - 2008

NOTA: CALIFICAR SIN DECIMALES

Nro	CI	NOMBRE	Not Fin.	2do Tur.	Observaciones
1	6726913	ACHÁ SUXO CLAUDIA ANDRÉA
2	8326651	APAZA COLQUE CARMEN
3	6966938	APAZA MAMANI WILMA VICTORIA
4	7037318	APAZA ZEGARRA JULIA
5	5948993	ARTEAGA GUTIERREZ EDGAR JOAQUIN
6	7020081	ARUQUIPA CHURA ROSSEMARY
7	6055864	BAPTISTA ARANDA ARIEL RODRIGO
8	6847324	CADENA HUANCA LIZETH MAGDALENA
9	4747273	CANDIA QUISPE AYDEE BAEZA
10	6783546	CASTRO ILLANES MARCO ANTONIO
11	4366747	CATARI ROJAS MODESTA MIRIAM

Fuente: Elaboración propia

Figura 4.14 Reporte de Historiales Académicos

HISTORIAL ACADEMICO										
CARRERA: Ciencias de la Educación										
NOMBRE: TERCEROS NUNEZ			NADESHDA LILIAN							
CI: 6145732										
GESTIONE: 1 - 2004										
SIGLA	DESCRIPCION	NOTA	2do TURNO	NOTA FINAL	LIB	FOL	RES	OBS		
CCE-110	INVESTIGACION EDUCATIVA I	66	o	66						
CCE-120	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION I	76	o	76						
CSO-107	SOCIOLOGIA GENERAL	57	o	57						
HIS-134	HISTORIA DE LA EDUCACION BOLIVIANA	59	o	59						
PSI-100	FUNDAMENTOS PSICOLOGICOS DE LA EDUCACION	75	o	75						
TL-100	TALLER DE LENGUAJE I	74	o	74						
No Materias		6	Aprobadas	6	Reprobadas	0	Abandono	0	Promedio	67
GESTIONE: 2 - 2004										
SIGLA	DESCRIPCION	NOTA	2do TURNO	NOTA FINAL	LIB	FOL	RES	OBS		
CCE-111	INVESTIGACION EDUCATIVA II	68	o	68						
CCE-121	INTRODUCCION A LAS CIENCIAS DE LA EDUCACION II	71	o	71						
CCE-150	FUNDAMENTOS BIOLÓGICOS DE LA EDUCACION	62	o	62						
CSO-132	SOCIOLOGIA DE LA EDUCACION	51	o	51						

Fuente: Elaboración propia

Figura 4.15 Pantalla de Ingreso del Estudiante

Fuente: Elaboración propia

Figura 4.16 Reporte de Materias inscritas

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE HUMANIDADES
CARRERA DE CIENCIAS DE LA EDUCACIÓN

LISTA DE MATERIAS A LLEVAR

CI: 6145732
NOMBRE: TERCEROS NÚÑEZ NADESHDA LILIAN

SIGLA	DESCRIPCIÓN	PAR.
CCE-187	PROBLEMAS DEL APRENDIZAJE	A
CCE-190	DISEÑO DE PROYECTOS EDUCATIVOS	B
CCE-193	COMUNICACIÓN EDUCATIVA	A
CCE-194	EDUCACIÓN POPULAR Y ALTERNATIVA	A
CCE-191	FUNDAMENTOS DE PSICOPEDAGOGÍA	A

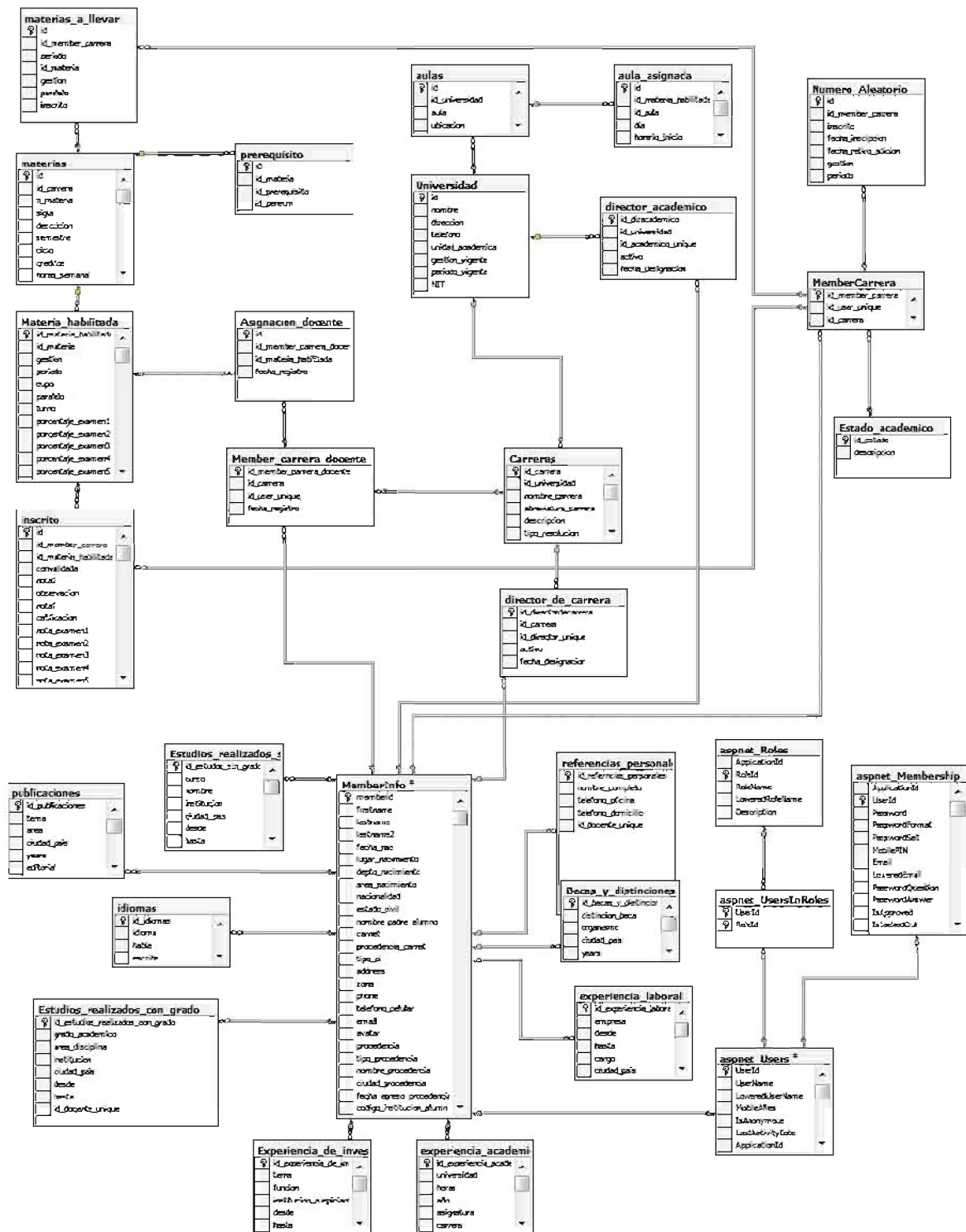
Fuente: Elaboración propia



4.5 MODELO FÍSICO DE LA BASE DE DATOS

El modelo físico de la Base de Datos de Campus Virtual, se muestra en la figura

Figura 4.17: Modelo físico de Base de Datos del Sistema Académico



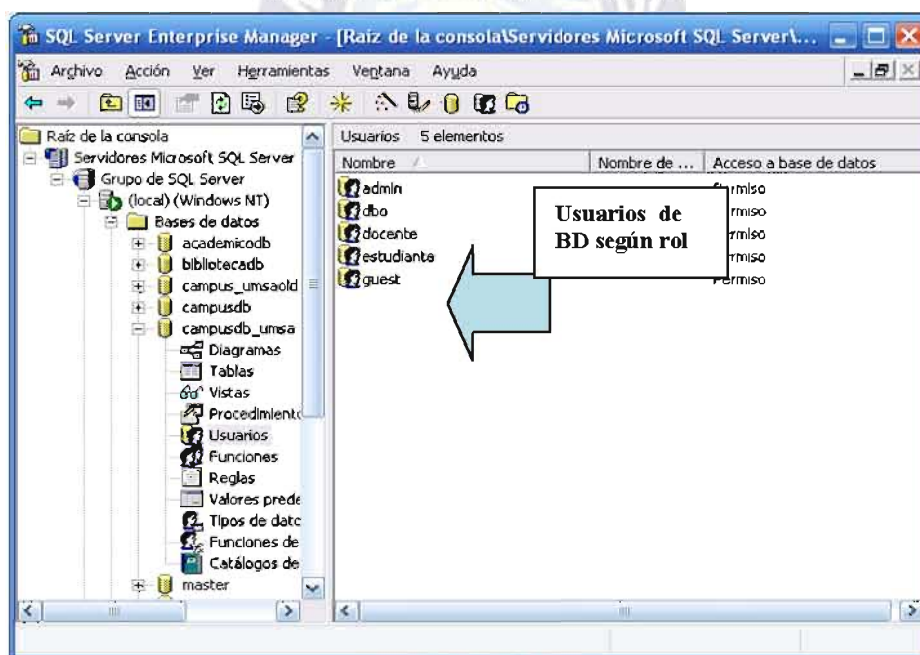
Fuente: Elaboración propia

4.6 SEGURIDAD

De acuerdo a las recomendaciones que sugiere la ISO 17799, Las medidas de seguridad que se tomaron con el propósito, de controlar el acceso de los usuarios al sistema y conservar la información que existe en ella, se tomaron en varias medidas, a continuación se citan algunas de ellas:

- Definición Propietario de los datos.
- Se creó usuarios con mínimos privilegios según al rol al que pertenecen, con el que establece la conexión con la base de datos.

Figura 4.18: Usuario para la conexión a la Base de Datos.



Fuente: SQL Server 2000 Enterprise Manager

- Se habilitaron las opciones de autenticación y autorización de ASP .NET, para controlar el acceso a la aplicación Web.
- A excepción de los administradores, los demás usuarios cuentan con privilegios mínimos.
- Se está realizando el control de posibles errores o excepciones que puedan ocurrir (caída del servicio Internet), evitando mostrar información innecesaria.

- Aviso y control de violaciones de seguridad y preparación de informes. Registros LOG.
- Las contraseñas de los usuarios se encuentran encriptados.
- En cuanto a la seguridad física, el servidor Web y el servidor de base de datos se encuentran en él un mismo ordenador, donde se tiene acceso solo por personal autorizado por la dirección de la Carrera.
- La seguridad lógica, el servidor toma los recaudos necesarios para proteger el servidor Web, la Base de Datos mediante Antivirus, Antispy y firewall.

4.7 VALIDACIÓN Y MANEJO DE ERRORES

Los errores encontrados no son mostrados a los usuarios finales, con la finalidad de no mostrar información que pueda ser útil al un atacante, se está realizando validaciones de los tipos de datos, de la longitud de los datos y los tipos de caracteres que se ingresan utilizando controles de validación y también utilizando expresiones regulares en los mismos.

Además, se están controlando las posibles excepciones que puedan surgir por diferentes causas (Uso Try Catch Exception). La Figura muestra uno de las medidas para la validación de datos.

Figura 4.19: Validación de entrada de datos

The figure displays two side-by-side screenshots of a web form titled "NUEVO DOCENTE".

Left Screenshot (Valid Form):

- Grado Académico:** Licenciado (dropdown menu)
- Apellido Paterno:** perez (text input)
- Apellido Materno:** Palomino (text input)
- Nombres:** juan (text input)
- N° de documento de identidad:** 5986633 (text input)
- Sexo:** Masculino Femenino
- Fecha nacimiento:** 26/07/1600 (text input)

Right Screenshot (Form with Error):

- Error Message:** ErrorDesbordamiento de SqlDateTime. Debe estar entre 1/1/153 12:00:00 AM y 12/31/9999 11:59:59 PM.
- Grado Académico:** Licenciado (dropdown menu)
- Apellido Paterno:** perez (text input)
- Apellido Materno:** Palomino (text input)
- Nombres:** juan (text input)
- N° de documento de identidad:** 5986633 (text input)
- Sexo:** Masculino Femenino
- Fecha nacimiento:** 26/07/1600 (text input)

Fuente: Elaboración propia

4.8 MODELO DE PRUEBAS

Probar un producto software es relativamente independiente de la metodología de desarrollo utilizada para construirlo.

El objetivo del modelo de pruebas de un programa es el de detectar todo posible malfuncionamiento antes de que entre en producción. Un error detectado en el laboratorio puede ser costoso de reparar; pero siempre es peor que el error le aparezca al usuario final.

Si pudiéramos probar un programa con todos los posibles datos de entrada, tendríamos una batería de pruebas perfecta. Lamentablemente, casi nunca es posible probar con todos los casos. En consecuencia necesitamos un criterio para elegir qué casos probamos.

Dentro de las metodologías de prueba se recomienda las siguientes:

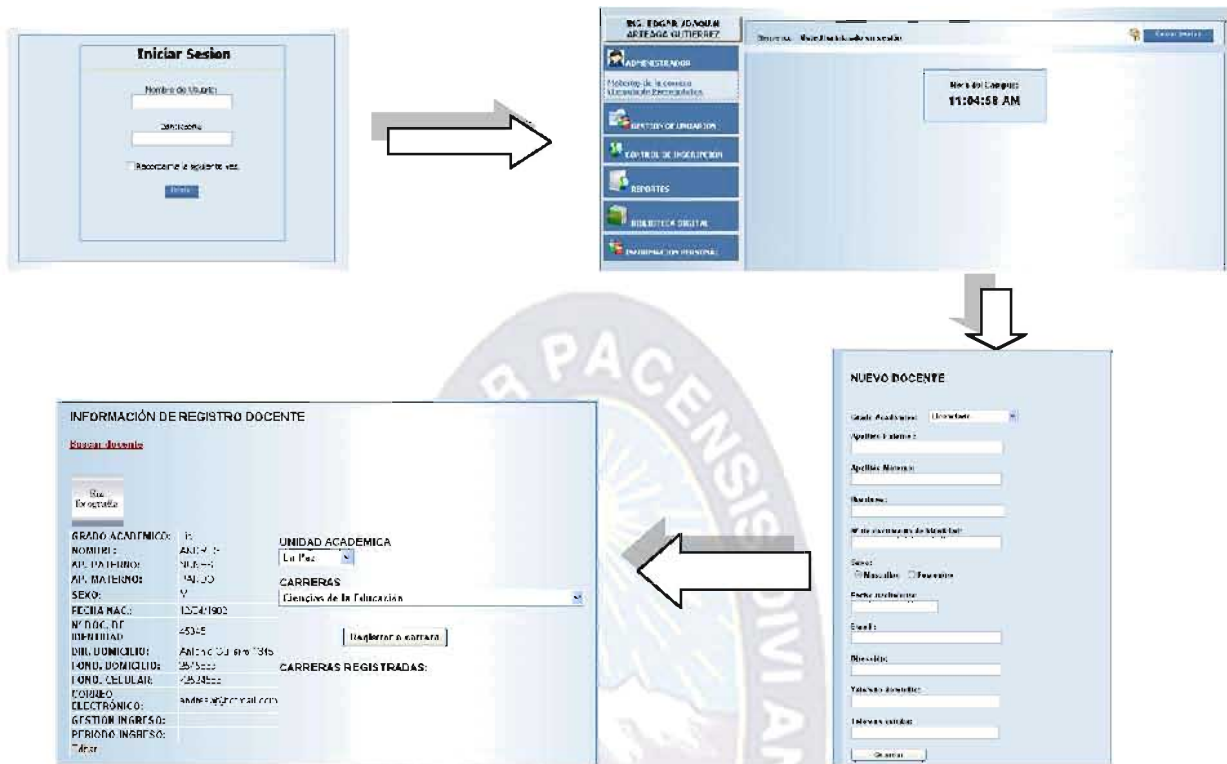
- ✓ En primer lugar realizar pruebas de "caja negra".
- ✓ Superadas las pruebas anteriores, realizar pruebas de "caja blanca" complementarias.
- ✓ Terminar con pruebas adicionales "de condiciones".

4.8.1 PRUEBA DE CAJA NEGRA

Denominada también la prueba de comportamiento o prueba de participación, está centrada en los requisitos funcionales del Software. Este tipo de prueba permite al ingeniero de software obtener un conjunto de condiciones de entradas que ejercitan completamente todos los requisitos funcionales de un programa.

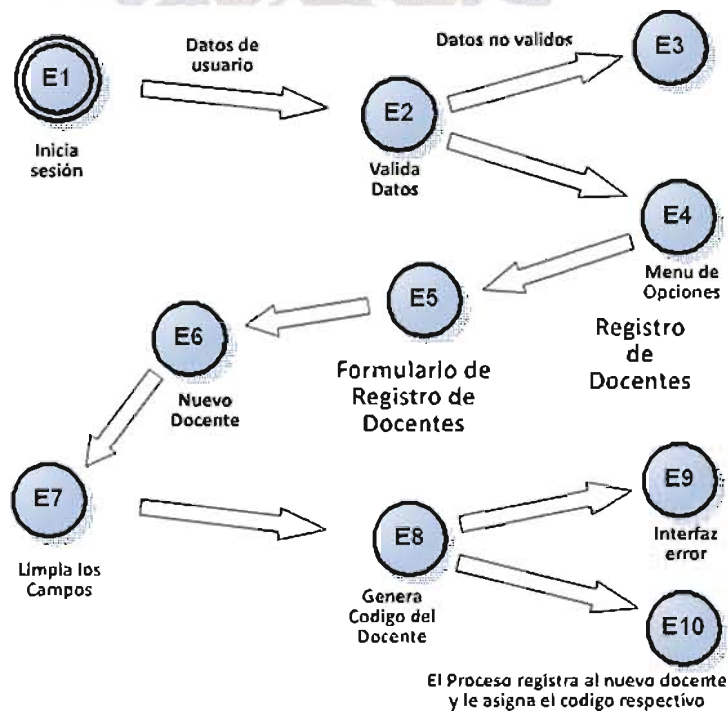
La Figura muestra la prueba de caja negra que comienza con la petición de la contraseña para un usuario determinado, seguidamente está la pantalla con menú principal donde elige el enlace correspondiente para ingresar a la interfaz de noticias, y presionar el botón nueva noticia.

Figura 4.20: Diagrama de caja negra



Fuente: Elaboración propia

Figura 4.21: Grado de la prueba de caja negra.



Fuente: Elaboración propia

Las condiciones de entrada inicialmente son

Login: condición de entrada, lógica – la palabra puede estar o no presente;
Condición de entrada, valor – cadena de hasta 50 caracteres

Password: condición de entrada, lógica – la palabra puede estar o no presente;
Condición de entrada, valor – cadena de hasta 50 caracteres

Los valores que pueden tener estas variables, se muestran en la siguiente tabla:

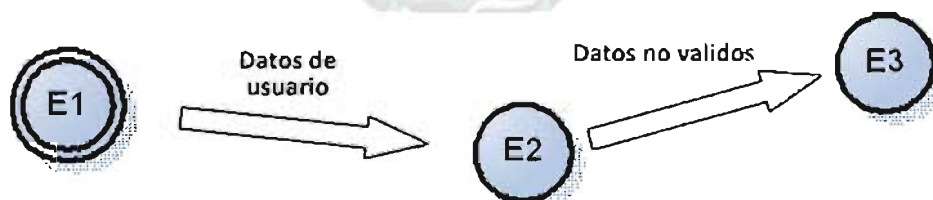
Tabla 4.1: Valores de las condiciones de entrada

Caso	Valores
A	Login: correcto Password: incorrecto
B	Login: incorrecto Password: correcto
C	Login: incorrecto Password: incorrecto
D	Login: correcto Password: correcto

Fuente: Elaboración propia

Para los casos A, B y C, el grafo toma el siguiente camino (clases de equivalencia no válidas):

Figura 4.22: Camino que se sigue en el caso de ingresar datos incorrectos



Fuente: Elaboración propia

Para el caso D, el grafo seguirá el curso normal (clase de equivalencia válida).

Posteriormente, cuando se elige el botón nueva noticia, las condiciones de entrada son:

Código de la noticia: condición de entrada, lógica – la palabra puede estar o no presente;

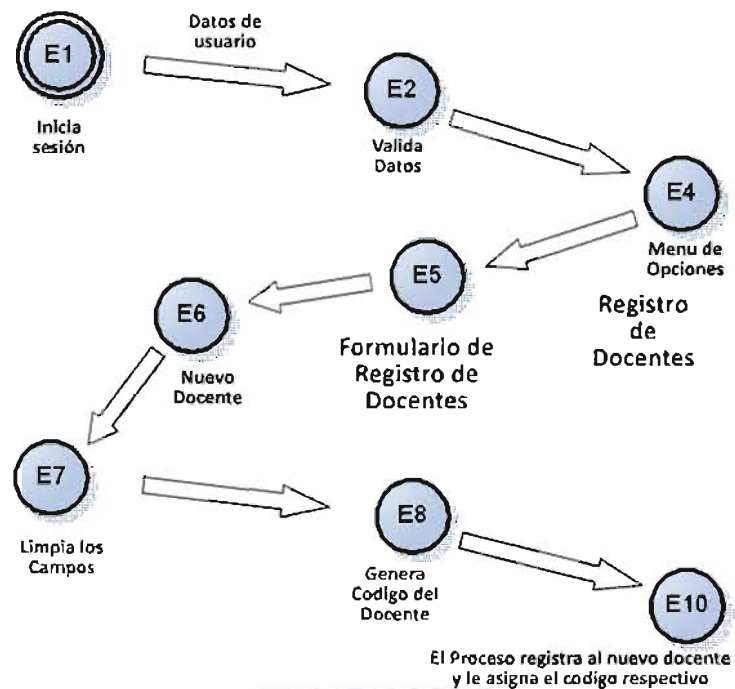
Tabla 4.2: Valores de las condiciones de entrada
 Condición de entrada, valor – cadena de hasta 10 caracteres.

Caso	Valores
A	Código: nulo
B	Código: no nulo

Fuente: Elaboración propia

Para el caso A, el camino será el siguiente, este caso puede ocurrir si falla conexión con la base de datos o la tabla no existiera.

Figura 4.23: Grafo completo para registrar un nuevo proyecto

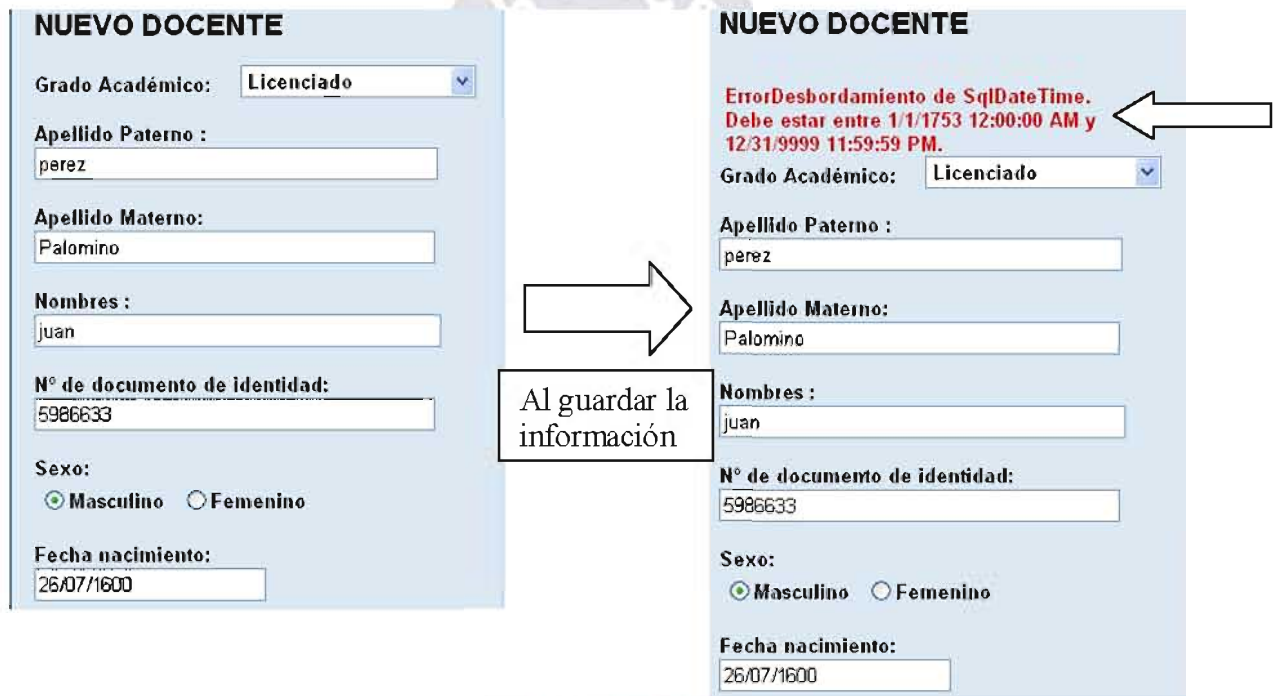


Fuente: Elaboración propia

4.8.2 PRUEBA DE CAJA BLANCA

Denominada también prueba de caja de cristal, es un método de diseño de casos de prueba, que utiliza la estructura de control del diseño procedimental, así obtener los casos de prueba. La Figura muestra la prueba de caja blanca que valida los datos introducidos para que estos no sean valores nulos, esta acción se efectúa cuando el usuario desea guardar los datos introducidos si hay campos incompletos, este prueba fue realizada con depurador de Visual Studio 2005.

Figura 4.24: Diagrama de prueba de caja blanca.



Fuente: Elaboración propia

CAPÍTULO V – CALIDAD DE LA APLICACIÓN WEB

5.1 CALIDAD DE SOFTWARE

Pressman [2] especifica un conjunto de factores de calidad, entre los que se tomaron para realizar el análisis de calidad:

- a) La funcionalidad se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.
- b) La fiabilidad se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de los resultados, el tiempo medio de fallos, la capacidad de recuperación de un fallo y la capacidad de predicción del programa.
- c) La capacidad de soporte combina la capacidad de ampliar el programa, adaptabilidad y servicios (estos tres atributos presentan un término más común, mantenimiento), así como capacidad de hacer pruebas, compatibilidad, capacidad de configuración, la facilidad de instalación y la facilidad con que se pueden localizar problemas.

a) Funcionalidad: Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como valor de normalización. Los Puntos de Función miden el software cualificando la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema [7]. Para calcular puntos de función (PF), se utiliza la ecuación siguiente:

$$PF = \text{cuenta-total} * [0.65 + 0.01 * \sum F_i] \quad (1)$$

Donde cuenta-total es la suma de total las entradas obtenidas, F_i ($i = 1$ a 14) son "valores de ajuste de complejidad". Los valores constantes de la ecuación anterior y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente.

Tabla 5.1, Cálculo de puntos de función

Parámetro de Medición	Cuenta	Factor de Ponderación			Resultados
		Simple	Medio	Complejo	
Número de entradas de Usuario	19	3	X4	6	76
Número de Salidas de Usuario	21	4	X5	7	105
Número de Peticiones de Usuario	14	3	X4	6	56
Número de Archivos	30	7	X10	15	300
Número de Interfaces Externas	6	5	X7	10	42
Cuenta Total					579

Fuente: Pressman

La cuenta - total tiene un valor de 579, se toma el valor de ponderación medio de acuerdo al tipo de proyecto, obteniendo así la cuenta - total que será utilizada en la fórmula del punto de función.

Para los valores de ajuste complejidad:

Después de la computación de los valores de ajuste de complejidad que se observan en la tabla 5.2, se tiene que $\sum F_i = 33$

Entonces

$$PF_{real} = 579 * [0.65 + 0.01 * 33]$$

$$PF_{real} = 567.42$$

Este valor representa una funcionalidad real de 567.42 con un nivel de confianza del 65 %.

La funcionalidad esperada con un nivel de confianza del 100% es:

$$PF_{esperado} = 579 * [1 + 0.01 * 33]$$

$$PF_{esperado} = 770.07$$

Calculando el porcentaje de funcionalidad al que se llegó, se tiene:

$$\%PF = (PF_{real} / PF_{esperado}) = (567.42 / 770.07) = 0.73$$

Lo cual significa que el Sistema de Información y gestión académica tiene una funcionalidad del 73%

Tabla 5.2, Valoración de ajustes de Complejidad

Nro.	Características	Significado	Valor
1.	Comunicación de Datos	Más de un ordenador front-end, pero la aplicación soporta un solo tipo de protocolo de comunicaciones.	4
2.	Funciones Distribuidas	El proceso distribuido y la transferencia de datos son on-line y sólo en una dirección.	3
3.	Rendimiento	Rendimiento y requisitos de diseño han sido definidos y revisados pero no requieren ninguna acción especial.	1
4.	Configuraciones fuertemente utilizadas	Existen restricciones operativas, pero no requieren un esfuerzo especial para conseguirlas.	1
5.	Frecuencia de Transacciones	No existe una definición del periodo punta de transacciones.	0
6.	Entrada on-line de datos	Más del 30% de las transacciones son interactivas.	5
7.	Diseño para la eficiencia del usuario final	6 o más funciones, pero no existen requisitos del usuario respecto a la eficiencia.	3
8.	Actualización on-line	Ninguno.	0
9.	Procesos Complejos	Controles especiales (procesos de auditoría) y/o aplicaciones de seguridad. Excesivas excepciones de proceso dando lugar a transacciones incompletas que deben ser procesadas de nuevo	3
10.	Reutilización	La aplicación fue empaquetada expresamente y/o documentada para ser fácilmente reusable. La aplicación es adaptada por el usuario a nivel de código fuente.	4
11.	Facilidad de Instalación	No se realizaron consideraciones especiales por el usuario pero se requirieron desarrollos especiales instalación.	1
12.	Facilidad de Operación	Procesos eficaces de arranque, respaldo y recuperación pero con intervención del operador	2
13.	Instalación de múltiples sitios	Se necesita diseñar la aplicación para ser utilizada en distintos lugares y funcionará bajo entornos distintos de hardware y software.	2
14.	Facilidad de Cambio	Facilidad para realizar consultas o informes de complejos. Se mantendrán datos de control en tablas que serán mantenidas por los usuarios a través de procesos interactivos on-line	4
TOTAL (I F j)			33

Fuente: Pressman

b) Fiabilidad: Para calcular la Habilidad, se emplea la métrica de eficacia de eliminación de defectos (EED), empleando la siguiente forma:

$$EED = E / (E + D) \quad (2)$$

Donde:

E es el número de errores encontrados antes de la entrega del software.

D es el número de errores encontrados después de la, entrega del software.

Calculando EDD se tiene:

$$EDD = 19 / (19 + 1) = 0.95$$

Esto significa una eficacia de eliminación de errores de 95 %.

d) Capacidad de soporte: Como el Sistema fue desarrollado de manera modular, el mantenimiento del mismo se hace de manera más específica (Ej. Inscripciones, registro de notas, etc.), en lo referente a las interfaces y procesos, con respecto a la base de datos, gracias al uso de los procedimientos almacenados, cualquier tipo de modificación en la estructura de base de datos o en las consultas de selección o modificación, solo es necesario ajustar los procedimientos almacenados, sin la necesidad de modificar el código fuente del sistema.

5.2 EVALUACIÓN DE USABILIDAD (FOCUS GROUP)

Mediante la metodología de evaluación **Focus Group** (Grupo de discusión dirigido), se realizaron varias reuniones, armando los grupos de discusión a través del MSN Messenger, de las cuales se obtuvieron los siguientes resultados:

PRIMER GRUPO DE DISCUSIÓN

Implementación del modulo de registro de notas.

Temas a Discutir

- Facilidad de manejo.
- Complejidad de las funciones.
- Colores de la aplicación.
- Sugerencias para mejorar la facilidad de manejo.

Número de participantes

5 docentes se conectaron vía MSN.

Moderador

Edgar Arteaga

Resultados obtenidos

- La aplicación es muy intuitiva, solo se necesita de dos pasos para ingresar al registro de notas.
- La paginación personalizable en el listado de alumnos, ayuda a cargar las notas de una manera más cómoda.
- La opción de guardar las notas una vez llenadas, debería ser de forma automática.

- El modo de impresión del reporte de notas, es sencillo desde el explorador, pero al momento de exportar a Excel, se tiene que ajustar el tamaño de las celdas.
- Los colores son amigables, no fatigan la visión.
- Se debería poder deslizarse mediante los cursores, entre los cajones de texto de la misma manera que en el Excel.
- El llenado celda por celda es un poco moroso.
- Se debería tener otras opciones de colores para escoger y personalizar la aplicación.
- No existen elementos distractores como en otras páginas.
- Se deberían poder importar las notas de un archivo de Excel, para no tener que estar transcribiéndolas.

Conclusiones

- De los resultados obtenidos se puede concluir que el sistema es de fácil aprendizaje, pero en lo que respecta al manejo se encuentra en el límite permitido por la tecnología, considerando que se está trabajando en un entorno web, en el cual se debe guardar un equilibrio entre la funcionalidad y la ligereza de la aplicación.
- La combinación de colores ayuda a que el usuario no se fatigue al manejar la aplicación, y pueda operarla por más tiempo.
- En una segunda instancia crear un módulo de importación de notas directamente desde archivos Excel, poniendo en línea un archivo Excel con un formato definido, para evitar los errores.

SEGUNDO GRUPO DE DISCUSIÓN

Implementación del módulo de inscripciones.

Temas a Discutir

- Facilidad de manejo.
- Complejidad de las funciones.
- Colores de la aplicación.
- Sugerencias para mejorar la facilidad de manejo.

Número de participantes

27 alumnos se conectaron vía MSN.

Moderador

Edgar Joaquin Arteaga Gutierrez

Resultados obtenidos

- No fue difícil inscribir mis materias, pero algunas materias deberían inscribirse directamente, ya que vienen en pares, como ser estadística I y estadística II, que si se aprueba con un docente se debería continuar con el mismo en el mismo paralelo.
- Las instrucciones de inscripción son muy aclarativas, solo hay que seguir los pasos.
- Se debería habilitar alguna forma de intercambiar paralelos entre nosotros.
- Los colores son suaves y agradables.
- La selección automática de paralelos confunde un poco al comienzo, pero ayuda a evitar los choques en los horarios.
- Solo necesite 3 minutos para poder inscribirme, aunque no conocía la aplicación, pero debería haber habido alguna capacitación en el sistema, con el fin de aprovechar al máximo las funcionalidades de la aplicación.

Conclusiones

- De los resultados obtenidos se puede concluir que el sistema es de fácil aprendizaje y manejo.
- Los alumnos no tuvieron contratiempos al momento de su inscripción.
- Los alumnos solo tuvieron problemas en lo que respecta a las materias que no pudieron llevar debido a la malla de prerrequisitos.
- La combinación de colores ayuda a que el usuario no se fatigue al manejar la aplicación, y pueda operarla por más tiempo.
- Realizar una capacitación previa, para familiarizar a los usuarios en las funciones del sistema.

CONCLUSIONES GENERALES

De los foros de discusión realizados se pueden concluir que el sistema cuenta con la usabilidad y ergonomía esperada, pero como en muchos otros proyectos, una vez que el usuario opera el sistema, se le ocurren nuevos requerimientos los cuales ayudan a que la aplicación mejore cada vez más, haciendo cumplir lo que nos dice las reglas de la ergonometría, que el sistema se adapte al usuario mediante el empleo de varias iteraciones de desarrollo, ajustándose a las necesidades y capacidades particulares de cada persona, consiguiendo un desarrollo centrado en el usuario.

5.3 ANÁLISIS COSTO/BENEFICIO

- a) Costos: En este punto se presenta el cálculo de costos que implica el desarrollo e implementación del Sistema Gestión Académica, para esto se toma en cuenta la experiencia laboral en el mercado boliviano. Los puntos tomados en cuenta se aprecian en la tabla 11, donde se observa un costo total de 6.880 dólares.

Tabla 5.3 Descripción de Costos

COSTOS	TOTAL \$us	OBSERVACIONES
Costo de Desarrollo		
Fase de Análisis	500	5 meses - 1 analista
Fase de Diseño	1.200	6 meses - 1 diseñador
Fase de Implementación	600	3 meses - 1 programador
Subtotal	2.300	
Hardware		
Servidor	2.500	Proliant HP
Subtotal	2.500	
Red		
Equipos de Red	100	1 switch
Configuración de Red	80	20 \$us por punto - 4 puntos
Subtotal	180	
Software		
Sistema Operativo Windows 2003	700	
Licencia de Visual Studio 2005 .NET	450	
Licencia de SQLServer 2000	400	
Subtotal	1550	
Implantación		
Instalación	50	2 día - 1 técnico
Capacitación	300	2 meses - 1 capacitador
Subtotal	350	
TOTAL	6.880	

- b) Beneficios: Se distinguen los siguientes

- Facilidad de manejo
- Mejoramiento de Procesos
- Se mejora y agiliza la toma de decisiones y la elaboración de reportes.
- Apoyo a procesos

- Facilidad de actualización de información personal de estudiantes y docentes.
- Apoyo la difusión de programas mediante la propagando automatizada por Internet.
- Reducción de esfuerzos en procesos
- Se realiza menos esfuerzo en mantener actualizado el historial de estudiantes y docentes
- La comunicación de resultados se realiza con menos tiempo y menos esfuerzo

Estos beneficios reducen la carga laboral del personal mejorando así su rendimiento, porque les permite trabajar enfocándose más en los proyectos de mejoramiento de la Carrera, incrementando la productividad del personal, sin la necesidad de contratar personal extra. Este beneficio se cuantifica comparando lo que se tendría que gastar contratando al personal necesario para obtener la misma productividad; en promedio un auxiliar en Kardex y Dirección de Carrera, para agilizar el manejo y procesamiento de la información Académica, durante dos años con un sueldo de 400 dólares por mes, en 2 años se tendría un ahorro de 9.600 dólares.



CONCLUSIONES

El presente Proyecto de Grado tiene como el logro principal la implementación del Sistema de Gestión Académica en la Carrera de Ciencias de la Educación.

Y los objetivos alcanzados son:

- ✓ Se realizó el relevamiento de datos de los procesos académicos, se obtuvieron las bases de datos de la carrera de Ciencias de la Educación.
- ✓ Se efectuó la migración de la base de datos al nuevo modelo de base de datos en SQLSERVER 2000, en la que se encontraron observaciones en los nombres, notas, materias de estudiantes por lo que se hizo una verificación y depuración de datos.
- ✓ Se redujo el tiempo de procesamiento de información de los procesos académicos, en la emisión de reportes: listados, record académico, historial académico, actas.
- ✓ Se implementó y está funcionando el sistema de gestión académica en la carrera. Se realizó un análisis, diseño y desarrollo módulos de acuerdo a los requerimientos de los usuarios de la carrera.
- ✓ Se ejecutaron las pruebas de funcionamiento para cada uno de los módulos que actualmente son utilizados por los usuarios del sistema.
- ✓ Se realizaron pruebas, para medir el grado de ergonomía de la aplicación mediante la utilización de grupos de discusión.
- ✓ Se realizaron manuales y capacitaciones a autoridades, kardixtas, docentes y estudiantes de la Carrera de Ciencias de la Educación.

RECOMENDACIONES

Se recomienda:

- ✓ Continuar con el proyecto realizar la implementación del sistema en la demás carreras de la Facultad de Humanidades.
- ✓ Continuar realizando las evaluaciones mediante los Focus Group, con la finalidad de que el sistema sea lo más ergonómico posible, para los usuarios finales.
- ✓ Módulo de envío de correos de forma automática, con información académica, solicitud de documentos, de esta manera se hará uso de las

tecnologías TIC para el crecimiento académico y la facilidad de comunicación entre usuarios.

- ✓ Automatizar el Manejo de la información del curso pre facultativo, realizando la integración con el sistema académico,
- ✓ La integración del sistema de gestión académica con el sistema de matriculación UMSATIC-DSIE para el caso de estudiantes nuevos.



BIBLIOGRAFÍA

1. KENDALL & KENDALL, Kenneth y Julie. (1997) "Análisis y Diseño de Sistemas". Tercera Edición. Editorial Prentice Hall. México.
2. PRESSMAN, ROGER S. (1997), "Ingeniería de Software, Un enfoque practico", Mc Graw Hill Interamericana de España S.A., 5ª edición.
3. FRANCESCO BALENA. (2003) "Programación avanzada con Microsoft Visual Basic.NET", Mc Graw Hill Interamericana de España, S.A.U.
4. STEPHEN WALTHER (2002), "ASP.NET Al descubierto", Editorial Prentice Hall.
5. SENN, J., (1992) "Análisis y Diseño de Sistemas de Información", Mc Graw México.
6. MORENO MUÑOZ, ANTONIO et al (1997). "Ergonomía y usabilidad diseño centrado en el usuario".
7. XAVIER FERRÉ GRAU, MARÍA ISABEL SÁNCHEZ SEGURA, "Desarrollo Orientado a Objetos con UML", Facultad de Informática – UPM, Artículo 3
8. PATRICIO LETELIER TORRES, "Desarrollo de Software Orientado a Objeto usando UML", Departamento Sistemas Informáticos y Computación (DSIC), Universidad Politécnica de Valencia (UPV) - España, letclirfojdsic.upv.es.
9. JESÚS LORÉS, TONI GRANOLLERS, "La Ingeniería de la Usabilidad y de la Accesibilidad aplicada al diseño y desarrollo de sitios web", Departamento de informática, Universidad de Lleida, Campus de Cappont
10. LORÉS, J. 2002, "Introducción a la Interacción Persona-Ordenador".
11. MORGAN-KAUFMAN 2002, "Usability for the Web".
12. LYNCH, PATRICK; HORTON, SARAH 2000, "Principios básicos de diseño para la creación de sitio WEB".
13. NIELSEN, J 2000, "Usabilidad diseños de sitios WEB". PrenticeHall.
14. J.B. SHNEIDERMAN, "Designing the User Interface: Strategies for Effective Human-Computer"