

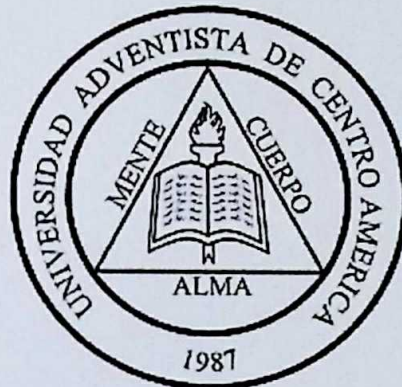
UNA DE CA

Alajuela, Costa Rica

Universidad Adventista de Centroamérica

Proyecto de Graduación

Desarrollo de los módulos de matrícula y horarios de la aplicación Web de
registro



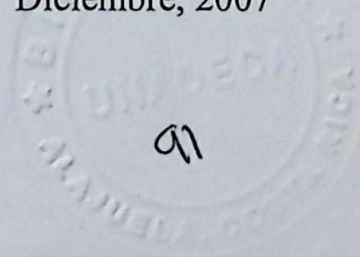
Ingeniero:

Jhonny Banegas

Asesor:

Osman Alejandro Longa

Diciembre, 2007



ÍNDICE GENERAL

ÍNDICE DE TABLAS	2
ÍNDICE DE FIGURAS	2
CAPÍTULO 1: ANTEPROYECTO	4
1. Introducción	4
1.1. Identificación del proyecto	4
1.2. Justificación	6
1.3. Finalidad y objetivos	7
1.3.1. Finalidad.....	7
1.3.2. Objetivos.....	8
1.3.2.1. General.....	8
1.3.2.2. Específicos	8
2. Marco de referencia institucional y teórico	9
2.1. Marco de referencia institucional	9
2.2. Marco de teórico	17
2.2.1. Lenguajes de programación	19
2.2.1.1. PHP	19
2.2.1.2. UML.....	28
2.2.2. Sistemas para la administración de bases de datos.....	33
2.2.2.1. MySQL.....	33
2.2.3. Servidores Web.....	35
2.2.3.1. Servidor Web Apache.....	36
3. Productos y resultados	36
4. Cobertura del proyecto.....	37
CAPÍTULO 2: ANÁLISIS DE REQUERIMIENTOS DE SOFTWARE.....	37
CAPÍTULO 3: ANÁLISIS Y DISEÑO DE SISTEMAS.....	51
CAPÍTULO 4: DESARROLLO	72
CAPÍTULO 5: PRUEBAS.....	77
CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES	89
CAPÍTULO 7: GLOSARIO	90
CAPÍTULO 8: BIBLIOGRAFIA.....	98
CAPÍTULO 9: ANEXOS	102

ÍNDICE DE TABLAS

Tabla # 1: Estructura administrativa.....	14
Tabla # 2: Estructura académica.....	14
Tabla # 3: Autoridades de apoyo	15
Tabla # 4: Carreras	16

ÍNDICE DE FIGURAS

Figura # 1: Código para introducir datos del alumnos a la base de datos	25
Figura # 2: Código para extraer datos del alumnos desde la base de datos.....	28
Figura # 3: Caso de uso para los directores de escuelas.....	31
Figura # 4: Diagrama de clases para los alumnos	32
Figura # 5:Diagrama de secuencia para los alumnos	33
Figura # 6: Diagrama de caso de usos del módulo de horarios para el vicerrector académico	52
Figura # 7: Diagrama de caso de usos del módulo de matrícula para el personal de registro.....	53
Figura # 8: Diagrama de caso de usos del módulo de matrícula para el vicerrector financiero/estudiantil.....	54
Figura # 9: Diagrama de caso de usos del módulo de matrícula para los directores de escuelas	55
Figura # 10: Diagrama de caso de usos del módulo de matrícula para los alumnos	56
Figura # 11: Diagrama de clases para el módulo de horario	57
Figura # 12: Diagrama de clases para el módulo de matrícula, usuario: registro.....	57
Figura # 13: Diagrama de clases para el módulo de matrícula, usuario: vice rectoria financiera/estudiantil	58
Figura # 14: Diagrama de clases para el módulo de matrícula, usuario: directores de escuela.....	58
Figura # 15: Diagrama de clases para el módulo de matrícula, usuario: alumnos	59
Figura # 16: Diagrama de secuencia para el módulo de horario.....	60
Figura # 17: Diagrama de secuencia para el módulo de matrícula, usuario: registro	61
Figura # 18: Diagrama de secuencia para el módulo de matrícula, usuario: vicerrector financiero/estudiantil. 62	
Figura # 19: Diagrama de secuencia para el módulo de matrícula, usuario: directores de escuelas	62
Figura # 20: Diagrama de secuencia para el módulo de matrícula, usuario: alumnos	63
Figura # 21: Formulario para introducir la informacion de un nuevo estudiante.....	73
Figura # 22: Formulario principal de estudiantes	74
Figura # 23: Página en la que los directores de escuela buscan la informacion de pre-matricula del alumno ..	74
Figura # 24: formulario de pre-matricula del alumno.....	75
Figura # 25: formulario que muestra las clases impartidas por un profesor.....	76

Figura # 26: Alumnos matriculados en una clase	76
Figura # 27: Sección para introducir fechas	78
Figura # 28: Fallo al intentar crear las fechas con algunos campos requeridos omitidos	78
Figura # 29: Pantalla para introducir clases	79
Figura # 30: Falló la creación de una clase	80
Figura # 31: Sección de la WebApp para agregar facultades	80
Figura # 32: Falló la creación de una facultada	81
Figura # 33: Página para crear planes de estudio.....	82
Figura # 34: Error al crear plan de estudio, no se especificó la escuela a la que pertenece el plan a crear	83
Figura # 35: Página que permite definir pre-requisitos de clases	84
Figura # 36: Reporte de matricula	85
Figura # 37: página principal de la WebApp de registro	86
Figura # 38: El usuario se autenticó satisfactoriamente como director de escuela	87
Figura # 39: El usuario se autenticó como administrador de la WebApp y esta en la seccion de campus	88

Capítulo 1: Anteproyecto

1. *Introducción*

Los estudios universitarios tienen la obligación de evolucionar tanto de acuerdo con el avance científico-tecnológico como con las demandas profesionales que se generan, y los proyectos de fin de carrera son un componente cada vez más importante, en la práctica total de las actuales titulaciones universitarias relacionadas con la informática y los sistemas de información, cualquiera que sea su nivel académico (estudios de pregrado y postgrado).

El desarrollo de estos proyectos académicos requiere que el estudiante domine un importante número de destrezas y capacidades, aparentemente dispersas entre sí, aunque extraordinariamente importantes para su formación profesional: efectuar una correcta revisión bibliográfica, redactar informes, documentar adecuadamente el software, planificación temporal y de recursos, conocimientos en la gestión de proyectos, etc. El dominio de todas estas áreas es un objetivo a conseguir por el estudiante.

1.1. *Identificación del proyecto*

La Universidad Adventista de Centro América (UNADECA), cuenta con el departamento de Registro, el cual posee un sistema que ayuda en el proceso de matrícula, llevando un registro de los estudiantes matriculados, de los horarios de clases, e imprimir reportes

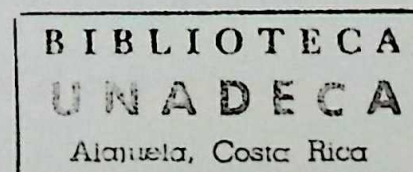
de matrícula de los alumnos; pero dicho sistema es incapaz de llevar un registro de todos los cuatrimestres (historial). Para archivar dicha información se debe imprimir los reportes del periodo actual porque dicho sistema no permite generar e imprimir información de cuatrimestres anteriores, es decir, no permite manipular información del periodo en curso. Por lo tanto este sistema ocasiona pérdida de tiempo y por ende de dinero para el departamento de registro.

El sistema que está siendo presentado existirá para permitir manipular y almacenar la información académica del alumno para tener la posibilidad de llevar un control histórico de los alumnos de manera fácil y segura.

El sistema en desarrollo básicamente comprende los módulos de:

1. Matrícula que automatizará el proceso de matrícula, permitiendo al director de escuela y alumnos llenar, modificar y visualizar el formulario de matrícula,
2. Horario que será usado por la vice-rectoría académica para definir, modificar y eliminar los diferentes horarios de clases.

La WebApp podrá ser operada por diferentes usuarios, los cuales son:



1. Vice-rectoría académica
2. Vice-rectoría financiera
3. Vice-rectoría estudiantil
4. Personal de registro
5. Director de escuela
6. Alumnos
7. Administradores

Por lo tanto, al iniciar la WebApp, ésta solicitará al usuario que se autentifique para poder cargar los módulos necesarios con los cuales trabajará el usuario dependiendo de sus credenciales.

1.2. Justificación

La automatización del sistema de registro de la Universidad Adventista de Centro América es una de las necesidades reales de dicha institución, por lo tanto no puede ser evadida en ésta generación tecnificada en la que la computadora se ha convertido en un medio indispensable para agilizar, procesar y almacenar grandes cantidades de información, tanto en el entorno empresarial como institucional.

La UNADECA debe tomar en cuenta los servicios académicos que otras instituciones similares ofrecen a sus alumnos para hacer del proceso de matrícula cómodo y práctico. Hecho por el cual se cree a

bien la incorporación de una WebApp que le brinde a la UNADECA una mejor imagen y la posibilidad de proporcionar mejores servicios al alumnado.

Este proyecto proveerá a la institución rapidez, solidez y seguridad en el proceso de matrícula, manejo y almacenaje de información de vital importancia.

1.3. Finalidad y Objetivos

1.3.1. Finalidad

Desde hace varios años, se han propuesto, analizado y desarrollado sistemas informáticos para el departamento de registro de la Universidad Adventista de Centro América, de los cuales ninguno se ha puesto en marcha para el beneficio de dicho departamento y personas que laboran en el.

Por lo tanto, el desarrollo de este proyecto tiene como meta primordial, el desarrollo e implementación de una aplicación Web, que llene las expectativas y necesidades organizativas, de control, mantenimiento y almacenamiento de la información relacionada con los alumnos de dicha institución.

1.3.2. Objetivos

1.3.2.1. General

Analizar y desarrollar una WebApp que cumpla con los requerimientos del departamento de registros de la UNADECA.

1.3.2.2. Específicos

1. Identificar en forma conjunta con el personal envuelto en el proceso de matrícula los requerimientos para el sistema en desarrollo.
2. Determinar claramente que posibilidades brinda la WebApp a las personas que tendrán acceso a dicha aplicación Web, así como también la redacción de la base de datos que almacenara la información de matrícula, horarios, profesores, estudiantes, etc.
3. Desarrollar la WebApp de matrícula de modo que se ajuste eficientemente a las necesidades de los usuarios.
4. Probar cada trayectoria de ejecución independiente en un módulo o página para asegurar que el sistema puede procesar la carga de información pretendida.

2. *Marco de referencia institucional y teórico*

2.1. *Marco de referencia institucional*

2.1.1. *¿Qué es UNADECA?*

UNADECA, Universidad Adventista de Centro América, es una institución educativa privada, que forma parte del gran movimiento Adventista en América Central, la cual es administrada por la Unión Centroamericana Sur de los Adventistas del Séptimo Día (UCSA), que permite a los jóvenes de dicho campo prepararse en áreas como Educación con énfasis en preescolar, primaria, nivel secundario en español, ciencias naturales, secretariado, matemáticas, computación, estudios sociales, inglés, religión, música además administración de empresas con énfasis en gerencia y contabilidad, ingeniería es sistemas computacionales, enfermería, teología y secretariado ejecutivo profesional bilingüe o español. Es una institución cuyos programas están estructurados con la finalidad de lograr el desarrollo armonioso de las facultades morales, espirituales y físicas del hombre (Wayne Casanova, Paúl. 1998. p. 5).

2.1.2. *Misión*

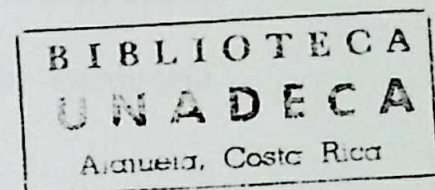
La misión de la Universidad Adventista reza de la siguiente manera:

“La Universidad Adventista de Centro América es una institución educativa religiosa comprometida con los valores y prácticas de la Iglesia Adventista del Séptimo Día. Opera con el fin de contribuir en la evangelización y la restauración por medio del entrenamiento y preparación de hombres y mujeres que aporten positivamente a la iglesia y a la sociedad, con sus talentos y capacidades dentro de un contexto altamente profesional, científico y actualizado” (Departamento de Mercadeo. UNADECA. 2003).

2.1.3. Visión

Ver a la Universidad Adventista de Centro América como una institución competitiva, que satisface las necesidades de la Iglesia Adventista del Séptimo Día y de la sociedad centroamericana, a través: de profesionales con una jerarquía relevante de valores espirituales, éticos, estéticos y cívicos; de presencia académica en los países de Centro América con programas de nivelación o de extensión; de nuevas oportunidades mediante carreras de informática y de postgrados en las escuelas de Educación, Administración de Empresas y de Enfermería; de la enseñanza del inglés como segundo idioma y del uso de la tecnología moderna y de Internet para el proceso de la enseñanza aprendizaje y para el desempeño profesional.

También como una institución difusora del mensaje evangélico a través de la obra testificadora individualizada y por Radio Lira, que a su vez alcanza el autofinanciamiento por medio de un sistema de empresas e industrias estudiantiles. (Manual de Ética Universitaria, 2003, p. 4)



2.1.4. Marco histórico-filosófico

La Universidad Adventista de Centroamérica UNADECA fue creada con el propósito de ofrecer educación universitaria cristiana a la juventud centroamericana, al mismo tiempo la misión de UNADECA es contribuir en la formación y la preparación de hombres y mujeres que con sus talentos y capacidades deje este mundo de lo que lo encontraron; con el propósito de lograr esto, la UNADECA tiene como objetivos:

Desarrollar el amor a Dios, la comunión personal con él, y desarrollar caracteres semejantes al de Cristo. (Mejía, 2004, p.1).

Comunicar a los alumnos los mejores conocimientos en las diferentes disciplinas, enseñándoles a evaluarlas y aplicarlas en la vida práctica con mente crítica, juicio independiente y disposición constructiva. Cultivar la investigación responsable tendiente al proceso. Promover las sanas relaciones interpersonales sazonadas y moduladas con la bondad, la

cortesía, la paciencia, la generosidad, la cooperación, la consideración, la ternura, la rectitud y la honestidad a toda prueba. (Ídem)

Enseñar los principios básicos que sirven de plataforma para la formación de hogares cristianos ejemplares y para la existencia responsable en la comunidad.

Guiar a los estudiantes a aceptar y practicar los conceptos éticos y morales del sistema cristiano de valores, a desarrollar respeto por las opiniones y derechos ajenos y adquirir el deseo de servir al prójimo sin hacer diferencias raciales, sociales o de cualquier otra índole. (Id)

Cultivar en la práctica, el aprecio por el arte y por lo bello en todas formas, reconocer que en el arte se encuentran experiencias que intiman a acercarse a la perfección y a Dios (Escuela de educación UNADECA.1998. p. 17).

Brindar oportunidad para practicar, en las horas libres, deportes sanos a manera de ejercicios, ayudarles a adquirir hábitos y prácticas que estimulen la vitalidad y la salud mental. Impartir a los estudiantes un conocimiento inteligente de los principios de gobierno, desarrollar la buena voluntad para aceptar los deberes y privilegios de la ciudadanía; cultivar en el alumno el amor sincero a

su patria y desarrollar en el un sentido de respeto, aprecio, admiración por el país donde reside. El estudiante llegará también a apreciar a sus semejantes de todas las nacionalidades como hijos de Dios. (Mejía, 2004, p.1).

Alcanzar tales objetivos representa el logro de la misión de la UNADECA, el cual es en pocas palabras preparar a hombres y a mujeres que sirvan a Dios y a la humanidad de le mejor manera, que ennoblézanle el mundo y que sean aptos para el reino de los cielos. Sobre todo porque nos ha tocado vivir en una época de decadencia espiritual y moral, es de gran importancia la creación de una institución como Universidad Adventista de Centro América.

La Universidad Adventista de Centro América fue fundada en el año 1925 como Centro Educativo en la comunidad de Pedregal, en la ciudad capital de Panamá. Actualmente, las instalaciones de UNADECA cuentan con un internado de señoritas y uno de varones, debido a que la gran mayoría de sus estudiantes son extranjeros, y no todos tienen la facilidad de vivir fuera del campus. En dichos internados hay reglamentos que cumplir por parte de los alumnos, y estos se ven involucrados en una variedad de actividades, tales como se indican en el logo de la institución: crecimiento de cuerpo, mente y alma.

2.1.5. Orígenes

“El origen de la Universidad Adventista de Centro América se remonta al año 1925, cuando la Iglesia Adventista del Séptimo Día estableció un Centro Educativo en la comunidad de las Cascadas, corregimiento de Pedregal, en la ciudad capital de Panamá. Digno de recordar al Pastor William Baxter quien propuso la fundación de una casa de estudios secundarios que gradualmente crecería y llegara a convertirse en columna y vida para la educación y formación de la entusiasta juventud centroamericana.

Con fe sencilla pero aferrada al brazo de la Omnipotencia, se da inicio a ese sueño con una significativa ceremonia en la que se mezclaron el canto y la plegaria, la emoción y el gozo, la audacia y la confianza, el ideal y el triunfo. Así nace la institución con el nombre descriptivo de: “Escuela Para Jóvenes de Habla Española”.

En 1932 el colegio se trasladó a un local propio, de tierra fértil y hermosa, ubicado en Dulce Nombre de Tres Ríos, en la provincia de Cartago, una propiedad de treinta y cuatro manzanas de terreno.

Bajo la dirección de C. F. Montgomery; en 1945 se tomó la decisión de cambiarle el nombre a la institución de “Academia Hispanoamericana Adventista” a COVAC (Colegio Vocacional Adventista de América Central). Años más tarde, el nombre de éste

colegio vuelve a ser cambiado, y fue llamado CADES (Centro Adventista de Estudios Superiores). En 1986, el CONESUP (Consejo Nacional de Enseñanza Superior Universitaria Privada) otorgó el reconocimiento a la institución como universidad, por lo que comenzó a funcionar como tal, a partir del año 1987. En 1985 se realizó un concurso para escoger el nombre de la futura universidad y ganó el de UNADECA" (Universidad Adventista de Centro América). (O. Camacho, *Una Década en UNADECA*, p. 52-54)

2.1.6. Estructura actual

2.1.6.1. Administrativa

Rectora	Dra. Herminia Perla
Vicerrector Académico	Lic. Lilian Thomas
Vicerrector Financiero	Lic. Carlos Martínez
Vicerrector Estudiantil	Elisabeth De León
Contador	Lic. Isabel Brenes

Tabla # 1: estructura administrativa

2.1.6.2. Académica

En la actualidad la UNADECA cuenta con seis escuelas.

<i>Escuela</i>	<i>Nivel académico que ofrece</i>	<i>Director</i>
Administración	Bachillerato, licenciatura, Maestría	MBA Marvin Rocha

Ingeniería	Bachillerato, licenciatura	Ing. Dodanim Castillo
Enfermería	Bachillerato, licenciatura	Lic. Julia Elena Ortiz
Educación	Bachillerato, licenciatura, maestría	Lic. Mireya de Ordóñez
Psicología	Licenciatura	Lic. Kathleen Solano
Teología	Bachillerato, licenciatura	Pr. Eugenio Vallejos

Tabla # 2: estructura académica

Autoridades de apoyo

Preceptor	Prof. Ivanhoe Sánchez
Preceptora	Lic. Mélida Avilés
Pastor de Iglesia	Pr. David Macario

Tabla # 3: autoridades de apoyo

2.1.6.3. *Auxiliar*

Vigilantes, monitores, asistentes de oficina (correctores).

2.1.6.4. *Carreras de la Universidad Adventista*

En la actualidad la UNADECA, ofrece las siguientes carreras:

<i>Escuela</i>	<i>Nivel académico</i>
Ingeniería en Sistemas Computacionales	<ul style="list-style-type: none"> ➤ Bachillerato en Ingeniería en Sistemas ➤ Licenciatura en Ingeniería en Sistemas
Administración de empresas	<ul style="list-style-type: none"> ➤ Bachillerato con énfasis en: <ul style="list-style-type: none"> ➤ Contabilidad ➤ Gerencia

	<ul style="list-style-type: none"> ➤ Licenciatura con énfasis en: <ul style="list-style-type: none"> ➤ Administración de empresas ➤ Contaduría pública ➤ Maestría con énfasis en: <ul style="list-style-type: none"> ➤ Gerencia ➤ Negocios internacionales
Enfermería	<ul style="list-style-type: none"> ➤ Bachillerato en Enfermería ➤ Licenciatura en Enfermería
Psicología	<ul style="list-style-type: none"> ➤ Licenciatura en Psicología
Educación	<ul style="list-style-type: none"> ➤ Bachillerato y licenciatura con énfasis en: <ul style="list-style-type: none"> ➤ Ciencias Naturales ➤ Educación religiosa ➤ Música ➤ Computación ➤ Español ➤ Estudios sociales ➤ Inglés ➤ Matemáticas ➤ Preescolar ➤ I y II ciclo ➤ Secretariado ➤ Maestría académica en Administración Educativa

Tabla # 4: carreras

2.2. Marco teórico

Según Luis Arroyo (*informática*, Espasa-Calpe, 1991), "los primeros en llegar a la profesión fueron los contratados para trabajar en las casas fabricantes de computadores, titulados superiores en su mayoría, que seguían un curso de

formación intensiva de un mínimo de un año. Los encargados de impartir estas enseñanzas eran empleados o expertos procedentes de otras oficinas del fabricante en el extranjero. El papel que desempeñaron estas promociones de profesionales fue decisivo, pues sobre ellas recaían tanto la formación de los clientes, nuevos usuarios de la informática, como la puesta en marcha de sus aplicaciones.

Ante el auge espectacular de la informática, empezaron a proliferar los centros privados de enseñanza a los cuales acudían miles de jóvenes atraídos por un futuro que, entonces, imaginaban brillante y bien remunerado.”

Los estudios universitarios tiene la obligación de evaluar tanto de acuerdo con el avance científico-tecnológico como con las demandas profesionales que se generan, manteniendo los principios que justifican la existencia de la propia institución universitaria. Con dicha evaluación también se pretende trabajar con los últimos conocimientos y con las herramientas más modernas y aceptadas que se demuestre su autonomía y criterios propios debidamente madurados, de forma que se garantice la incorporación del estudiante al mundo profesional con unos conocimientos actualizados y con habilidades basadas en una formación sólida.

A manera de probar las sólidas bases adquiridas en los estudios universitarios se ha decidido desarrollar un proyecto que englobe herramientas y tecnologías modernas que ayudan a la realización de sistemas robustos y factibles en la actualidad.

Para la realización de este proyecto hay varios elementos tecnológicos explícitos los cuales son:

1. Lenguajes de programación para el desarrollo y el análisis (modelado) de la WebApp.
2. Sistemas para la administración de bases de datos relacionales que permite modelar problemas reales y administrar datos dinámicamente.
3. Un servidor Web para alojar y mantener el contenido de la WebApp.
4. Tecnología de desarrollo Web para crear WebApps interactivas.

2.2.1 Lenguajes de programación

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, y un grupo de términos monosémicos (es decir, con sentido único).

2.2.1.1. PHP

PHP (Preprocesador de Hipertexto) un lenguaje de programación frecuentemente usado para el desarrollo de contenidos de sitios Web, que, junto con HTML, permite crear sitios web dinámicos.

En la actualidad hay muchos lenguajes de programación Web: Asp, Java Script, HTML, DHTML, Perl, entre otros. Pero para fines de la realización de este proyecto, se ha decidido utilizar PHP porque es un lenguaje que funciona en diversas plataformas (puede ser ejecutado en la mayoría de los sistemas operativos (Windows, Mac OS, Linux, Unix), tiene la capacidad de conexión con la mayoría de los sistemas de

administración de bases de datos que se utilizan en la actualidad (MySQL, Oracle, ODBC, etc.), tiene la capacidad de leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML, posee la capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones), posee una amplia documentación, es libre, por lo que se presenta como una alternativa de fácil acceso para todos, permite las técnicas de Programación Orientada a Objetos, Admite la creación de formularios para la web.

Por estas características PHP se convierte en el lenguaje de programación Web principal y adecuado para el desarrollo de este proyecto.

Por lo tanto se ha usado PHP para la creación de cada uno de los formularios que los módulos de matrícula y horarios de la WebApp de registro. Además de la creación de formularios, PHP se uso con el fin de introducir la información de los estudiantes y la definición de los horarios a la base de datos con el objetivo de almacenar esta valiosa información para ser usada subsiguientemente. También fue utilizado para desarrollar procesos que permiten manipular y desplegar la información a los usuarios; esto se hace a través de consultas SQL que permiten extraer la información desde la base de datos para posteriormente ser publicada en cada uno de los distintos formularios.

Ejemplo de código PHP para introducir información a la base de datos

```

<?php
## +-----+
## | 1. Creación & llamadas |
## +-----+

define ("DATAGRID_DIR", "./");
define ("PEAR_DIR", "./pear/");

require_once(DATAGRID_DIR.'datagrid.class.php');
require_once(PEAR_DIR.'PEAR.php');
require_once(PEAR_DIR.'DB.php');

## *** Inclusión del archivo que contiene las variables para la conexión
a la base de datos
include 'conexion.php';

ob_start();

$db_conn = DB::factory('mysql');
$result_conn = $db_conn-
>connect(DB::parseDSN('mysql://'.$DB_USER.':'.$DB_PASS.'@'.$DB_HOST.'/'.$DB_NAME));
if(DB::isError($result_conn)){ die($result_conn->getDebugInfo()); }

$sql = " SELECT `S`.`stud_id`, CONCAT(` `,'S`.`stud_firstname`,`
        `,`S`.`stud_middlename`,` `,'S`.`stud_lastname1`,` `,
        `S`.`stud_lastname2`) AS Student,
        `S`.`stud_dob`,
        `S`.`stud_addrlocal`,
        `S`.`stud_citylocal`,
        `S`.`stud_statelocal`,
        `S`.`stud_phonelocal`,
        `S`.`stud_addrperm`,
        `S`.`stud_cityperm`,
        `S`.`stud_stateperm`,
        `S`.`stud_countryperm`,
        `S`.`stud_phoneperm`,
        `S`.`stud_email`,
        `S`.`stud_user`,
        `S`.`stud_campus`,
        `S`.`stud_type`,
        `S`.`stud_admissionstat`,
        `S`.`stud_academicstat`,
        `U`.`user_uname`,
        `U`.`user_id`,
        `C`.`campus_id`,
        `C`.`campus_name`
FROM
`student` AS S
Inner Join `users` AS U ON `U`.`user_id` = `S`.`stud_user`
Inner Join `campus` AS C ON `C`.`campus_id` =
`S`.`stud_campus`;";

## *** Establecer las opciones necesarias e instanciar la nueva clase
$db_debug_mode = false;
$db_messaging = true;

```

```

$unique_prefix = "f_";
$datagrid = new DataGrid($debug_mode, $messaging, $unique_prefix,
DATAGRID_DIR);

## *** establecer el origen de datos con las opciones necesarias
$default_order_field = "stud_id";
$default_order_type = "DESC";
$datagrid->dataSource($db_conn, $sql, $default_order_field,
$default_order_type);

## +-----+
## | 2. Configuraciones generales: |
## +-----+

## *** establecer codificación (default: utf8/utf8_unicode_ci)
$dg_encoding = "utf8";
$dg_collation = "utf8_unicode_ci";
$datagrid->setEncoding($dg_encoding, $dg_collation);

## *** establecer el lenguaje de la interfaz (default - English)
$dg_language = "es";
$datagrid->setInterfaceLang($dg_language);

$direction = "ltr";
$datagrid->setDirection($direction);

## *** establecer los layouts: 0 - tabular(horizontal) - default,
##                               1 - columnar(vertical)

$layouts = array("view"=>0, "edit"=>1, "filter"=>1);
$datagrid->setLayouts($layouts);

## *** establecer el modo para la operaciones
("type" => ink|button|image")
## *** "byFieldValue"=>"fieldName" - hace que el campo sea un vinculo
para la pagina de edición
$modes = array(
    "add"      =>array("view"=>true, "edit"=>false, "type"=>"link"),
    "edit"    =>array("view"=>true, "edit"=>true, "type"=>"link",
                    "byFieldValue"=>""),
    "cancel"  =>array("view"=>true, "edit"=>true, "type"=>"link"),
    "details" =>array("view"=>true, "edit"=>false, "type"=>"link"),
    "delete"  =>array("view"=>true, "edit"=>true, "type"=>"image")
);
$datagrid->setModes($modes);

## *** permitir operaciones en multiples filas
$multirow_option = true;
$datagrid->allowMultirowOperations($multirow_option);
$multirow_operations = array(
    "delete" => array("view"=>true),
    "details" => array("view"=>true)
);
$datagrid->setMultirowOperations($multirow_operations);

## *** 22onfigurac la clase CSS para el grid
## *** "default" or "blue" or "gray" or "green"

```

```

$css_class = "blue";

## +-----+
## | 3. Configuraciones de exportación e impresión de datos: |
## +-----+

## *** opciones de impresión: true(default) or false
    $printing_option = true;
    $dgrid->allowPrinting($printing_option);

## *** opciones de exportación de datos: true(default) or false
    $exporting_option = true;
    $dgrid->allowExporting($exporting_option);

## +-----+
## | 4. Configuraciones de paginación y ordenación de datos: |
## +-----+

    $sorting_option = true;
    $dgrid->allowSorting($sorting_option);

    $paging_option = true;
    $rows_numeration = false;
    $numeration_sign = "N #";
    $dgrid->allowPaging($paging_option, $rows_numeration,
        $numeration_sign);

    $bottom_paging = array("results"=>true, "results_align"=>"left",
        "pages"=>true, "pages_align"=>"center",
        "page_size"=>true,
        "page_size_align"=>"right");

    $top_paging = array();
    $pages_array = array("10"=>"10", "25"=>"25", "50"=>"50",
        "100"=>"100", "250"=>"250", "500"=>"500",
        "1000"=>"1000");

    $default_page_size = 10;
    $dgrid->setPagingSettings($bottom_paging, $top_paging,
        $pages_array, $default_page_size);

## +-----+
## | 6. Configuración de los modos de vista: |
## +-----+

    $vm_table_properties = array("width"=>"75%");
    $dgrid->setViewModeTableProperties($vm_table_properties);

    $vm_columns = array(
        "stud_id" =>array("header"=>"Student ID", "type"=>"label",
            "width"=>"80px", "align"=>"left",
            "wrap"=>"nowrap", "text_length"=>"-1",
            "case"=>"normal"),
        "Student" =>array("header"=>"Student Name", "type"=>"label",
            "align"=>"left", "width"=>"80px",
            "wrap"=>"nowrap", "text_length"=>"-1",
            "case"=>"normal"),

```

```

"campus_name" =>array("header"=>"Campus", "type"=>"label",
    "align"=>"left", "width"=>"60px",
    "wrap"=>"nowrap", "text_length"=>"-1",
    "case"=>"normal"),
"stud_email" =>array("header"=>"Studen E-Mail", "type"=>"label",
    "align"=>"left", "width"=>"80px",
    "wrap"=>"nowrap", "text_length"=>"-1",
    "case"=>"normal"),
"user_uname" =>array("header"=>"Studen System User",
    "type"=>"label", "align"=>"left",
    "width"=>"80px", "wrap"=>"nowrap",
    "text_length"=>"-1", "case"=>"normal"),
);
$datagrid->setColumnsInViewMode($vm_columns);

```

```

## +-----+
## | 7. Add/Edit/Details Mode Settings: |
## +-----+

```

```

$table_name = "student";
$primary_key = "stud_id";
$condition = "";
$datagrid->setTableEdit($table_name, $primary_key, $condition);

```

```

$em_columns = array(
    "stud_id" =>array("header"=>"Student ID", "type"=>"textbox",
        "width"=>"210px", "req_type"=>"rt",
        "title"=>"Student ID"),
    "stud_firstname" =>array("header"=>"FirsName",
        "type"=>"textbox", "width"=>"210px",
        "req_type"=>"rt", "title"=>"Student
        FirsName"),
    "stud_middlename" =>array("header"=>"MiddleName",
        "type"=>"textbox",
        "width"=>"210px",
        "req_type"=>"rt",
        "title"=>"Student MiddleName"),
    "stud_lastname1" =>array("header"=>"Firts LastName",
        "type"=>"textbox", "width"=>"210px",
        "req_type"=>"rt", "title"=>"Student
        Firts LastName"),
    "stud_lastname2" =>array("header"=>"Second LastName",
        "type"=>"textbox",
        "width"=>"210px",
        "req_type"=>"rt",
        "title"=>"Student Second
        LastName"),
    "stud_dob" =>array("header"=>"Date Of Birthday" ,
        "type"=>"date", "width"=>"210px",
        "req_type"=>"rt", "title"=>"Date Of
        Birthday"),
    "stud_addrlocal" =>array("header"=>"Address Local" ,
        "type"=>"textbox",
        "width"=>"210px", "req_type"=>"rt",
        "title"=>"Address Local"),
    "stud_citylocal" =>array("header"=>"City Local" ,

```

```

        "type"=>"textbox",
        "width"=>"210px",
        "req_type"=>"rt", "title"=>"City
        Local"),
    "stud_statelocal" =>array("header"=>"State Local",
        "type"=>"textbox",
        "width"=>"210px",
        "req_type"=>"rt",
        "title"=>"State Local"),
    "stud_phonelocal" =>array("header"=>"Phone Local",
        "type"=>"textbox",
        "width"=>"210px",
        "req_type"=>"rt", "title"=>"Phone
        Local"),
    "stud_email" =>array("header"=>"Student E-Mail" ,
        "type"=>"textbox", "width"=>"210px",
        "req_type"=>"rt", "title"=>"Student
        E-Mail"),
    "stud_user" =>array("header"=>"Student User" ,
        "type"=>"textbox", "width"=>"210px",
        "req_type"=>"rt", "title"=>"Student
        User"),

    $dgrid->setColumnsInEditMode($em_columns);

    $foreign_keys = array(
        "stud_campus"=>array("table"=>"campus",
            "field_key"=>"campus_id",
            "field_name"=>"campus_name",
            "view_type"=>"dropdownlist",
            "order_by_field"=>"campus_name",
            "order_type"=>"ASC"),
        "stud_user"=>array("table"=>"users",
            "field_key"=>"user_id",
            "field_name"=>"user_uname",
            "view_type"=>"dropdownlist",
            "order_by_field"=>"user_id",
            "order_type"=>"ASC"),
    );
    $dgrid->setForeignKeysEdit($foreign_keys);

## +-----+
## | 8. Generación del DataGrid: |
## +-----+

    $dgrid->bind();
    ob_end_flush();

?>

```

Figura # 1: Código para introducir datos del alumno a la base de datos

Fragmento de código PHP que permite extraer información a la base de datos para ser mostrada en formularios.

```
<?php
## +-----+
## | Se incluye el archivo de conexión para tener acceso a la BD |
## +-----+
include 'conexion.php';

$Carne = $_POST['Carne'];

if(!empty($Carne)){
    $SelectStudent = "SELECT preregistration.pre_student,
                        student.stud_firstname,
                        student.stud_lastname1,
                        student.stud_addrlocal,
                        student.stud_phonelocal, student.stud_email
FROM
    student
    INNER JOIN preregistration ON
        (preregistration.pre_student =
         student.stud_id)
WHERE
    preregistration.pre_student = '$Carne'";

    $ResultadoStudent = mysql_query($SelectStudent) or die ('Consulta
                        fallida: ` . mysql_errno());
    while ($StudentRow = sql_fetch_array($ResultadoStudent,MYSQL_NUM)) {
        $nn=1;
        foreach ($StudentRow as $StudentId){
            if($nn==1){
                $StudentCarne = $StudentId;
            }
            elseif($nn==2){
                $StudentNombre = $StudentId;
            }
            elseif($nn==3){
                $StudentApellido = $StudentId;
            }
            elseif($nn==4){
                $StudentDireccion = $StudentId;
            }
            elseif($nn==5){
                $StudentTelefono = $StudentId;
            }
            elseif($nn==6){
                $StudentEmail = $StudentId;
            }
            $nn++;
        }
    }
}
```

```

?>
<html>
  <head>
    <title>:: Pre-Matricula ::</title>
  </head>
<body>
  <link rel="stylesheet" href="style.css" type="text/css">
  <form name="signup" id="signup" method="POST"
action="DirectorAprobacion.php">
    <table align="center">
      <tr>
        <td class="labelcell" colspan="6">
          <span class="error">
            <?php if(isset($_GET['error'])) echo $_GET['error'];?>
          </span>
        </td>
      </tr>
      <tr>
        <?php
          $SelectMatricula = mysql_query("SELECT * FROM
            registration WHERE reg_activa = '1'") or
            die ('consulta fallida' . mysql_error());

          if (mysql_num_rows($SelectMatricula) > 0) {
            echo '<td colspan="2" class="labelcell"
              align="right"><label
                for="name">Matricula 27lectr</label>
              </td>';
            echo '<td colspan="2"
              class="smallfieldcell"><select
                name="matriculaAct"
                id="cmbMatriculaAct"/>';
            while($mat =
              mysql_fetch_row($SelectMatricula)){
              echo '<option
                value="" . $mat[0] . "'>
                ' . $mat[1] . '</option>';
            }
            echo '</select></td>';
          }

          ?>
          <td colspan="2" class="labelcell">&nbsp;</td>
          <td colspan="2" class="fieldcell">&nbsp;</td>
          <td colspan="2" class="labelcell">&nbsp;</td>
          <td colspan="2" class="fieldcell">&nbsp;</td>
        </tr>
        <tr>
          <td class="labelcell"><label for="name">Numero de
            Carné</label></td>
          <td class="smallfieldcell"> <input type="text"
            name="Carne" id="txtCarne" tabindex="1"
            <?php
              if(!empty($StudentCarne))
                echo " value=\"\" . $StudentCarne . "\"";
            ?>
          /></td>
          <td colspan="2" class="labelcell"><label

```

```

        for="Nombre">Nombre</label></td>
<td colspan="2" class="fieldcell"> <input type="text"
    name="Nombre" id="txtNombre" tabindex="2"
    <?php
        if(!empty($StudentNombre))
            echo " value=\"\" . $StudentNombre . "\"";
    ?>
/></td>
<td colspan="2" class="labelcell"><label
    for="Apellido">Apellido</label></td>
<td colspan="2" class="fieldcell"> <input type="text"
    name="Apellido" id="txtApellido" tabindex="3"
    <?php
        if(!empty($StudentApellido))
            echo " value=\"\" . $StudentApellido . "\"";
    ?>
/></td>
</tr>
<tr>
<td class="labelcell">Direccion</td>
<td class="smallfieldcell"><input type="text"
    name="Dirección" id="txtDireccion" tabindex="4" /

    <?php
        if (!empty($StudentDireccion))
            echo " value=\"\" . $StudentDireccion . "\"";
    ?>
    ></td>
<td colspan="2" class="labelcell">Telefono</td>
<td colspan="2" class="fieldcell"><input type="text"
    name="Telefono" id="txtTelefono" tabindex="5"
    <?php
        if (!empty($StudentTelefono))
            echo " value=\"\" . $StudentTelefono . "\"";
    ?>
/></td>
<td colspan="2" class="labelcell">Correo
    28lectronic</td>
<td colspan="2" class="fieldcell"><input type="text"
    name="email" id="txtemail" tabindex="6"
    <?php
        if(!empty($StudentEmail))
            echo " value=\"\" . $StudentEmail . "\"";
    ?>

```

Figura # 2: Código para extraer datos del alumno de la base de datos

2.2.1.2. UML

UML (Lenguaje unificado de modelado) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, que prescribe un conjunto de notaciones, diagramas estándares para modelar sistemas orientados a objetos, y

para describe la semántica esencial de lo que estos diagramas y símbolos significan, además ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Con este fin, UML cuenta con trece tipos diferentes de diagramas, los cuales muestran diferentes aspectos de las entidades representadas; estos diagramas se dividen en tres grandes grupos:

1. **Diagramas de estructura:** estos diagramas tienen como objetivo enfatizar los elementos que deben existir en el sistema modelado:

- a. Diagrama de clases
- b. Diagrama de componentes
- c. Diagrama de objetos
- d. Diagrama de estructura compuesta
- e. Diagrama de despliegue
- f. Diagrama de paquetes

2. **Diagramas de comportamiento:** estos diagramas subrayan lo que debe suceder en el sistema modelado:

- a. Diagrama de actividades
- b. Diagrama de casos de usos
- c. Diagrama de estado

3. **Diagramas de Interacción:** un subtipo de diagramas de comportamiento, que enfatizan sobre el flujo de control y de datos entre los elementos del sistema modelado:
 - a. Diagrama de secuencia
 - b. Diagrama de comunicación
 - c. Diagrama de tiempo
 - d. Diagrama de vista de interacción

En el proceso de análisis y diseño de la WebApp, UML se convierte en un lenguaje importante ya que ofrece las herramientas (diagramas) necesarios para tener un escenario completo de cómo los usuarios interactuaran con cada uno de los componentes de la WebApp. En este proceso se diseñaron tres tipos de diagramas para definir el comportamiento que tendrá la WebApp con los usuarios:

1. *Diagramas de casos de usos:* son gráficos que esbozan una descripción coherente del comportamiento deseado y los límites del sistema y del usuario. Estos gráficos fueron utilizados para delimitar el comportamiento tanto del usuario como de la WebApp y por ende permitimos tener una amplia visión de cómo la WebApp reaccionara ante las peticiones de los usuarios.

Ejemplo de diagrama de casos de uso para el director de escuela

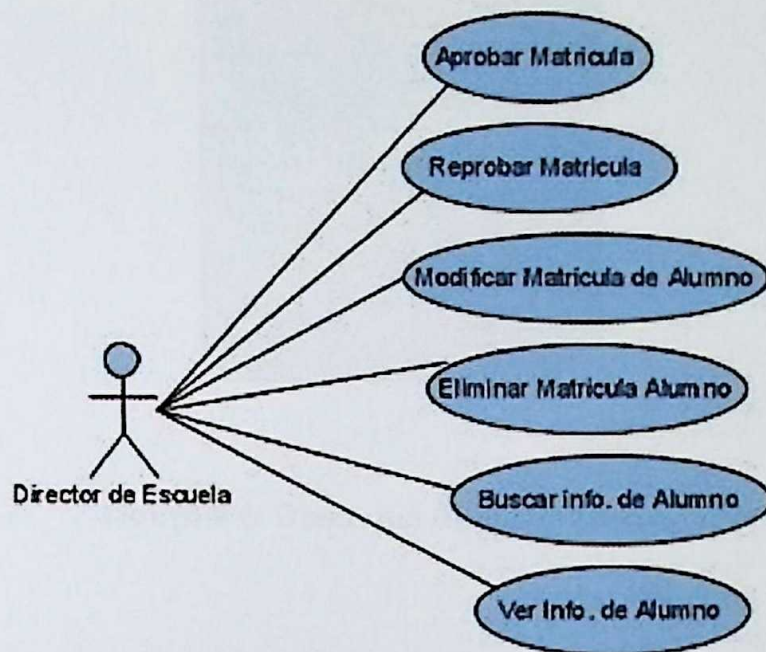


Figura # 3: Caso de uso para los directores de escuelas

2. *Diagramas de clases:* permiten crear el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre ellos. Estos diagramas fueron usados para esculpir las distintas clases (funciones) que van a permitir establecer las restricciones y los métodos que se usaran para administrar la interacción entre la WebApp y los usuarios.

Ejemplo de diagrama de clases para los alumnos

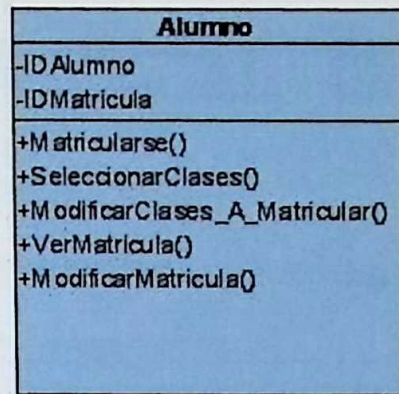


Figura # 4: Diagrama de clases para alumnos

3. *Diagramas de secuencia*: muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo, los objetos que intervienen en el escenario son colocados con líneas discontinuas verticales, y los mensajes (actividades) pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior. Estos diagramas son importantes porque nos conceden una combinación de objetos/actividades y tiempo en que estos se llevan a cabo. Estos diagramas se utilizaron para tener una organización completa de cada una de las actividades que llevaran a cabo los usuarios en la WebApp según el tiempo en que tienen lugar.

Ejemplo de diagrama de secuencia para los alumnos

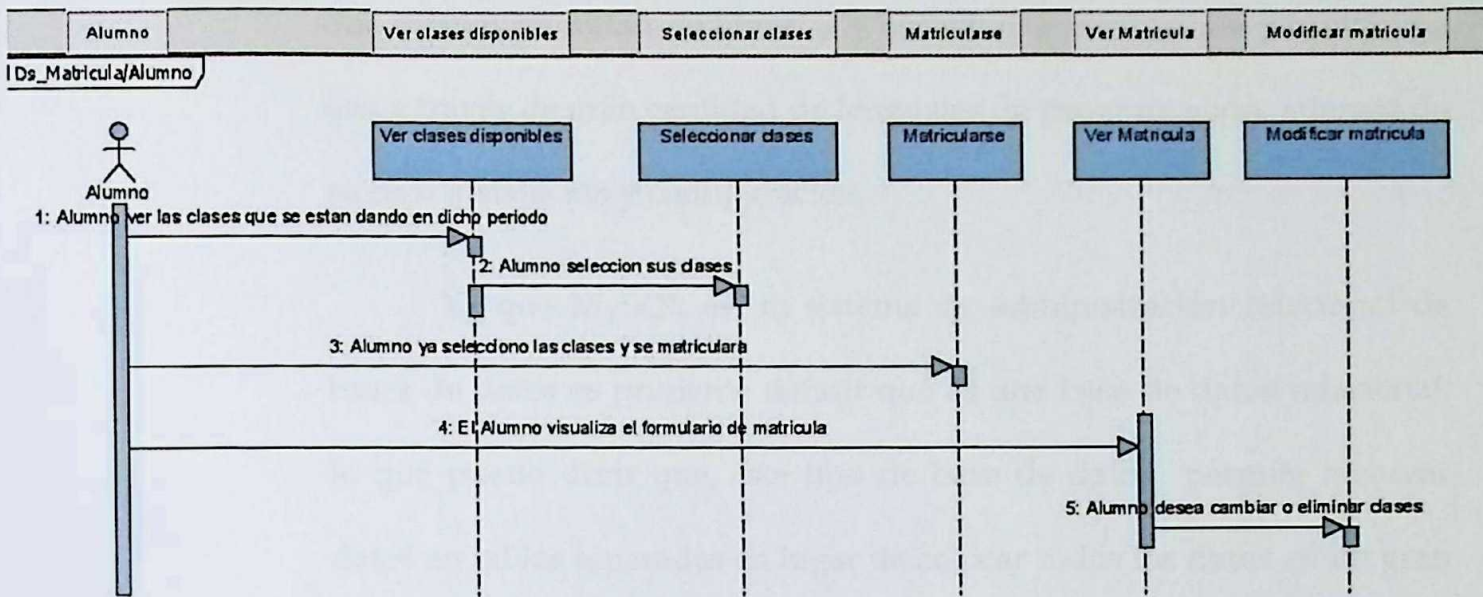


Figura # 5: Diagrama de secuencia para alumnos

2.2.2. Sistemas para la administración de bases de datos

Un sistema gestor o administrador de bases de datos (SGBD) es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, asegurando así la integridad, confidencialidad, y seguridad de los datos que se almacenan en la base de datos.

2.2.2.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional, su diseño multihilo le permite soportar una gran carga de información de manera eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran

rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Ya que MySQL es un sistema de administración relacional de bases de datos es prudente definir qué es una base de datos relacional; lo que puedo decir que, este tipo de base de datos permite archivar datos en tablas separadas en lugar de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Para el desarrollo de este proyecto se ha decidido usar MySQL como sistema para administrar la base de datos, porque en primer lugar existen varias APIs que permiten, a las aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Es un sistema que esta en constante desarrollo, ofrece un rico y muy útil conjunto de funciones.

Aprovechando todas estas facilidades que ofrece MySQL, se ha usado para:

1. *Definir la base de datos:* especificar tipos, estructuras y restricciones de datos.

2. *Construir la base de datos:* guardar los datos en algún medio controlado por el SGBD.

3. *Manipular la información que se encuentra en la base de datos:* ingresar datos, realizar consultas, actualizaciones, eliminar datos innecesarios.

4. *Controlar la redundancia:* La redundancia de datos tiene varios efectos negativos como: duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos.

5. *Restricción de los accesos no autorizados:* cada usuario debe de tener los permisos debidos para la autenticación.

6. *Cumplir las restricciones de integridad:* son las prohibiciones que se imponen para que la base de datos nunca llegue a un estado inconsistente.

2.2.3. Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (*protocolo de transferencia de hipertexto*). Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas web o páginas HTML (*lenguaje de marcas hipertextuales*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador.

2.2.3.1. *Servidor web Apache*

Apache es un servidor web gratuito, multiplataforma, potente, continuamente actualizado, adaptado a los nuevos protocolos, que ofrece un servicio estable y sencillo de administrar y configurar.

En el amplio mundo de Internet se pueden encontrar otras alternativas de servidores web, entre las cuales están, IIS (Internet Information Service) de Microsoft, Cherokee que al igual que apache es un software libre y muy eficiente, pero se ha decidido usar como servidor web apache porque funciona sobre muchas plataformas (Unix, Linux, Windows), es un servidor modular ya que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos, es extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP y MySQL.

3. *Producto y resultado*

Al final del proyecto, se pretende obtener como producto final una WebApp robusta y de fácil manejo para los usuarios tanto del departamento de registro como para los directores de escuela, profesores, alumnos, vicerrectores estudiantil y académico y administradores del sistema, con los módulos de matrícula y horario

que permitirá automatizar y agilizar el proceso de matrícula, y la manera en que maneja tan valiosa información.

4. Cobertura del proyecto

Este proyecto cubre los módulos de horarios y matrícula de la WebApp de registro de la Universidad Adventista de Centro América.

En el módulo de horarios se especifican las clases y el horario en que se estarán impartiendo dichas clases, además permite especificar el profesor que impartirá la clase y el salón en que se dará la clase a lo largo del cuatrimestre; una vez que se tiene definido el horario para la matrícula activa, el alumno puede proceder a realizar su pre-matrícula, seleccionando las clases que pretende tomar en el cuatrimestre en curso, posteriormente dicha pre-matricula será revisada por el director de escuela y aprobada si el estudiante puede llevar las clases seleccionadas; el vicerrector estudiantil, vicerrector financiero, y director de registro podrán ya sea aprobar o reprobar una matrícula en base a la situación estudiantil, financiera o de registro del estudiante.

Capítulo 2: Análisis de requerimientos de software

Los requerimientos de un sistema de software, cuando se ven en su conjunto son extensos y detallados, y además contienen múltiples relaciones entre si. Con el análisis de requerimientos se obtiene la

posibilidad de especificar sistemas complejos al documentar especificaciones simples y concisas para el sistema. Esto se logra mediante tres procesos:

- Clasificación
- Estructuración y
- Organización

De todo lo que el sistema debe de hacer.

El análisis de requerimientos consiste brevemente en los siguientes pasos:

- Obtener información acerca de lo que los usuarios desean.
- Clasificar esos deseos para comenzar a estructurar requerimientos
- Identificar los niveles de jerarquía del sistema y empezar a alojar los ya clasificados requerimientos en cada nivel.
- Especificar formalmente los requerimientos de acuerdo al nivel de audiencia que se desea.

Para el desarrollo de este proyecto, la etapa de análisis de requerimiento fue entregada por el departamento de registro de la UNADECA los cuales son:

Abstracto

El requerimiento número uno para la creación del sistema de registro de la Universidad Adventista de Centro América, es el de “informatizar” el proceso de matrícula de dicha institución. Por el contrario, otro de los requerimientos de este sistema es de no cambiar el proceso de matrícula de la UNADECA. Dada la peculiaridad de nuestra institución y de acuerdo con reglamentos y directrices creadas en la División Interamericana, el proceso de matrícula no se debe de cambiar y se debe de mantener como está.

Detalles

Para poder cumplir con los objetivos y requerimientos principales del sistema es importante comprender cuál es el proceso de matrícula de un estudiante para UNADECA. El proceso de matrícula tiene sus inicios en la Vice-rectoría Académica y tiene involucrado un gran número de personas y oficinas que aportan al proceso. En resumen abajo encontramos los pasos de una matrícula.

1. Creación de horario: un horario nace en la oficina de vice-rectoría académica. En este horario se especifican las clases que se impartirán, quien las impartirá, en donde y a qué hora.

2. Partiendo de dicho horario el estudiante tendrá una idea de las clases que puede llevar y puede así llenar su matrícula.
3. Seguidamente, el estudiante pasa con el director de escuela y se hace el proceso de llenar la matrícula en la cual se escogen las clases a tomarse y el director de escuela aprueba esta matrícula. El director de escuela aprobará la matrícula de este estudiante de acuerdo a las reglas de la institución y la política interna de la escuela.
4. En el siguiente paso, el estudiante debe de conseguir que las oficinas de vice-rectoría financiera y vice-rectoría estudiantil aprueben su matrícula. Estas oficinas aprobarán la matrícula de este estudiante de acuerdo a las reglas de la institución y la política interna de estos departamentos.
5. El último paso pide que el estudiante consiga la aprobación de la oficina de Registro a su matrícula. Esta oficina aprobará la matrícula del estudiante de acuerdo a las reglas de la institución y la política interna de este departamento.
6. El estudiante tiene, por iniciativa propia o por que se la ha requerido, la oportunidad de hacer cambios a la matrícula; al realizar los cambios, el proceso de aprobación de matrícula de las cuatro entidades correspondientes

(director de escuela, finanzas, asuntos estudiantiles y registro) tiene que repetirse.

7. El estudiante también tiene la oportunidad de hacer pedidos académicos de acuerdo a las necesidades individuales de cada uno y estos se estudiarán y se resolverán de acuerdo a políticas internas del departamento.

Restricciones

Se identificaron un número de restricciones para el sistema unadecano de registro, las cuales se detallan a continuación.

1. El sistema debe de ser flexible y expansible
2. El sistema debe de ser flexible para que se pueda manejar más de una institución
3. El sistema debe de ser flexible para poder manejar más de un campus
4. El sistema debe de funcionar internamente pero es el deseo de la institución convertirlo en un sistema que funcione en línea
5. El sistema debe de permitir el ingreso de matrículas históricas y notas históricas de los estudiantes.
6. El sistema debe de permitir que un estudiante termine más de un título

7. El sistema no debe de cambiar el proceso solo agilizarlo e informatizarlo
8. El sistema debe de diseñarse de forma expansible para en un futuro, añadir otros módulos como contabilidad
9. El sistema debe de tener una interfaz fácil de usar

Requerimientos

Para transformar el proceso actual de matrícula en un proceso informático se han desarrollado los siguientes requerimientos

1. La creación de diferentes niveles de autenticación y autorización para los diferentes usuarios del sistema. Para poder utilizar el sistema, el usuario tiene que tener una cuenta activa del sistema y un nivel de permiso, o sea el usuario tiene que autenticarse antes de ingresar al sistema. No todos los recursos del sistema están disponibles para todos los usuarios sino que un usuario en específico tiene que estar autorizado para utilizar dicho recurso.
2. Los diferentes tipos de usuarios identificados son
 - a. Estudiantes
 - b. Profesores
 - c. Directores de Escuela
 - d. Finanzas

- e. Asuntos Estudiantiles
 - f. Registro
 - g. Administradores
3. Los administradores podrán hacer uso de las operaciones CLEB (esto significa Crear Leer Editar y Borrar) para cada uno de los siguientes componentes
- a. Instituciones
 - b. Campus
 - c. Facultades
 - d. Profesores
 - e. Estudiantes
 - f. Usuarios
 - g. Títulos
 - h. Planes de Estudio
 - i. Horarios
4. Un campus pertenece a una institución, una institución puede tener más de un campus.
5. Una facultad (escuela) pertenece a una institución y no a un campus ya que las facultades trascienden los campus pero no las instituciones. Por consiguiente una institución puede tener muchas facultades.

6. Una facultad puede impartir muchas clases y cada clase pertenece a una sola facultad. Las clases se identifican por un código, un nombre, la cantidad de créditos, si es electiva o no y una descripción.
7. Una facultad puede tener múltiples programas de estudio y cada programa de estudio pertenece a una sola facultad. Por ejemplo, la facultad de Administración tiene los siguientes programas: Administración de Empresas, Gerencia, Contabilidad, Contaduría Pública y Negocios Internacionales.
8. Cada programa puede ofrecer diferentes títulos de dicho programa de estudio cada título pertenece a solo un programa de estudio. Por ejemplo, el programa de estudio Administración de Empresas ofrece los títulos de Bachiller y Licenciatura.
9. Planes de Estudio: Cada PDE estará identificado por un nombre, una descripción, un título (de un programa de estudio de una facultad), el año en que se creó, y una estructura de tiempo.
10. Una estructura de tiempo está dividida en divisiones (Año I de Bachillerato por Semestres) y las divisiones están divididas en períodos (Semestre 3 del Año I de Bachillerato por Semestres). Esto permite que se les aplique diferentes estructuras de

tiempo a los PDEs por campus. Por ejemplo, en campus 1 se enseña tres cuatrimestres al año, en el campus 2 se enseña por cuatro trimestres al año y los dos campus utilizan su propio PDE para el mismo título.

11. Se debe de permitir el ingreso de diferentes clases a los diferentes períodos del PDE. Por ejemplo, en el cuatrimestre 1 del primer año de bachillerato por cuatrimestres se llevan las clases: Vida y Enseñanza de Jesús, Educación I y Matemática I.
12. Se debe de permitir el ingreso de nuevos estudiantes
13. Al ingresar un estudiante el sistema debe de automáticamente crear su número de carné que consisten en los últimos dos números del año de ingreso del estudiante y un número indicando el momento de ingreso. Por ejemplo, el vigésimo quinto estudiante que ingresó en el 2004, tendrá un número de carné de 04-0025.
14. Al estudiante se le asigna un plan de estudio el cual indicará los requerimientos a terminar para obtener el título asociado a ese plan de estudio.
15. Se debe de permitir el ingreso de nuevos profesores
16. Un profesor pertenece a una facultad y una facultad tiene múltiples profesores. Un profesor puede enseñar diferentes

clases de diferentes facultades pero solo pertenece a una facultad.

17. Cada facultad tiene su director que tiene que ser un profesor que pertenezca a esa facultad.

18. Se debe de permitir la creación de horarios para Matrícula.

Estos horarios se describen por las diferentes clases que se pueden impartir, el horario de estas clases, el salón en donde se impartirá la clase, el profesor que impartirá la clase y la sección de la clase.

19. Una matrícula se identifica por las clases de un horario que se le aprobó a un estudiante en específico.

Acciones

Cada tipo de usuario podrá realizar diferentes acciones de acuerdo con su nivel de autorización. Detallamos estas acciones a continuación.

Estudiantes

1. Ver noticias
2. Ver calendario
3. Ver planes de estudio de la institución
4. Ver su plan de estudio

5. Ver notas de Matrículas anteriores
6. Pre matricularse
7. Hacer cambios a matrícula
8. Hacer pedidos académicos

Profesores

1. Ver noticias
2. Ver calendario
3. Ver planes de estudio
4. Ver notas de clases impartidas en matrículas anteriores
5. Ver estudiantes matriculados en sus clases
6. Establecer notas en el tiempo establecido para ello

Directores de Escuela

1. Ver noticias
2. Ver calendario
3. Ver planes de estudio
4. Ver notas de clases impartidas en matrículas anteriores
5. Ver estudiantes matriculados en sus clases
6. Establecer notas en el tiempo establecido para ello
7. Ver planes de estudio de todos los estudiantes de la escuela
8. Ver notas de todos los estudiantes de la escuela

9. Aprobar matricula

Vice-rectoría Financiera

1. Ver noticias
2. Ver calendario
3. Ver planes de estudio
4. Aprobar Matricula

Vice-rectoría Estudiantil

1. Ver noticias
2. Ver calendario
3. Ver planes de estudio
4. Aprobar Matricula

Registro

1. (Todo lo de un director de escuela pero a nivel de institución)
2. Imprimir reportes

Administradores

1. (Como se mencionó en los requerimientos, operaciones CLEB para todo lo relacionado con una institución)
2. (Todo lo de un miembro del grupo de permisos 2Registro"

Procesos

A continuación se detallan algunos de los procesos más importantes que ocurren en el sistema URECON.

Matricula

El estudiante ingresa al sistema y se autentica, puede de allí ver su plan de estudio para ver cuáles son las clases que le hacen falta. Seguidamente, con esta información, el estudiante puede seleccionar la clases que quiere tomar en la matricula activa. De allí, el estudiante visita al director de escuela que puede ver la pre-matricula del estudiante y determinar si lo que ha seleccionado está correcto. Si todo está bien, el director puede aprobar la matricula o hacer cambios a la mismo y después aprobarla.

De allí el estudiante pasa a vice-rectoría financiera en donde se hace la consulta financiera correspondiente y se aprueba o no la matrícula. Si se aprueba el estudiante pasa a vice-rectoría estudiantil y si no, tiene que regresar al director de escuela para hacer los ajustes necesarios. Los cambios que se realizan a la matricula quedan guardados.

De allí el estudiante pasa a vice-rectoría financiera en donde se hace la consulta estudiantil correspondiente y se aprueba o no la

matrícula. Si se aprueba el estudiante pasa a registro y si no, tiene que regresar al director de escuela para hacer los ajustes necesarios. Los cambios que se realizan a la matrícula quedan guardados.

Por último registro revisa la matrícula y decide si aprobar o no la matrícula, si hay que realizarse cambios, el estudiante tendrá que dirigirse de nuevo al director de escuela para empezar el proceso desde allí. Los cambios que se realizan a la matrícula quedan guardados.

Procesos del Profesor

El profesor, puede entrar en el sistema y ver la lista de estudiantes matriculados para las clases que va a impartir. En el tiempo adecuado se habilita la opción de ingresar notas para el profesor y este ingresará las notas de cada una de sus clases. Seguidamente pasará por la oficina de registro para firmar cada una de las planillas ya existentes. Al ingresar la nota, ésta pasará al plan de estudio del estudiante como también quedará registrada como la nota de esa clase para esa matrícula.

Capítulo 3: análisis y diseño de sistemas

3.1 *Análisis y diseño del sistema*

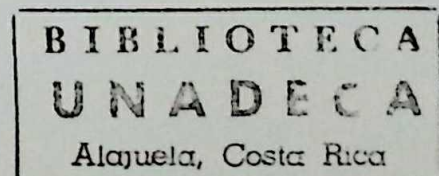
Esta etapa se puede definir como: etapa que permite alcanzar un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. También es la etapa que permite lograr un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información. Esto se lleva a cabo teniendo en cuenta ciertos principios:

- Debe presentarse y entenderse la información de un problema.
- Definir las funciones que debe realizar el software.
- Representar el comportamiento del software.
- Divida en forma jerárquica los módulos que representan la información, funciones y comportamientos.

Para el análisis y diseño de este proyecto se ha implementado el lenguaje grafico UML con el fin de modelar cada parte (módulo de matrícula y módulo de horario) de dicho proyecto. Los diagramas que se obtuvieron del modelaje UML son los siguientes:

3.2 *Diagramas de casos de usos*

Estos diagramas permiten tener una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular.



1601.42
B212 Q11

3.2.1 Casos de usos para el módulo de horario

Dado que el vicerrector académico es el encargado de definir los horarios de las clases de todas las escuelas, la WebApp debe permitirle tener el control total del módulo de horarios, la siguiente figura muestra las acciones que este actor puede realizar.

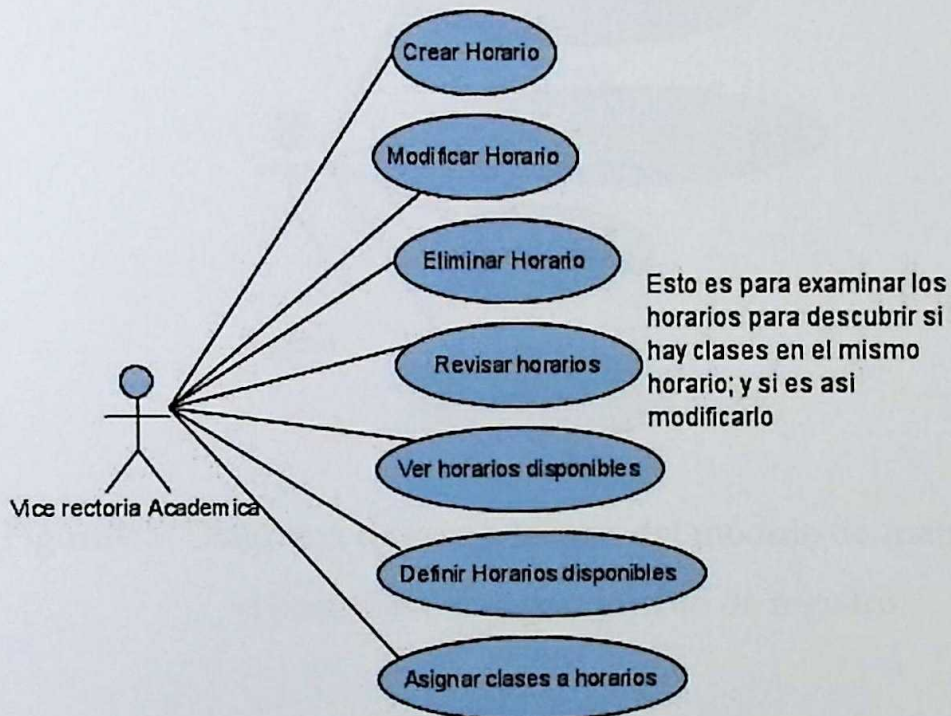
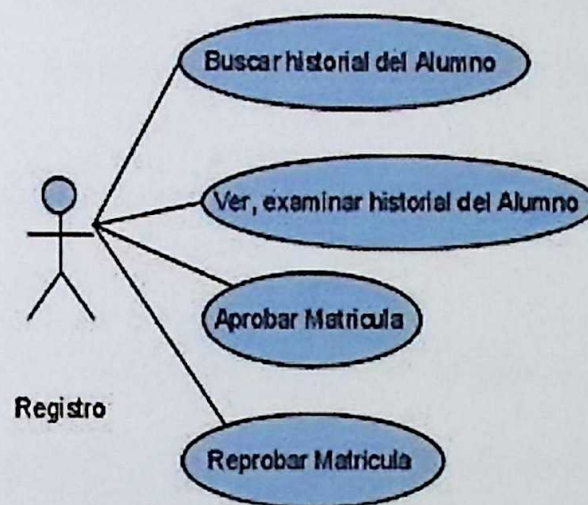


Figura #6: Diagrama de casos de uso del módulo de horario para el vicerrector académico.

3.2.2 Casos de usos para el módulo de matrícula

3.2.2.1 Personal del departamento de registro

El personal del departamento de registro tiene la posibilidad de visualizar toda la información de matrícula de los alumnos, además de aprobar o rechazar dicha matrícula, por lo tanto la WebApp debe posibilitarles realizar estas acciones. En la siguiente figura se muestran dichas acciones.



Figura# 7: Diagrama de casos de usos del módulo de matrícula para el personal del departamento de registro

3.2.2.2 *Vice rectoría financiera/estudiantil*

Estos usuarios básicamente lo que necesitan saber es la información financiera, para el vicerrector financiero y estudiantil para el vicerrector estudiantil de cada alumno, a partir de esta información dichos usuarios pueden aprobar o reprobar la matrícula.

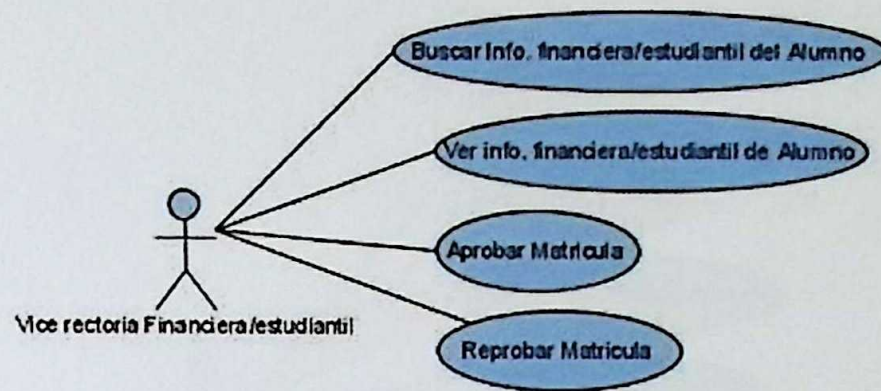


Figura # 8: Diagrama de casos de uso del módulo de matrícula para el vicerrector financiero/estudiantil

3.2.2.3 *Director de escuela*

La WebApp debe brindar a los directores de escuela la posibilidad de ver la información de los alumnos, esta información básicamente es: datos personales, posibles clases a matricular, matrícula. Este usuario puede manipular la información antes mencionada en el momento que estime conveniente.

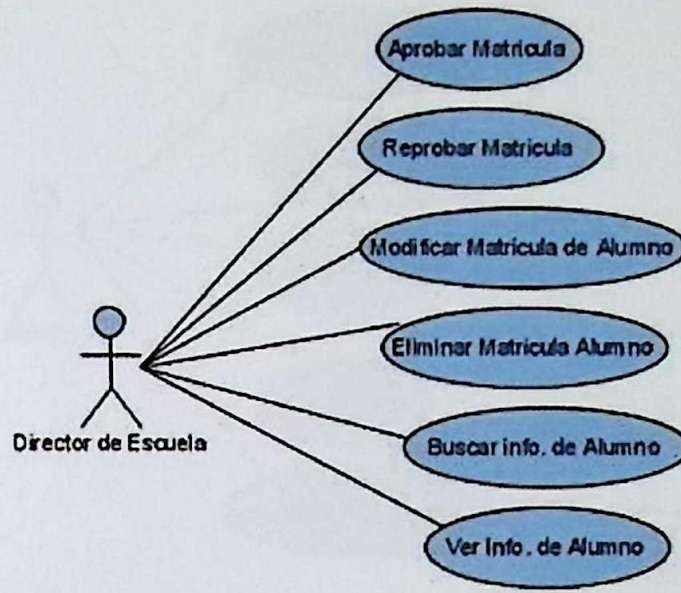


Figura # 9: Diagrama de casos de usos del módulo de matrícula para los directores de escuelas

3.2.2.4 Alumno

Los alumnos en primera instancia deben de ver las clases disponibles (horario) para posteriormente seleccionar las clases y demás información relacionada con el horario de clases, para seguir con el proceso de matrícula. Este usuario también puede ver y modificar la información de la matrícula.

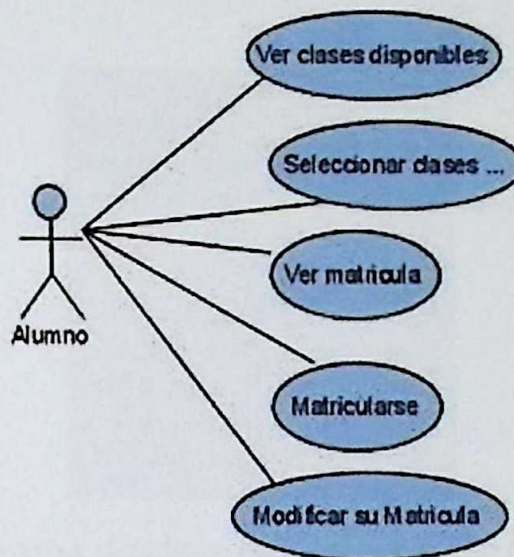


Figura # 10: Diagrama de casos de usos del módulo de matrícula para los alumnos

3.3 Diagramas de clases

Dichos diagramas describen la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Estos permiten crear un diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

3.3.1 Clase para el módulo de horario

Esta es la clase que permite la manipulación de los horarios. En la siguiente figura se muestran los atributos (IDHorario,

NombreHorario, Descripción) y operaciones (CrearHorario (), ModificarHorario (), EliminarHorario (), etc.) de dicha clase.

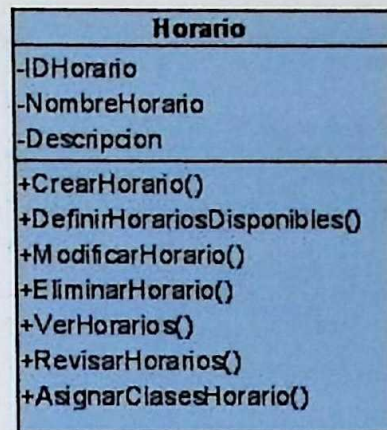


Figura # 11: Diagrama de clase para el módulo de horario

3.3.2 Clase para el módulo de matrícula

3.3.2.1 Personal de registro

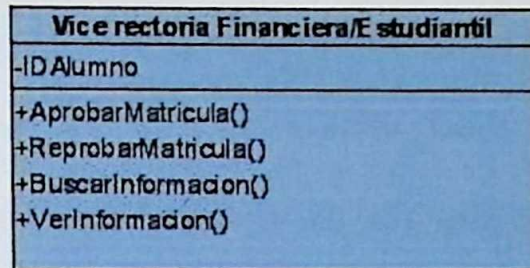
El personal de registro puede realizar las operaciones que se muestran en la siguiente figura, tomando como valores principales los Id's del alumno, matrícula y clase.



Figura # 12: Diagrama de clase para el módulo de matrícula, usuario: personal de registro

3.3.2.2 *Vice rectoría financiera/estudiantil*

Estos usuarios podrán buscar, visualizar, aprobar o reprobar una matrícula con el Id del alumno.



Figura# 13: Diagrama de clase para el módulo de matrícula, usuario: vice rectoría financiera/estudiantil

3.3.2.3 *Directores de escuela*

Este diagrama define el atributo principal por el cual el director de escuela podrá acceder a la información de matrícula del alumno y las operaciones que dicho usuario puede llevar a cabo.



Figura # 14: Diagrama de clase para el módulo de matrícula, usuario: directores de escuelas

3.3.2.4 Alumnos

La siguiente figura muestra los atributos y operaciones que componen la clase de alumno. El alumno podrá llevar a cabo todas estas operaciones mediante su numero de identificación (número de carné); el identificador de matrícula, determinará a cual de las diferentes matriculas el alumno está registrado.

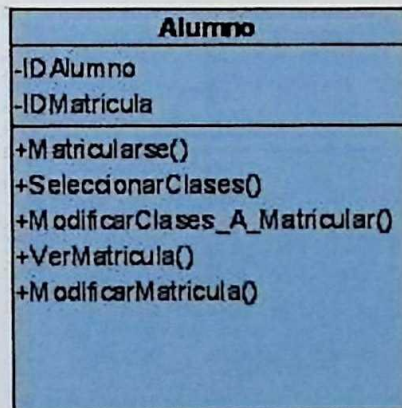


Figura # 15: Diagrama de clase para el módulo de matrícula, usuario: alumnos

3.4 Diagramas de secuencia

Estos diagramas muestran las interacciones entre los objetos organizados en una secuencia temporal. En particular muestra los

objetos participantes en la interacción y la secuencia de mensajes intercambiados.

3.4.1 Diagrama de secuencia para el módulo de horario

Este diagrama muestra la secuencia de acciones que debe llevar el vicerrector académico para realizar las diversas operaciones relacionadas con los horarios.

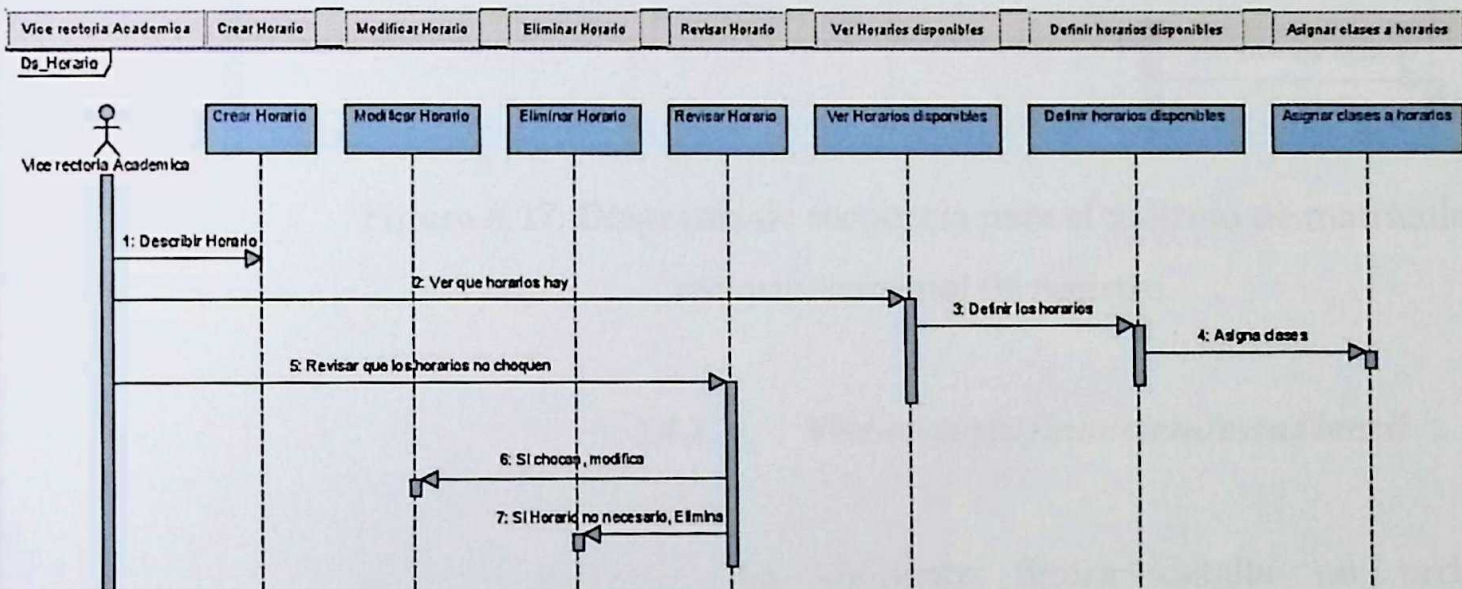


Figura # 16: Diagrama de secuencia para el módulo de horario

3.4.2 Diagrama de secuencia para el módulo de matrícula

3.4.2.1 Personal de registro

Este diagrama traza los eventos o la interacción de objetos que tendrá que llevar a

acabo el personal del departamento de registro para completar el proceso de matrícula.

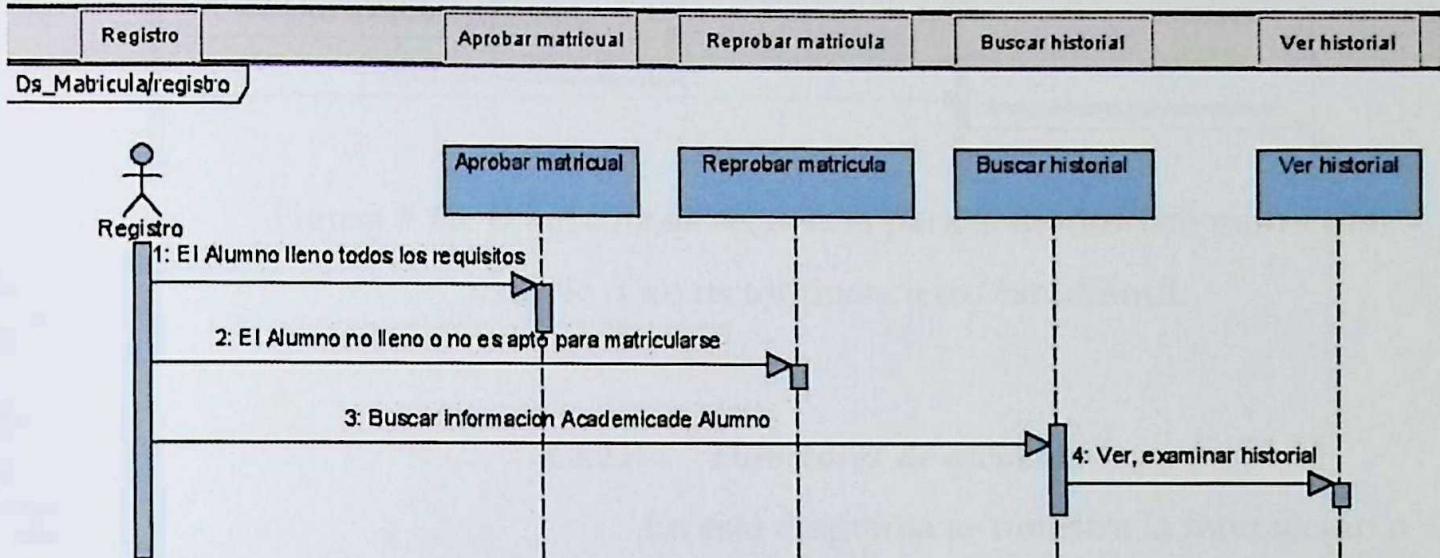


Figura # 17: Diagrama de secuencia para el módulo de matrícula, usuario: personal de registro

3.4.2.2 *Vice rectoría financiera/estudiantil*

La siguiente figura detalla en orden cronológico las acciones que debe seguir dicho usuario para comprobar el estado financiero/estudiantil del alumno y aprobar o reprobar una matrícula.

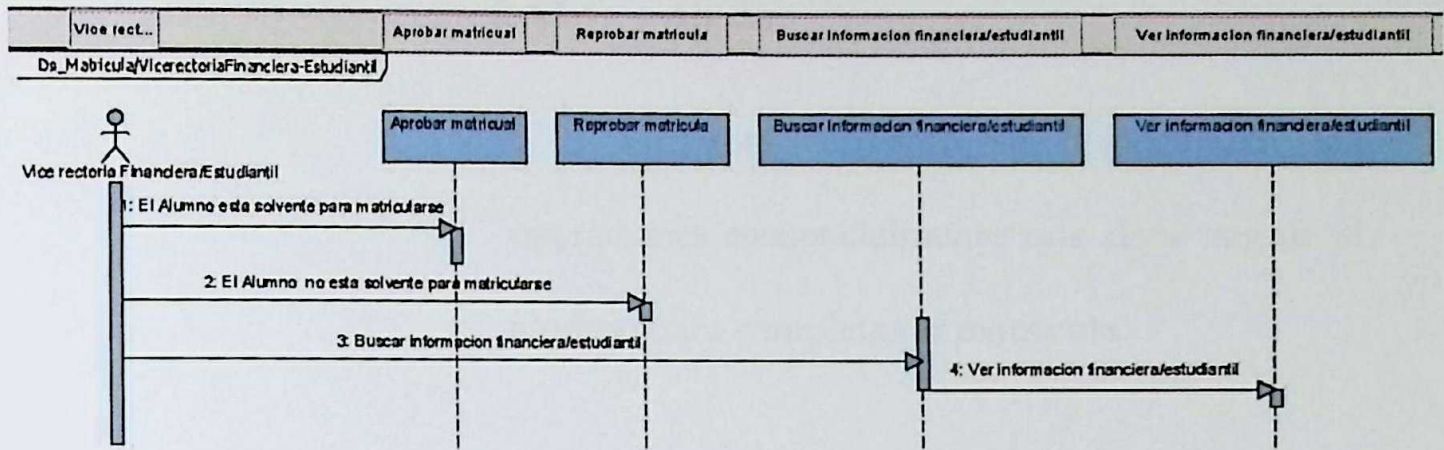


Figura # 18: Diagrama de secuencia para el módulo de matrícula, usuario: vice rector financiero/estudiantil.

3.4.2.3 Directores de escuelas

En este diagrama se muestra la interacción o el conjunto de acciones en orden progresivo que efectúan los directores de escuela con la WebApp.

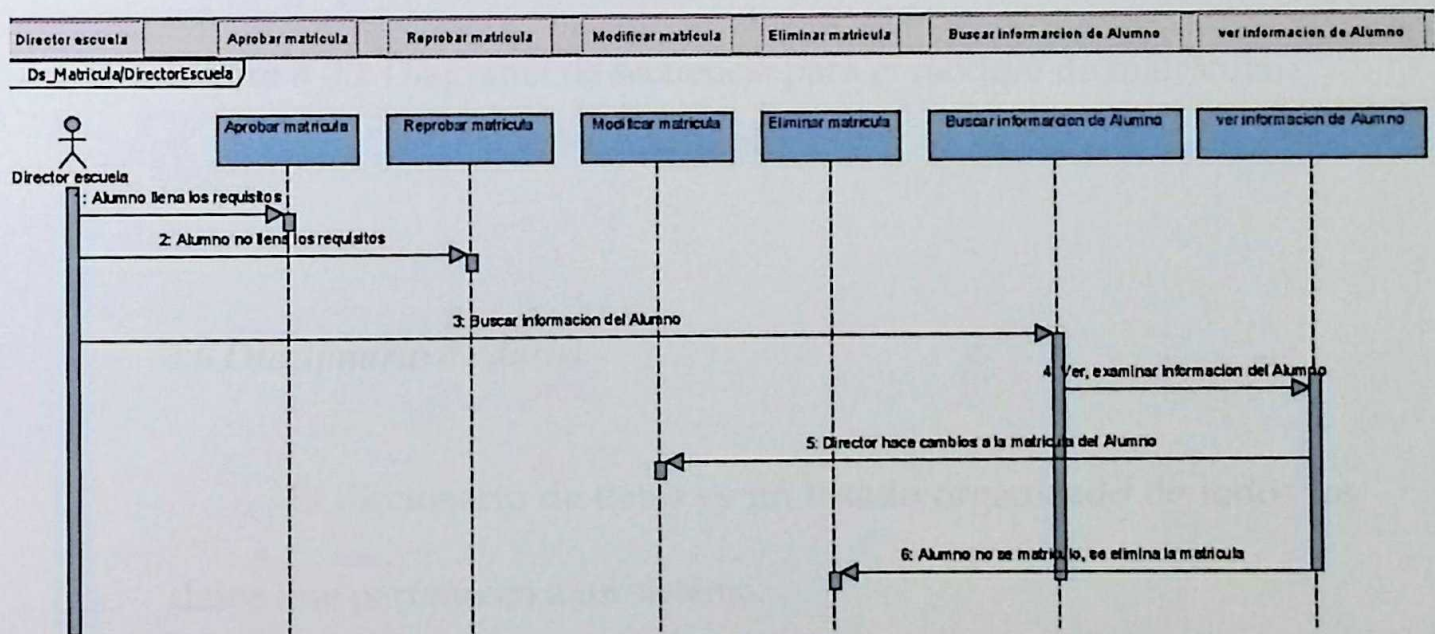


Figura # 19: Diagrama de secuencia para el módulo de matrícula, usuario: director de escuela

3.4.2.4 Alumno

En esta imagen se especifican las operaciones secuencialmente que debe seguir el alumno para completar la matrícula.

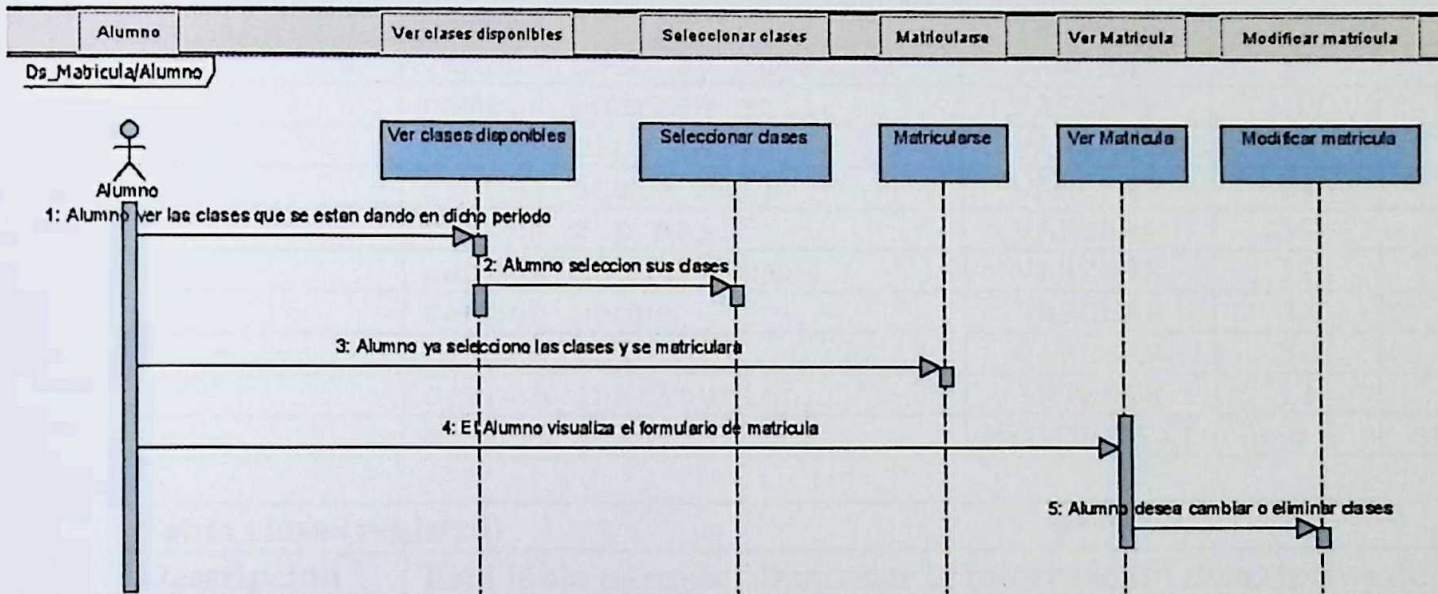


Figura # 20: Diagrama de secuencia para el módulo de matrícula, usuario: alumno

3.5 Diccionario de datos

El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema.

El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

Tabla campus (Registro)				
Descripción	Tabla que almacena los datos de los diferentes campus que puede tener una institución			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	campus_id	INTEGER	11	✓
	campus_name	VARCHAR	128	
	campus_abbr	VARCHAR	8	
	campus_address_physical	VARCHAR	128	
	campus_city_physical	VARCHAR	30	
	campus_state_physical	VARCHAR	64	
	campus_zip_physical	VARCHAR	32	
	campus_country_physical	VARCHAR	32	
	campus_address_mail	VARCHAR	128	
	campus_city_mail	VARCHAR	64	
	campus_state_mail	VARCHAR	64	
	campus_zip_mail	VARCHAR	32	
	campus_country_mail	VARCHAR	32	
	campus_phone	VARCHAR	32	
	campus_fax	VARCHAR	32	
	campus_institution	INTEGER	11	
	campus_desc	MEDIUMTEXT	0	

Tabla class (registro)				
Descripción	Esta tabla permite almacenar la información descriptiva de la clase mas la información de la facultad a la que pertenece			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	class_id	BIGINT	20	✓
	class_code	VARCHAR	8	✓
	class_name	VARCHAR	64	✓
	class_faculty	INTEGER	11	
	class_credits	SMALLINT	6	
	class_elective	TINYINT	4	
	class_desc	TEXT	9	

Tabla classdates (registro)				
Descripción	En esta tabla se definen el comienzo y fin del periodo de matrícula, matrícula extraordinaria, clases			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	cd_id	BIGINT	20	✓
	cd_regstart	DATE	0	
	cd_regend	DATE	0	
	cd_regspecialstart	DATE	0	

	cd_regspecialend	DATE	0	
	cd_classtart	DATE	0	
	cd_classend	DATE	0	
	cd_notasstart	DATE	0	
	cd_classend	DATE	0	

Tabla day (registro)				
Descripción	Tabla que permite almacenar los días de estudio			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	day_id	INTEGER	11	✓
	day_name	VARCHAR	25	

Tabla degree (registro)				
Descripción	Almacena los diferentes grados académicos			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	degree_id	INTEGER	11	✓
	degree_name	VARCHAR	128	
	degree_desc	TEXT	0	
	degree_program	INTEGER	11	

Tabla division (registro)				
Descripción	Se guardan las divisiones de tiempo de los grados académicos			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	div_id	INTEGER	11	✓
	div_name	VARCHAR	48	
	div_desc	TEXT	0	
	div_abbr	VARCHAR	8	
	div_ts	INTEGER	11	

Tabla faculty (registro)				
Descripción	Tabla que permite guardar los datos de las facultades			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	faculty_id	INTEGER	11	✓
	faculty_name	VARCHAR	64	
	faculty_director	INTEGER	11	
	faculty_description	MEDIUMTEXT	0	

	faculty_abbr	VARCHAR	16	
	faculty_institution	INTEGER	11	

Tabla institution (registro)

Descripción Almacena la información de las instituciones

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	institution_id	INTEGER	11	✓
	institution_name	VARCHAR	90	
	institution_president	VARCHAR	64	
	institution_academics	VARCHAR	64	
	institution_financial	VARCHAR	64	
	institution_student	VARCHAR	64	
	institution_desc	MEDIUMTEXT	0	
	institution_website	VARCHAR	128	
	institution_abbr	VARCHAR	0	

Tabla pde (registro)

Descripción Guarda los datos de los planes de estudio

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	pde_id	BIGINT	20	✓
	pde_name	VARCHAR	64	
	pde_desc	TEXT	0	
	pde_degree	INTEGER	11	
	pde_ts	INTEGER	11	
	pde_year	INTEGER	11	

Tabla pdecpc (registro)

Descripción En esta tabla se almacena y relaciona la información sobre los planes de estudio con clases y sus periodos

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	pdecpc_pde	BIGINT	11	✓
✓	pdecpc_class	BIGINT	11	✓
✓	pdecpc_period	INTEGER	11	✓

Tabla pdecs (registro)				
Descripción	En esta tabla se almacena y relaciona la información sobre los planes de estudio con clases y estudiantes			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	pdec_pde	BIGINT	20	✓
✓	pdecs_class	BIGINT	20	✓
✓	pdecs_stud	VARCHAR	16	✓
	pdecs_grade	INTEGER	11	

Tabla period (registro)				
Descripción	Cantidad de periodos en que se divide un plan de estudio			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	period_id	INTEGER	11	✓
	period_name	VARCHAR	48	
	period_desc	TINYTEXT	0	
	period_abbr	VARCHAR	12	
	period_div	INTEGER	11	

Tabla preregistration (registro)				
Descripción	Tabla para almacenar la información de pre matrícula de los estudiantes			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	pre_id	INTEGER	11	✓
✓	pre_student	VARCHAR	20	✓
✓	pre_registration	BIGINT	20	✓
✓	pre_class	BIGINT	20	✓
✓	pre_section	INTEGER	11	✓
	pre_room	INTEGER	11	
	pre_teacher	INTEGER	11	
	pre_schedule	INTEGER	11	

Tabla prereq (registro)				
Descripción	Tabla en donde se guardan los prerrequisitos de clases			
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	prereq_id	BIGINT	20	✓
	prereq_class	BIGINT	20	
	prereq_req	BIGINT	20	

Tabla professor (registro)				
Descripción	Tabla en donde se almacenan los datos de los profesores			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	prof_id	INTEGER	11	✓
	prof_identification	VARCHAR	64	
	prof_abbr	VARCHAR	4	
	prof_firstname	VARCHAR	64	
	prof_midname	VARCHAR	64	
	prof_lastname1	VARCHAR	64	
	prof_lastname2	VARCHAR	64	
	prof_address	VARCHAR	160	
	prof_email	VARCHAR	128	
	prof_phone1	VARCHAR	32	
	prof_phone2	VARCHAR	32	
	prof_religion	INTEGER	11	
	prof_faculty	INTEGER	11	
	prof_user	BIGINT	20	

Tabla program (registro)				
Descripción	Se coleccionan los programas de estudio y una referencia a la facultad a la que pertenecen			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	program_id	INTEGER	11	✓
	program_name	VARCHAR	96	
	program_desc	TEXT	0	
	program_faculty	INTEGER	11	
	program_abbr	VARCHAR	16	

Tabla registration (registro)				
Descripción	Tabla para almacenar los datos de las diferentes matriculas			
<i>Campos</i>				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	reg_id	BIGINT	11	✓
	reg_name	VARCHAR	64	
	reg_desc	MEDIUMTEXT	0	
	reg_year	INTEGER	11	
	reg_timestruct	INTEGER	11	
	reg_status	INTEGER	11	
	reg_classdates	BIGINT	20	
	reg_campus	INTEGER	11	

	reg_activa	CHAR	4	
--	------------	------	---	--

Tabla registration_class (registro)				
Descripción		En esta tabla se almacenan los horarios de cada matrícula		
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	regcl_registration	BIGINT	20	✓
✓	regcl_class	BIGINT	20	✓
✓	regcl_seccion	INTEGER	11	✓
	regcl_schedule	INTEGER	20	
	regcl_room	INTEGER	11	
	regcl_professor	INTEGER	20	

Tabla registration_student (registro)				
Descripción		Guarda la información de los estudiantes matriculados		
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	regst_id	INTEGER	11	✓
✓	regst_registration	BIGINT	20	✓
✓	regst_student	VARCHAR	16	✓
	regst_facultyapproved	SMALLINT	6	
	regst_financeapproved	SMALLINT	6	
	regst_studentapproved	SMALLINT	6	
	regst_regisapproved	SMALLINT	6	
	regst_facultycomment	MEDIUMTEXT	0	
	regst_financecomment	MEDIUMTEXT	0	
	regst_studentcomment	MEDIUMTEXT	0	
	regst_registrationcomment	MEDIUMTEXT	0	
	regst_status	SMALLINT	6	

Tabla regstcl (registro)				
Descripción		Tabla de matrícula		
Campos				
Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	student_autoid	INTEGER	11	✓
✓	student_id	VARCHAR	16	✓
✓	reg_id	BIGINT	20	✓
✓	regcl_id	BIGINT	20	✓
	Nota	INTEGER	11	
	Classtype	SMALLINT	6	
	tipo	SMALLINT	6	
	observaciones	MEDIUMTEXT	0	

	regstcl_change	SMALLINT	6	
--	----------------	----------	---	--

Tabla room (registro)

Descripción Se guardan los salones de estudio

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	room_id	INTEGER	11	✓
	room_name	VARCHAR	32	
	room_code	VARCHAR	16	
	room_desc	TINYTEXT	0	
	room_campus	INTEGER	11	

Tabla schedule (registro)

Descripción Se guardan la información de los horarios

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	sched_id	INTEGER	11	✓
	sched_name	VARCHAR	32	
	sched_abbr	VARCHAR	8	
	sched_start	TIME	0	
	sched_end	TIME	0	
	sched_day	SMALLINT	6	

Tabla student (registro)

Descripción Tabla en la que se almacena la informacion relacionada con los estudiantes

Campos

Llave primaria	Nombre	Tipo de dato	Tamaño	No nulos
✓	stud_id	VARCHAR	16	✓
	stud_firstname	VARCHAR	64	
	stud_middlename	VARCHAR	64	
	stud_lastname1	VARCHAR	64	
	stud_lastname2	VARCHAR	64	
	stud_dob	DATE	0	
	stud_addrlocal	VARCHAR	128	
	stud_citylocal	VARCHAR	64	
	stud_statelocal	VARCHAR	64	
	stud_phonelocal	VARCHAR	16	
	stud_addrperm	VARCHAR	128	
	stud_cityperm	VARCHAR	64	
	stud_stateperm	VARCHAR	64	
	stud_countryperm	VARCHAR	64	

	repeatx	INT	7	
	title	TEXT	0	
	desc	TEXT	0	
	days	INT	7	
	stop	INT	7	
	month	TINYINT	2	
	weekDay	TINYINT	1	
	weekNumber	TINYINT	1	
	category	TEXT	0	
	eventKey	TEXT	0	

Capítulo 4: Desarrollo

Desarrollo del sistema

Se puede definir como, el proceso de construir una solución funcional al problema; y consiste prácticamente en codificar la solución mediante un particular lenguaje de programación. Para llevar a cabo esta etapa, se requiere el conocimiento de un lenguaje de programación en particular en lo referente a su gramática, sintaxis y semántica, con el fin de plasmar las ideas en código y hacer funcional la WebApp.

En el trabajo de ingeniería de software la codificación puede ser:


- La creación directa del código fuente en un lenguaje de programación.
- La generación automática de código fuente al utilizar una representación intermedia del diseño del componente que será construido.

- La generación automática de código mediante un lenguaje de programación de cuarta generación.

Para el desarrollo de este proyecto se utilizó la creación directa del código fuente, y el lenguaje de programación elegido para la realización de dicho proceso fue PHP.

A través de las siguientes imágenes se muestra algunas de las secciones de la WebApp que ya han sido desarrolladas.

Registros marcados con un * son requeridos

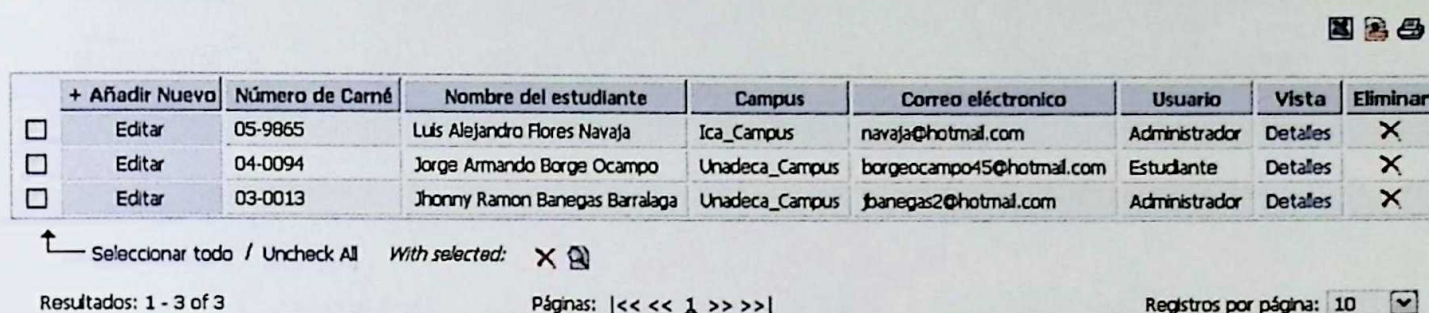
Campo	Valor del campo
* Número de Carné	03-0013
* Primer Nombre	Jhonny
* Segundo Nombre	Ramón
* Primer Apellido	Banegas
* Segundo Apellido	Barralaga
* Fecha de nacimiento	1984-12-19  [2007-11-29]
* Dirección local	100 Sur de la UNADECA
* Ciudad local	Alajuela
* Estado local	Alajuela
* Número de teléfono	899-1603
* Dirección permanente	Bo los laureles fte. auto partes Xavier
* Ciudad permanente	Tocoa
* Estado permanente	Colón
* País de origen	Honduras
* Número de teléfono permanente	444-3547
* Correo electrónico	jbaneas_07@hotmail.com
* Usuario	Estudiante <input type="checkbox"/>
* Campus	Unadeca_Campus <input type="checkbox"/>
* Tipo de estudiante	1
* Estado de admisión	1
* Estado académico	1

Cancelar Crear

Figura # 21: Formulario para introducir la información de un nuevo estudiante

La siguiente figura, muestra el formulario que mostrará todos los estudiantes que han sido registrados, este formulario le brinda la posibilidad a los usuarios con los permisos adecuados de agregar los

datos de un nuevo estudiante, editar y actualizar dicha información, además ver y eliminar la información de los estudiantes.



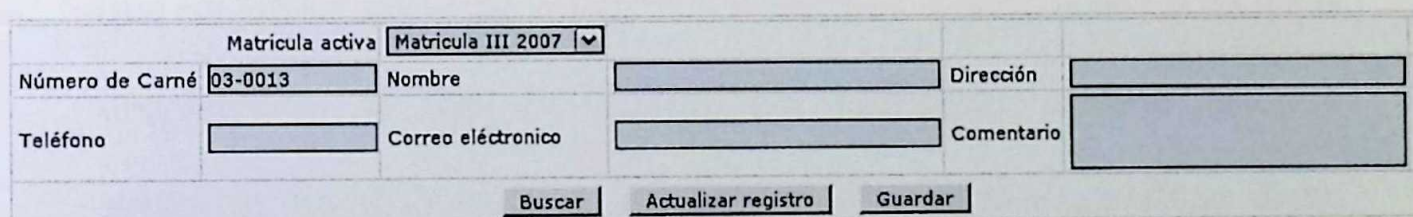
	+ Añadir Nuevo	Número de Carné	Nombre del estudiante	Campus	Correo electrónico	Usuario	Vista	Eliminar
<input type="checkbox"/>	Editar	05-9865	Luis Alejandro Flores Navaja	Ica_Campus	navaja@hotmail.com	Administrador	Detalles	X
<input type="checkbox"/>	Editar	04-0094	Jorge Armando Borge Ocampo	Unadeca_Campus	borgeocampo45@hotmail.com	Estudiante	Detalles	X
<input type="checkbox"/>	Editar	03-0013	Jhonny Ramon Banegas Barralaga	Unadeca_Campus	banegas2@hotmail.com	Administrador	Detalles	X

Seleccionar todo / Uncheck All With selected: X

Resultados: 1 - 3 of 3 Páginas: <<< << 1 >>> >>> Registros por página: 10

Figura # 22: formulario principal de estudiantes

A continuación se muestra una serie de imágenes que ejemplifica el proceso que se debe llevar a cabo para que un director de escuela apruebe una pre-matrícula.



Matricula activa		Matricula III 2007	
Número de Carné	<input type="text" value="03-0013"/>	Nombre	<input type="text"/>
Dirección	<input type="text"/>		
Teléfono	<input type="text"/>	Correo electrónico	<input type="text"/>
Comentario	<input type="text"/>		
<input type="button" value="Buscar"/>		<input type="button" value="Actualizar registro"/>	
<input type="button" value="Guardar"/>			

Figura # 23: página en la que el director de escuela busca la información del estudiante mediante el número de carné.

Cuando el director de escuela presiona el botón de buscar, si el estudiante ya pasó por el proceso de pre-matrícula, se mostraran los datos de pre-matrícula de lo contrario no mostrará ninguna información.

Matrícula activa		Matrícula III 2007			
Número de Carné	03-0013	Nombre	Jhonny Banegas Barralaga	Dirección	100 sur UNADECA
Teléfono	445-9868	Correo electrónico	jbaneas2@hotmail.com	Comentario	Aprobado
Clases a matricular					
Filtrar clases por escuela	...				
Código	Asignatura	Horario	Salón	Profesor	
IS-140	Estructura de Datos	HIT	C2	DL	Delete
	
Código	Asignatura	Horario	Salón	Profesor	
IS3-62	Administracion de Centros de Com	HML	LAB Comp 1	DC	Delete
	
<input type="button" value="Buscar"/> <input type="button" value="Actualizar registro"/> <input type="button" value="Guardar"/>					

Figura # 24: pantalla que muestra los datos de pre-matricula

A través de este formulario, los directores de escuela podrán buscar la información de pre-matricula de un estudiante, además de actualizar un registro, esto ocurre cuando el estudiante ha pre-matriculado una clase que no puede llevar porque aun no ha cumplido su pre-requisito o simplemente se equivocó al seleccionar al profesor que impartirá una clase, entonces el director de escuela a través de listas desplegables de bajo de cada sección (asignatura, horario, salón y profesor), podrá seleccionar la información correcta y actualizar dicho registro. En este mismo formulario el director tiene la opción de eliminar una clase pre-matriculada.

Otro de los formularios desarrollados es el que muestra las clases que un profesor esta impartiendo en el periodo en curso, de cada una de estas clases se muestran los estudiantes que se han matriculado en dichas clases. A continuación se muestra dicho formulario.

Código	Asignatura	Salón	Horario	Detalles
ENF-267	Nutrición y Dietoterapia	A3	18:30:00 - 21:30:00	Detalles
IS-140	Estructura de Datos	C2	14:30:00 - 17:30:00	Detalles
IS-777	Estadística	A2	14:30:00 - 17:30:00	Detalles
LM-987	Algebra lineal	A1	14:30:00 - 17:30:00	Detalles

Figura # 25: formulario de clases impartidas por un profesor

Cuando el profesor de clic sobre el vinculo "Detalles", se muestra la lista de estudiantes que han matriculado dicha clase.

Carné	Alumno	Créditos	Nota	Tipo	Observaciones
04-0094	Jorge Armando Borge Ocampo	3	65	AUS ▼	Fracasada por ausencias!!
05-9865	Luis Alejandro Flores Navaja	3	78	... ▼	

Figura # 26: Alumnos matriculados en una clase

Como se logra apreciar, se muestra la información básica del estudiante más la cantidad de créditos que tiene la clase, es importante notar que la WebApp permite introducir la nota que el estudiante obtuvo y un comentario, dicho comentario se usa cuando en el campo tipo se ha especificado una de las siguientes opciones: RET significa que la clase fue retirada por es estudiante, AUS fue fracasada por ausencias y INC dice que la clase esta incompleta.

Capítulo 5: Pruebas

5.1 Pruebas del sistema

En esta etapa el sistema es utilizado en forma experimental, con el objetivo principal de asegurar que la WebApp no falle, es decir, que trabaje de acuerdo a las especificaciones y de la manera en la que los usuarios esperan que lo haga.

Para el desarrollo de las pruebas se realizaron pruebas en relación al flujo de datos, estas pruebas consisten en verificar que los datos sean introducidos en el orden adecuado y que los datos que se introducen sean consistentes con el tipo de datos que almacenara la base de datos.

5.2 Desarrollo de pruebas

5.2.1 Pruebas de consistencia de datos

Con esta prueba se pretende verificar que la aplicación solo permite introducir los datos que vayan de acuerdo con la definición del tipo de datos que se haya especificado cuando se diseño la tabla almacenará dicha información.

Para realizar esta prueba se usa como página experimental, la página que permite introducir las fechas de inicio y fin de: clases, matrícula, matrícula extraordinaria e introducción de notas.

Registros marcados con un * son requeridos



Campo	Valor del campo
* Comienzo Período Matricula	[2007-11-27]
* Fin Período Matricula	[2007-11-27]
* Comienzo Período Matricula Extraordinaria	[2007-11-27]
* Fin Período Matricula Extraordinaria	[2007-11-27]
* Comienzo de Clases	[2007-11-27]
* Fin de Clases	[2007-11-27]
* Comienzo Ingresar notas	[2007-11-27]
* Fin Ingresar notas	[2007-11-27]

Cancelar Crear

Figura # 27: sección para introducir fechas.

Como se muestra en la figura # todos los campos que están marcados con asterisco (*) son campos que no se pueden obviar. En la siguiente imagen se muestra que si se intenta realizar la creación de un registro que almacene estas fechas evitando alguna de ellas la WebApp no permitirá realizar dicho creación.

Registros marcados con un * son requeridos



Campo	Valor del campo
* Comienzo Período Matricula	2007-09-24 [2007-11-27]
* Fin Período Matricula	2007-09-04 [2007-11-27]
* Comienzo Período Matricula Extraordinaria	2007-09-05
* Fin Período Matricula Extraordinaria	2007-09-11
* Comienzo de Clases	
* Fin de Clases	
* Comienzo Ingresar notas	
* Fin Ingresar notas	

La página en <http://localhost:8080> dice:

The <Fecha Comienzo de Clases> is a required field!
The <Fecha Fin de Clases> is a required field!
The <Fecha inicial para poner notas> is a required field!
The <Fecha final para poner notas> is a required field!

Aceptar

Cancelar Crear

Figura # 28: fallo al intentar realizar la creación de fechas con algunos campos requerido omitidos.

La segunda prueba de consistencia de datos se desarrolla en la página para ingresar nuevas clases a la tabla de que almacena toda la información relacionada con las clases.

A continuación se muestra la pantalla en la que se introducen los datos necesarios para la creación de una clase nueva.

Registros marcados con un * son requeridos 🖨️ 🗑️ 📄

Campo	Valor del campo
* Facultad	Ingeniería <input type="button" value="v"/>
* Código	<input type="text"/>
* Nombre	Sistemas Operativos II <input type="text"/>
Créditos	3 <input type="text"/>
Electiva	<input type="checkbox"/>
* Descripción	Esta clase pretende profundizar sobre las técnicas para el desarrollo y mejora de rendimiento de sistemas operativos

Cancelar Crear

Figura # 29: pantalla para introducir clases.

Como se muestra en la figura anterior también hay campos que son requeridos, y si alguno o todos de estos campos se evade, la WebApp no permitirá realizar la creación.

Registros marcados con un * son requeridos



Campo	Valor del campo
* Facultad	Ingeniería
* Código	
* Nombre	Sistemas Operativos II
Créditos	3
Electiva	<input type="checkbox"/>
* Descripción	Esta clase pretende profundizar sobre las técnicas para el desarrollo y mejora de rendimiento de sistemas operativos

La página en http://localhost:8080 dice:

The <Código de Clase> is a required field!

Aceptar

Cancelar Crear

Figura # 30: falló la creación de una clase.

5.2.2 Pruebas de flujo de datos

Estas pruebas permiten comprobar que la WebApp tiene la capacidad de manejar datos que dependen de la existencia de otra información.

Para llevar a cabo estas pruebas se utiliza la página que permite ingresar la información de las facultades de una institución.

Registros marcados con un * son requeridos



Campo	Valor del campo
* Institución	-- seleccionar --
* Nombre de la facultad	-- seleccionar -- Ingeniería en Sistemas
* Director	Dodarin Castillo
Descripción de la facultad	Facultad de Ingeniería en sistemas computacionales.
* Abreviatura	ING

Cancelar Crear

Figura # 31: sección de la WebApp para agregar facultades.


En la figura anterior se puede apreciar que hay ciertos campos requeridos para poder agregar una facultad con éxito, entre estos campos esta el campo de institución, que especifica a que institución pertenece una facultad, si aun no se ha ingresado una o mas instituciones al sistema entonces no se podrá mostrar las instituciones disponibles por lo tanto esto generaría un error de flujo de datos ya que se esta intentando ingresar una facultad cuando ni siquiera se ha ingresado una institución.

La siguiente figura se muestra que la WebApp no permite agregar una facultad si que exista una institución a la cual relacionar dicha facultad.

Registros marcados con un * son requeridos

Campo	Valor del campo
* Institución	-- seleccionar --
* Nombre de la facultad	Ingeniería en sistemas
* Director	Dodanin Castillo
Descripción de la facultad	Facultad de Ingeniería en sistemas computacionales.
* Abreviatura	ING

La página en http://localhost:8080 dice:

 The <Institution Name> is a required field!

Cancelar Crear

Figura # 32: falló la creación de una facultad.

Otra prueba es la perteneciente a los programas de estudio, cuando se pretende crear un nuevo programa de estudio se tiene que especificar, entre otros campos, a que facultad o escuela pertenece el programa de estudio.

En la siguiente figura se muestra la página y lo campos requeridos para ingresar un nuevo programa de estudio.

Registros marcados con un * son requeridos

Campo	Valor del campo
* Nombre del programa	Programa de estudio de Ingeniería
* Abreviatura	PDE-ING
* Facultad	-- seleccionar --
Descripción	<div style="border: 1px solid black; padding: 5px;"> -- seleccionar -- estudio de la facultad de Ingeniería de sistemas computacionales. </div>

Cancelar Crear

Figura # 33: página para agregar programas de estudio.

Si cuando se esta agregando un programa de estudio, previamente no se han definido una o varias facultades, ningún programa de estudio podrá ser ingresado al sistema.

Registros marcados con un * son requeridos

Campo	Valor del campo
* Nombre del programa	Programa de estudio de Ingeniería
* Abreviatura	PDE-ING
* Facultad	- seleccionar -
Descripción	Programa de estudio de la facultad de Ingeniería de sistemas computacionales.

La página en http://localhost:8080 dice:

The <Faculty Name> is a required field!

Aceptar

Cancelar Crear

Figura # 34: error al agregar programa de estudio, no se especificó la escuela a la que pertenece el plan a crear.

La WebApp además de comprobar que el usuario ha brindado la información para los campos requeridos, permite usar información que debería existir para evitar que el usuario este escribiendo información existente.

5.2.3 Pruebas de usabilidad

La pruebas usabilidad (dentro del campo del desarrollo web) es la disciplina que estudia la forma de diseñar WebApps para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible.

Para el desarrollo de esta prueba, se usa la página en la que se introducen los pre-requisitos de clases.

Campo	Valor del campo
* Clases	Contabilidad II <input type="button" value="v"/>
* Pre-requisito	Contabilidad I <input type="button" value="v"/>

Figura # 35: Página que permite definir pre-requisitos de clases.

Como se detalla en la figura anterior, esta sección de la WebApp permite al usuario definir pre-requisitos a través de listas desplegables que muestran las clases que se están almacenadas en la base de datos; esto facilita al usuario ya que él no escribe los datos de la clase y su pre-requisito, solo selecciona de los disponibles.


5.2.4 Pruebas de las bases de datos

La interfaz de las WebApps con conexiones a sistemas de bases de datos sofisticados gestionan sistemas y construyen objetos de contenido dinámicos que se crean en tiempo real aprovechando datos adquiridos de una base de datos.

El objetivo de estas pruebas es validar que los datos solicitados por el usuario son exactamente los que la WebApp despliega para uso de dicho usuario.

Para llevar a cabo estas pruebas se usa la unidad de reportes de matriculas, en donde el usuario introduce el número de carné y despliega las clases matriculadas en el cuatrimestre en curso.

Generar reporte para alumno:



Universidad Adventista de Centro América

Matricula activa: Matricula III 2007

Número de Carné	05-9865	Nombre	Luis Flores Navaja	Facultad	Administracion
Teléfono	456-9898	Dirección	Alajuela	Correo electrónico	navaja@hotmail.com

CLASES MATRICULADAS

CÓDIGO	ASIGNATURA	HORARIO	SALÓN	PROFESOR
IS-777	Estadística	HVL	A2	OL
IS-987	Calculo II	HML	C1	DC

Firma del Alumno	Firma director de registro
------------------	----------------------------

Figura # 36: Reporte de matrícula.

5.2.5 Pruebas de seguridad

Las pruebas de seguridad a WebApp permiten detectar las vulnerabilidades del sistema a ataques de personas no autorizadas para ingresar a ella; para una WebApp de registro es de vital importancia tener bases solidas de seguridad ya que manejan información que solo las personas autorizadas deben manipula, por lo tanto dichas pruebas son imprescindibles para desarrollo de este proyecto.

Seguidamente se muestra una imagen con la pantalla principal del sistema en donde cada uno de sus usuarios deben de proveerle sus credenciales de autenticación necesarias para poder ingresar a la WebApp.

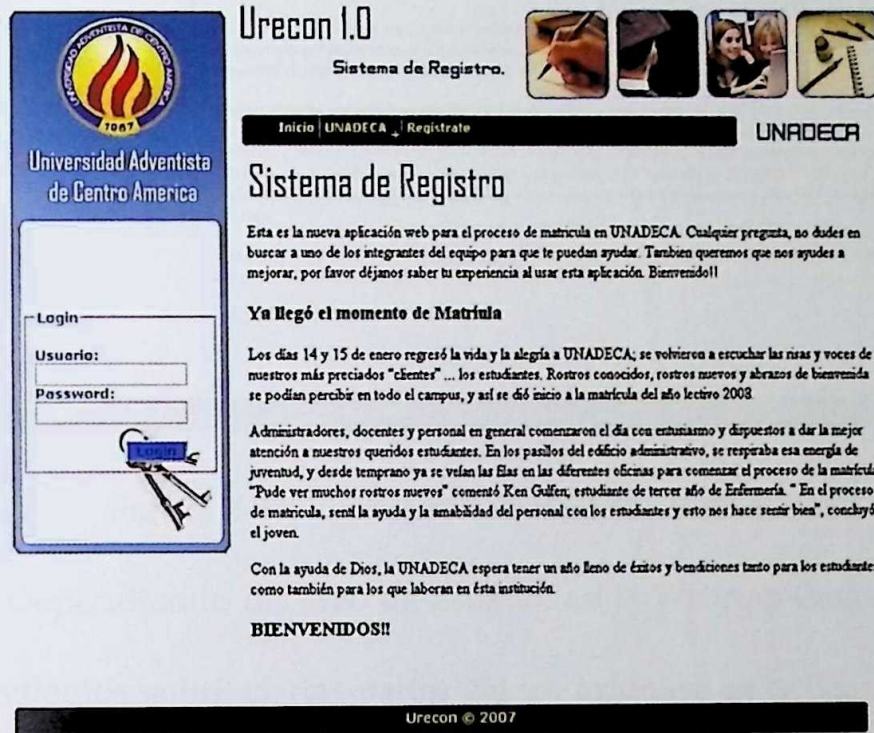
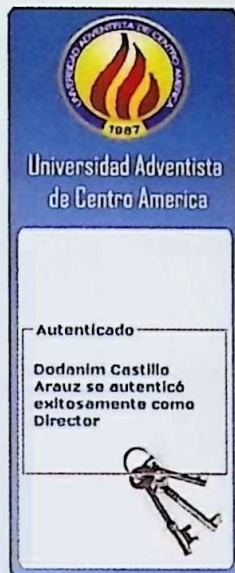


Figura # 37: Página principal de la WebApp de registro.

Si el usuario no introduce su nombre de usuario y contraseña correctos no podrá acceder a la WebApp.

En la siguiente figura se muestra el resultado de que un usuario se haya autenticado correctamente.



Urecon 1.0

Sistema de Registro.



Inicio UNADECA Académicos Registro Salir

UNADECA

Sistema de Registro

Esta es la nueva aplicación web para el proceso de matrícula en UNADECA. Cualquier pregunta, no dudes en buscar a uno de los integrantes del equipo para que te puedan ayudar. También queremos que nos ayudes a mejorar, por favor déjanos saber tu experiencia al usar esta aplicación. Bienvenido!!

Ya llegó el momento de Matrícula

Los días 14 y 15 de enero regresó la vida y la alegría a UNADECA, se volvieron a escuchar las risas y voces de nuestros más preciados "clientes" ... los estudiantes. Rostros conocidos, rostros nuevos y abrazos de bienvenida se podían percibir en todo el campus, y así se dio inicio a la matrícula del año lectivo 2008.

Administradores, docentes y personal en general comenzaron el día con entusiasmo y dispuestos a dar la mejor atención a nuestros queridos estudiantes. En los pasillos del edificio administrativo, se respiraba esa energía de juventud, y desde temprano ya se veían las filas en las diferentes oficinas para comenzar el proceso de la matrícula. "Pude ver muchos rostros nuevos" comentó Ken Guillen, estudiante de tercer año de Enfermería. "En el proceso de matrícula, sentí la ayuda y la amabilidad del personal con los estudiantes y esto nos hace sentir bien", concluyó el joven.

Con la ayuda de Dios, la UNADECA espera tener un año lleno de éxitos y bendiciones tanto para los estudiantes como también para los que laboran en esta institución.

BIENVENIDOS!!

Urecon © 2007

Figura # 38: El usuario se autenticó satisfactoriamente.

Dependiendo del tipo de usuario, así la WebApp dará o quitará privilegios sobre ciertas partes de los módulos, en la figura # 38 se muestra que un usuario con privilegios de director de escuela se ha autenticado.

En la siguiente imagen se muestra que el administrador de la WebApp se ha autenticado con éxito.



Universidad Adventista
de Centro America

Autenticado

Felicidades usted se
autenticó
exitosamente como
Administrador



Urecon 1.0

Sistema de Registro.



Inicio UNADECA Académicos Registro Reportes CLEB Salir UNADECA

CLEB CAMPUS



+ Add New		Institution	Campus	Abreviation	Description	Phone No.	Fax No.	View	Delete
<input type="checkbox"/>	Edt	Iter-American Division Theological Seminar	Ica_Campus	IcaCamp	unadeca campus	4452365	4452158	Details	X
<input type="checkbox"/>	Edt	Universidad Adventista De Centro America	Unadeca_Campus	Un3Camp	ica campus	4404560	4450698	Details	X

↑ Check All / Uncheck All With selected: X

Results: 1 - 2 of 2

Pages: |<< < 1 > >>|

Page size: 10

Figura # 39: El administrador de la WebApp en la sección de campus

Capítulo 6: conclusiones y recomendaciones

Conclusiones

- Una WebApp que englobe el proceso de registro y matrícula de estudiantes, es indispensable para una institución de nivel universitario, ya que esta facilita los dichos procesos y da una mejor imagen de la institución.
- El uso de una WebApp para llevar el control de los alumnos y las clases que estos han matriculado permite un ahorro significativo de tiempo y dinero tanto para el departamento de registro como para todas las personas involucradas en este proceso.
- UML es una valiosa herramienta para el análisis y diseño de sistemas orientados a objetos.
- Apache, PHP y MySQL en conjunto forman un poderoso grupo de aplicaciones para el desarrollo de aplicaciones Web.

Recomendaciones

- Ya que solo se desarrollaron dos módulos de la aplicación Web para el registro y matrícula de los alumnos de la Universidad Adventista de Centro América, se recomienda que se dedique el tiempo y dinero adecuado para el desarrollo de los módulos restantes que conllevará a tener una WebApp de registro completa.

Capítulo 7: Glosario

Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

API

La interfaz de Programación de Aplicaciones es un Conjunto estándar de funciones software y formatos de datos utilizados como interfaz para construir programas.

ASP

Active Server Pages es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

Base de datos

Conjunto de información almacenada y accesible de forma sistemática a varios usuarios. Suele admitir la selección de acceso aleatorio y múltiples vistas o niveles de abstracción de los datos. Se divide normalmente en registros, y estos, a su vez, en campos.

Base de datos relacional

Una base de datos relacional es un conjunto de dos o más tablas estructuradas en registros (filas) y campos (columnas), que se vinculan

Capítulo 6: conclusiones y recomendaciones

Conclusiones

- Una WebApp que englobe el proceso de registro y matrícula de estudiantes, es indispensable para una institución de nivel universitario, ya que esta facilita los dichos procesos y da una mejor imagen de la institución.
- El uso de una WebApp para llevar el control de los alumnos y las clases que estos han matriculado permite un ahorro significativo de tiempo y dinero tanto para el departamento de registro como para todas las personas involucradas en este proceso.
- UML es una valiosa herramienta para el análisis y diseño de sistemas orientados a objetos.
- Apache, PHP y MySQL en conjunto forman un poderoso grupo de aplicaciones para el desarrollo de aplicaciones Web.

Recomendaciones

- Ya que solo se desarrollaron dos módulos de la aplicación Web para el registro y matrícula de los alumnos de la Universidad Adventista de Centro América, se recomienda que se dedique el tiempo y dinero adecuado para el desarrollo de los módulos restantes que conllevará a tener una WebApp de registro completa.

Capítulo 7: Glosario

Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

API

La interfaz de Programación de Aplicaciones es un Conjunto estándar de funciones software y formatos de datos utilizados como interfaz para construir programas.

ASP

Active Server Pages es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

Base de datos

Conjunto de información almacenada y accesible de forma sistemática a varios usuarios. Suele admitir la selección de acceso aleatorio y múltiples vistas o niveles de abstracción de los datos. Se divide normalmente en registros, y estos, a su vez, en campos.

Base de datos relacional

Una base de datos relacional es un conjunto de dos o más tablas estructuradas en registros (filas) y campos (columnas), que se vinculan

entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave.

Consulta SQL

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

DHTML

Dynamic HTML (HTML dinámico), combinación de HTML, hojas de estilo y Java script que permiten modificaciones automáticas en los elementos de las páginas.

Diagrama de actividades

Estos diagramas son útiles para representar los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema.

Diagrama de casos de usos

Es una especie de diagrama de comportamiento en el que se esbozan las actividades realizadas por los actores de un sistema.

Diagrama de clases

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones.

Diagrama de comunicación

Un diagrama de comunicación muestra la colaboración dinámica entre los elementos. Es similar al diagrama de secuencia y la intención es para enfocar cómo los objetos colaboran con cada otro.

Diagrama de despliegue

Diagrama que describe cómo una aplicación se despliega a través de una infraestructura. La intención del modelo de despliegue no es para describir la infraestructura, pero mejor dicho el camino en cual los componentes específicos deben corresponder a una aplicación que despliega a través de él.

Diagrama de estado

Los diagramas de estado muestran los diferentes cambios de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Diagrama de estructura compuesta

El diagrama de estructura compuesta toma el modelo para describir las relaciones entre los elementos para trabajar junto a una clasificación.

Diagrama de objetos

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto.

Diagrama de paquetes

Un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

Diagrama de secuencia

Diagrama que muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

HTML

Hyper Text Mark-up Language (lenguaje de marcas hipertextuales).
Lenguaje de programación para el desarrollo de páginas web.

HTTP

Hipertext Transfer Protocol (Protocolo de transferencia de hipertexto) es un conjunto de estándares que permite a los usuarios de la Web intercambiar información. Método que se utiliza para transferir documentos desde el sistema donde se almacenan las páginas hasta los usuarios individuales.

Internet

Es una red de computadoras alrededor de todo el mundo, que comparten información unas con otras por medio de páginas o sitios Web.

Intranet

Una intranet es una red de ordenadores de una red de área local (LAN) privada empresarial o educativa que proporciona herramientas de Internet.

Java Script

Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Lenguaje de programación

Sistema de escritura para la descripción precisa de algoritmos o programas informáticos.

Linux

Sistema operativo de código fuente libre para computadoras personales derivado de Unix.

Mac OS

Abreviatura de Macintosh Operating System (Sistema Operativo de Macintosh), es el nombre del primer sistema operativo de Apple para los

ordenadores Macintosh. El Mac OS original fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito.

Multihilo

Programación que permite que múltiples hilos se ejecuten de manera simultánea en el mismo programa.

MySQL

Servidor de bases de datos multiplataforma recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultáneas estadísticas.

ODBC

Open Database Connectivity (Conectividad de base de datos abierta) es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos.

Oracle

Sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Se considera como uno de los sistemas de bases de datos más completos, destacando su: soporte de transacciones, estabilidad, escalabilidad, es multiplataforma.

ordenadores Macintosh. El Mac OS original fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito.

Multihilo

Programación que permite que múltiples hilos se ejecuten de manera simultánea en el mismo programa.

MySQL

Servidor de bases de datos multiplataforma recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultáneas estadísticas.

ODBC

Open Database Connectivity (Conectividad de base de datos abierta) es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos.

Oracle

Sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Se considera como uno de los sistemas de bases de datos más completos, destacando su: soporte de transacciones, estabilidad, escalabilidad, es multiplataforma.

Perl

Lenguaje de programación multiplataforma para la manipulación de textos, ficheros y procesos. Se estructura en forma de script.

PHP

Es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "preprocesador de hipertexto".

POO

La Programación Orientada a Objetos (POO) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

Servidor Web

El servidor Web es un programa que corre sobre el servidor (máquina que almacena y maneja los sitios web) que escucha las peticiones HTTP que le llegan y las satisface.

SGBD

Software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Software

Programas, procedimientos y reglas para la ejecución de tareas específicas en un sistema de cómputo.

Tabla

Estructura que permite representar y almacenar información de forma más compacta y organizada.

Unix

Sistema operativo multitarea y multiusuario de gran importancia en el desarrollo y evolución de Internet.

UML

UML es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

WebApp

Es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet.

Windows

Es un sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft

Capítulo 8: Bibliografía

- Enciclopedia.us.es. (Página consultada el 22 de septiembre del 2007).
Lenguaje de programación, [Online]. Dirección URL:
http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n
- Joaquín García Murugarren. (Página consultada el 1 de octubre del 2007). *Manual de php*, [Online]. Dirección URL:
<http://www.webestilo.com/php/>
- Refsnes Data. (Página consultada el 22 de eptiembre del 2007). *Php tutorial*, [Online]. Dirección URL:
<http://www.w3schools.com/php/default.asp>
- programacionweb.net. (página consultada el 10 de octubre del 2007).
Checkbox y bases de datos, [Online]. Dirección URL:
<http://www.programacionweb.net/foros/mensaje/?num=314>
- mcF Preview. (página consultada el 10 de octubre del 2007). *Crear un checkbox con php y MySQL*, [Online]. Dirección URL:
http://www.mctekk.com/desarrollo/leer/php/crear_un_checkbox_con_php_y_mysql/40
- Daniel de la Cruz Heras, Carlos Rodriguez. (página consultada el 4 de octubre del 2007). *PHP y MySQL, Contenidos dinámicos*, [Online].
Dirección URL: <http://www.desarrolloweb.com/manuales/12/>

- Refsnes Data. (Página consultada el 6 de noviembre del 2007). *PHP Array functions*, [Online]. Dirección URL:
http://www.w3schools.com/php/php_ref_array.asp
- Joaquin Garcia. (Página consultada el 23 de septiembre del 2007). *UML: diagramas UML. ¿Qué es UML?*, [Online]. Dirección URL:
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>
- Joaquin Garcia. (Página consultada el 23 de septiembre del 2007). *UML: Casos de uso*, [Online]. Dirección URL:
<http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>
- Paul Hensgen. (Página consultada el 24 de septiembre del 2007). *Elementos de UML*, [Online]. Dirección URL:
<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>
- James Rumbaugh, Ivar Jacobson y Grady Booch, (2005). *El lenguaje unificado de modelado. Manual de referencia*. Addison wesley.
- Grupo CIENTEC, (Página consultada el 23 de septiembre del 2007). *UML: un lenguaje modelado*, [Online]. Dirección URL:
<http://www.cientec.com/analisis/uml.asp>
- Wikimedia Foundation, Inc. (Página consultada el 24 de septiembre del 2007). *Sistemas de gestión de base de datos*, [Online]. Dirección URL:

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos

- ALEGSA, (Página consultada el 2 octubre del 2007). *Definición de SGDB*, [Online]. Dirección URL:

<http://www.alegsa.com.ar/Dic/sghd.php>

- Wikimedia Foundation, Inc. (Página consultada el 24 de septiembre del 2007). *MySQL*, [Online]. Dirección URL:

<http://es.wikipedia.org/wiki/MySQL>

- Joaquín, (Página consultada el 2 de Octubre del 2007). *Manual de MySQL*, [Online]. Dirección URL:

<http://www.webestilo.com/mysql/>

- Jesus vega, (Página consultada el 10 de Octubre el 2007). *El servidor Web*, [Online]. Dirección URL:

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/nod e20.html>

- Ciberneta, (Página consultada el 12 de octubre). *Conceptos básicos del servidor Web*, [Online]. Dirección URL:

http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php

- Larman, C. (2003). *UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid: Pearson educación.

- Whitten, J. L., Bentley, L. D., Barlow V. M. (2003). *Análisis y diseño de sistemas de información*. Madrid: McGraw-Hill.
- Bruegge, B. y Dutoit, A. H. (2002). *Ingeniería de software orientado a objetos*. México: Pearson educación.
- Freedman, A. (2000). *Diccionario de computación*. Santafé de Bogotá, Colombia: McGraw-Hill.
- Kroenke, D. M. (2000). *Procesamiento de bases de datos. Fundamentos, diseño e instrumentación*. México: Prentice-Hall.
- Dawson, C. W. y Martín, G. (2002). *El proyecto fin de carrera en ingeniería informática: una guía para el estudiante*. Madrid: Pearson Educación.
- Pressman, R. S. (2006). *Ingeniería del software, un enfoque práctico*. México: McGraw-Hill.

Capítulo 9

9.1 Anexos

9.1.1 ¿Cómo crear respaldo de la base de datos?

Los respaldos o backups en ingles se pueden definir como: Acción de copiar archivos o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales, Son usados como un plan de contingencia en caso de que una falla o error se presente.

Dado que la información que se maneja en el departamento de registro de la UNADECA es de vital importancia es necesario saber como realizar estos respaldos periódicamente.

En este anexo se expondrá el uso de dos herramientas para realizar respaldos, la primera es una herramienta de distribución libre y la segunda una herramienta de distribución comercial.

Dichas herramientas son:

1. MySQL Administrator

MySQL Administrator es una aplicación de código abierto para gestionar el servidor de base de datos MySQL de forma gráfica; es un software multiplataforma, que por el momento se encuentra disponible para Linux y Microsoft Windows.

Esta herramienta permite realizar tareas administrativas sobre servidores de MySQL incluyendo:

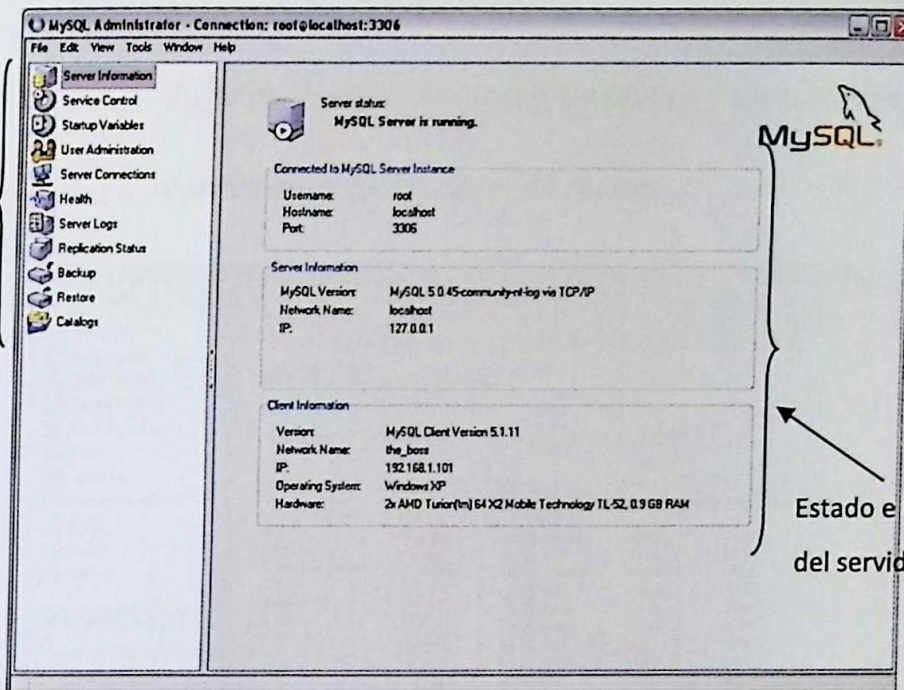
- Monitorización de conexiones al servidor.
- Administración de usuarios.
- Monitorización del estado del servidor, incluyendo estadísticas de uso.
- Gestión de copias de seguridad y recuperaciones.

1.1. Creación de respaldos con MySQL Administrator

A continuación se muestran los pasos a seguir para realizar un respaldo de base de datos como también como restaurar dicho respaldo.

La siguiente pantalla es la pantalla principal de MySQL Administrator en la que se muestra el estado e información básica del servidor y las diferentes operaciones que brinda a los usuarios.

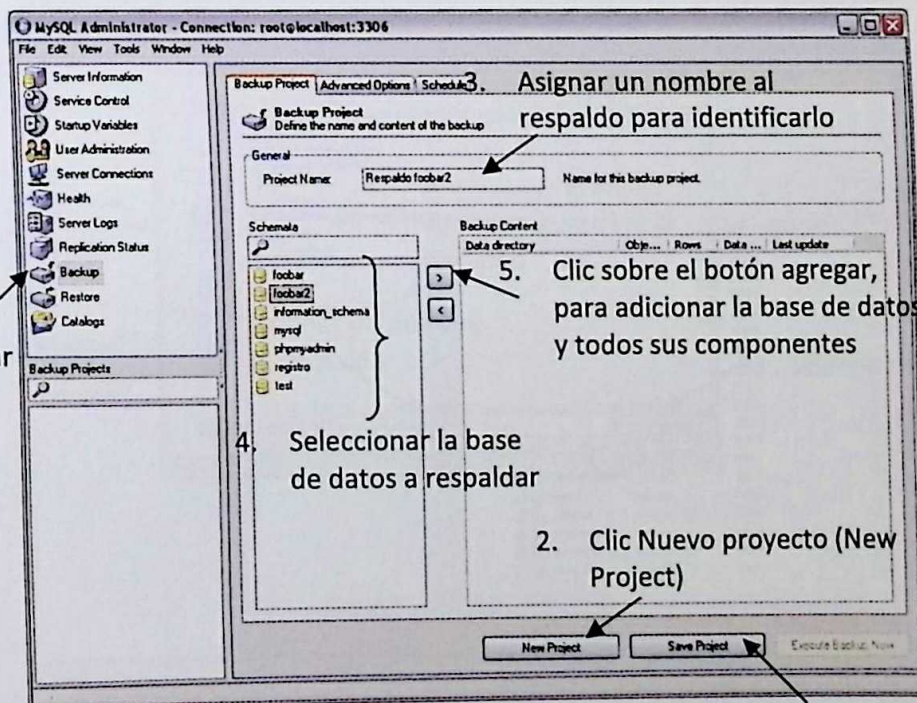
Diferentes operaciones



Estado e información del servidor

Se selecciona la operación respaldo (Backup) y dicha herramienta muestra la siguiente pantalla en la cual se deben seguir los siguientes pasos.

1. Seleccionar respaldos



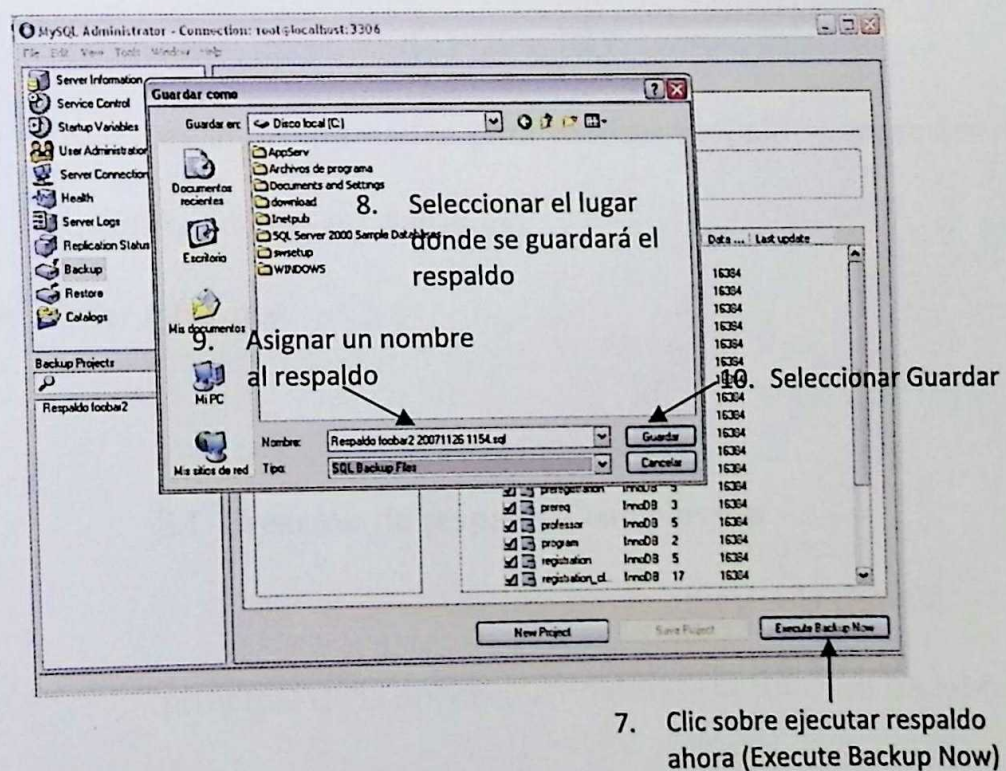
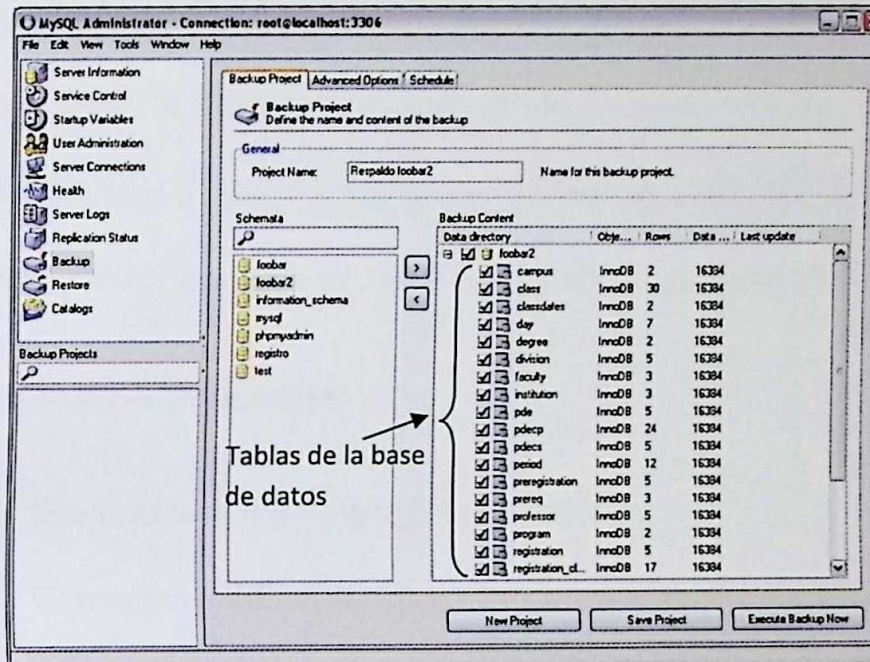
5. Clic sobre el botón agregar, para adicionar la base de datos y todos sus componentes

4. Seleccionar la base de datos a respaldar

2. Clic Nuevo proyecto (New Project)

6. Clic salvar proyecto (save project) para guardar el perfil respaldo de la base de datos.

Una vez seleccionada la base de datos, MySQL Administrator mostrará las tablas y demás objetos que conforman dicha base de datos.



2. Navicat 8 para MySQL

Navicat, una aplicación disponible para Windows, Mac y Linux que permite administrar servidores MySQL de una forma gráfica desde una aplicación de escritorio. Se puede conectar a servidores locales, de red o a través de Internet, para esto último se pueden utilizar conexiones seguras a través de un túnel SSH.

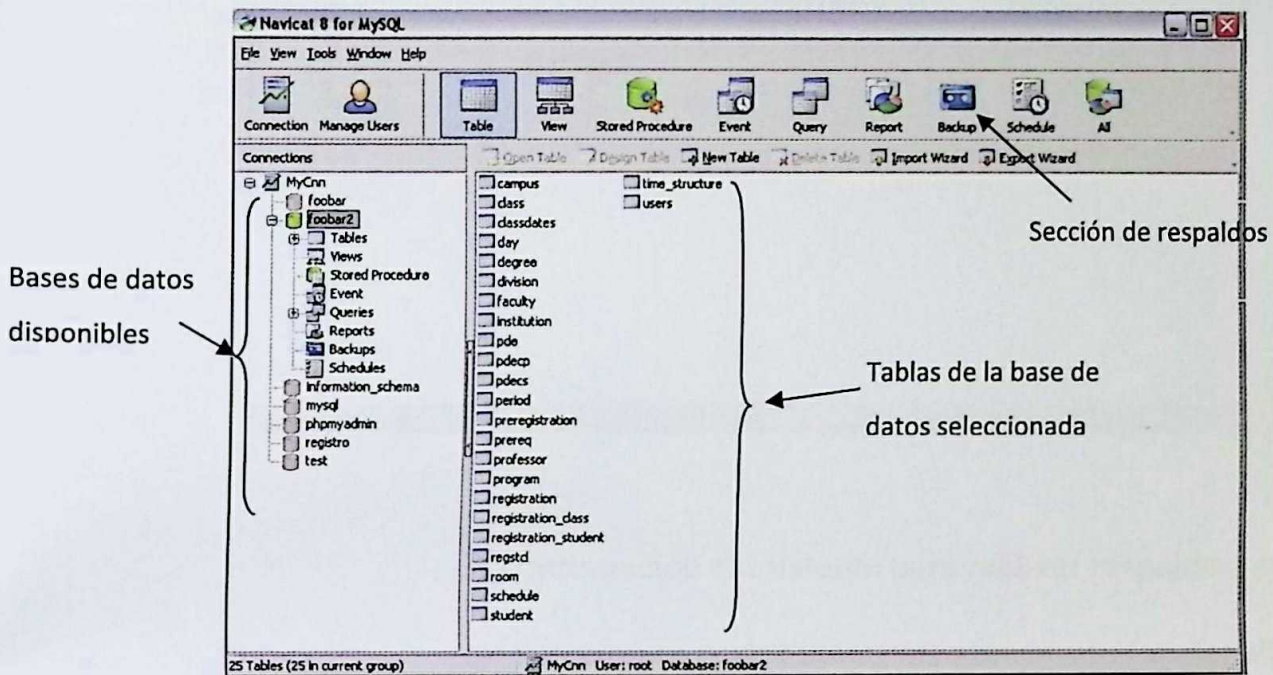
Características principales:

- Sincronización de estructuras y datos.
- Consultas a diferentes bases de datos de forma simultánea.
- Soporta todas las versiones de MySQL.
- Importar / Exportar datos de múltiples formatos conocidos, como Access, Excel, XML, PDF y TXT.
- Crear tareas programadas para realizar respaldos, importar o exportar, sincronizar datos, etc.
- Crear informes.

2.1 Creación de respaldos con Navicat

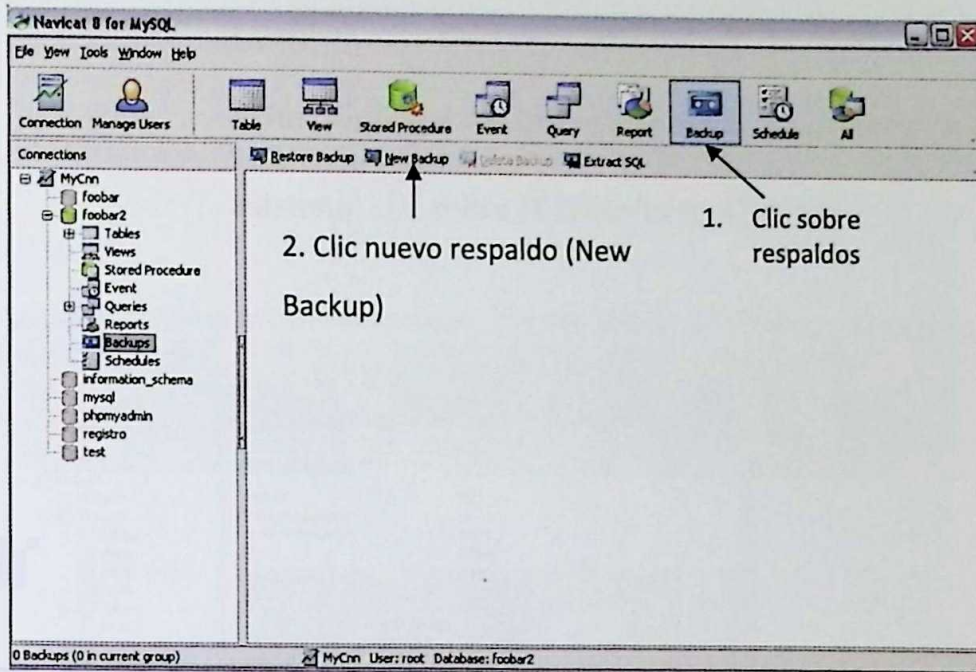
En la siguiente imagen se muestra la pantalla principal de la aplicación en donde se muestra las tablas

de la base de datos seleccionada, en este caso la base de datos seleccionada es foobar2.

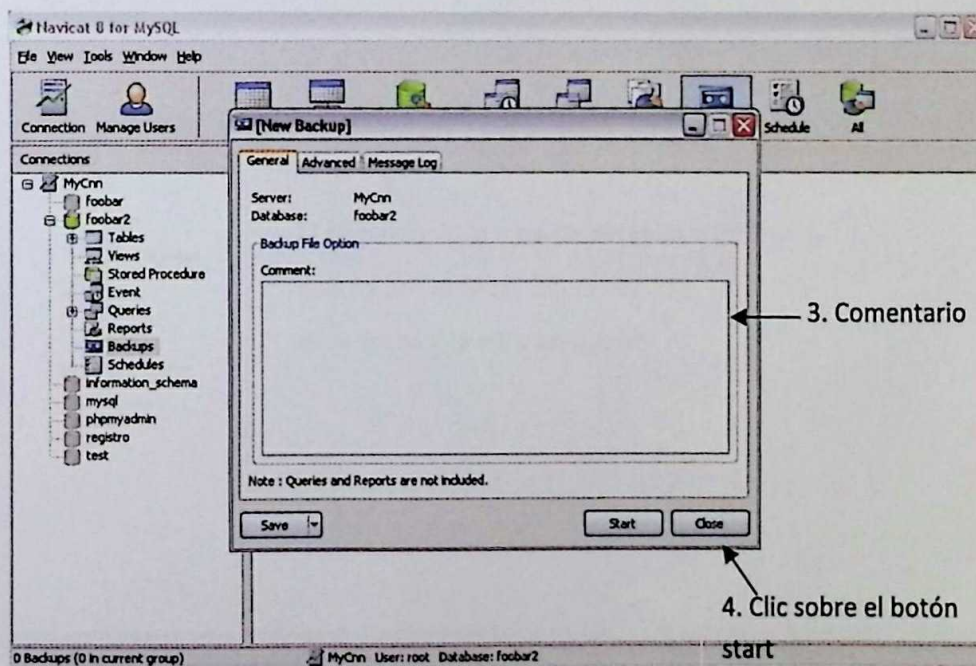


Después de haber dado una pequeña descripción de la interfaz de navicat, procederemos a mostrar los pasos que se deben llevar a cabo para realizar un respaldo de la base de datos.

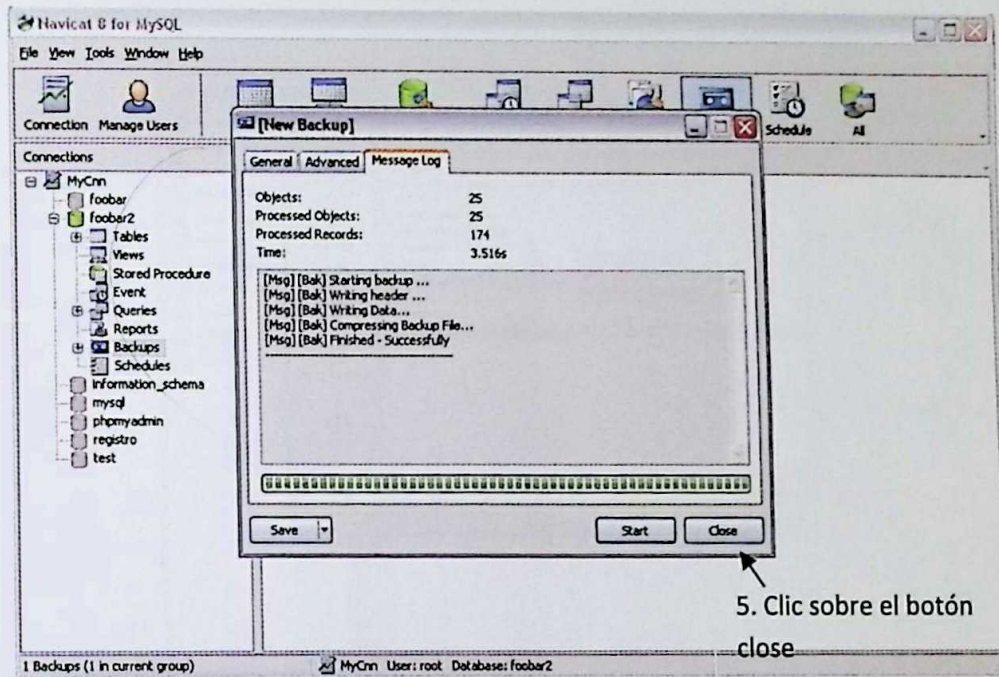
En primer lugar se selecciona la sección de respaldos, luego se procede a dar clic en nuevo respaldo.



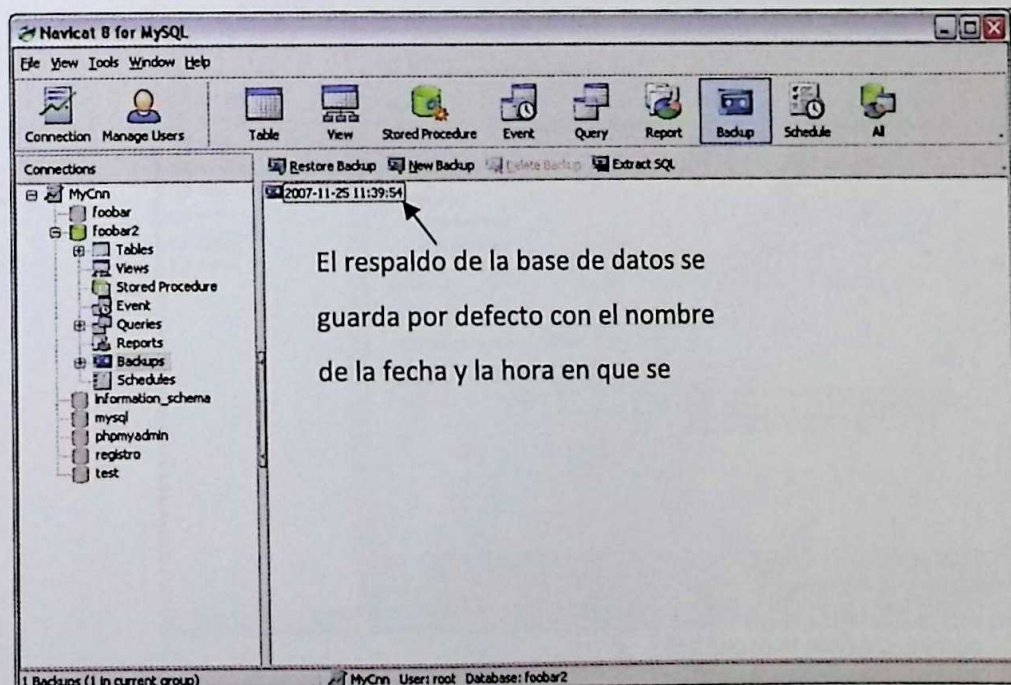
A continuación el asistente para realizar respaldos, en esta sección se podrá poner un comentario (opcional) con el propósito de diferenciar diferentes respaldos y para comenzar con la realización del respaldo clic sobre el botón start.



Luego el asistente dará un mensaje de finalizado con éxito (Finished - Successfully) o de error, para cerrar el asistente clic sobre el botón cerrar (close)



El respaldo se realizó con éxito

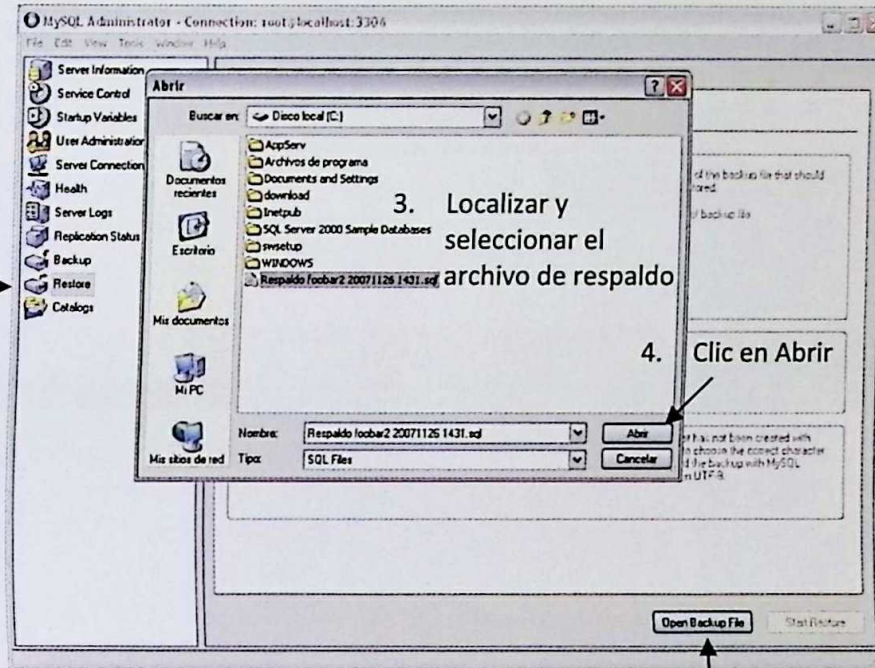


9.1.2 ¿Cómo restaurar respaldo de base de datos?

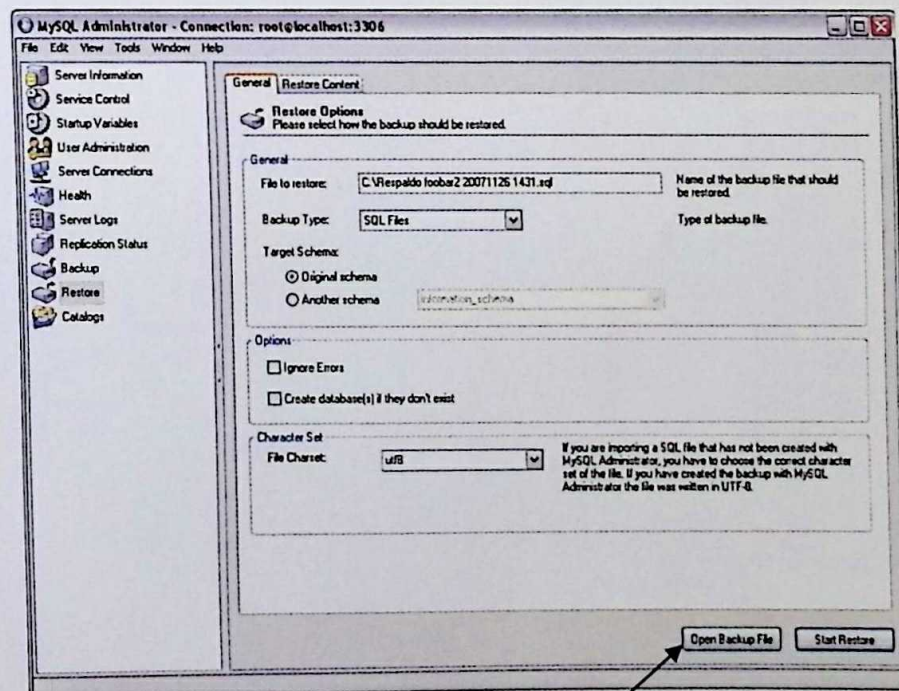
9.1.2.1 Restauración de un respaldo MySQL Administrator

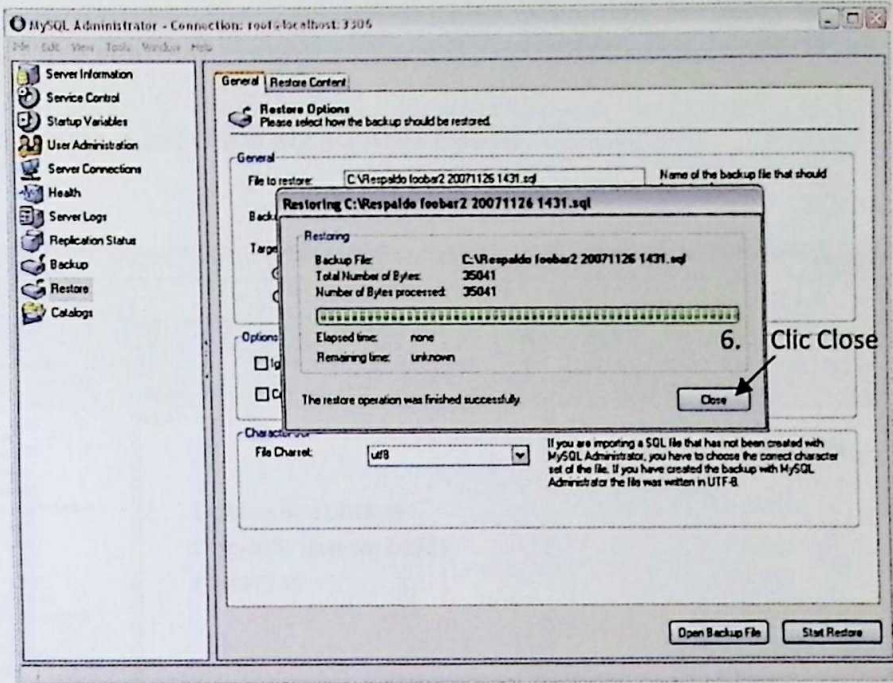
Seguidamente se muestran los pasos a seguir para restaurar un respaldo de base de datos.

1. Seleccionar restaurar (Restore)



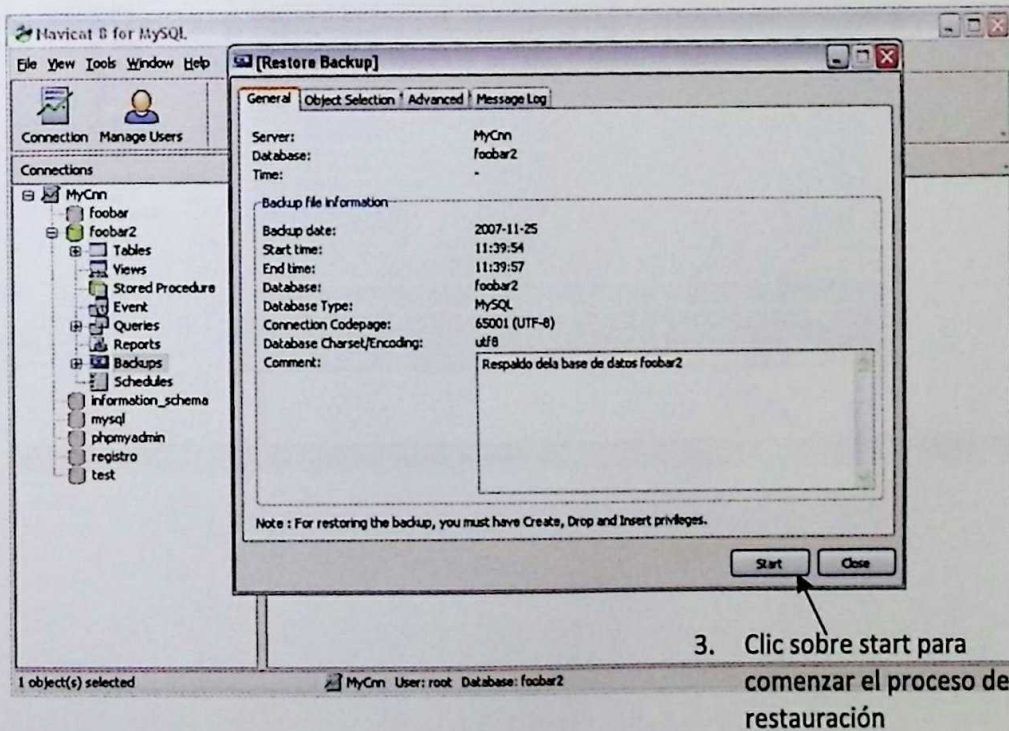
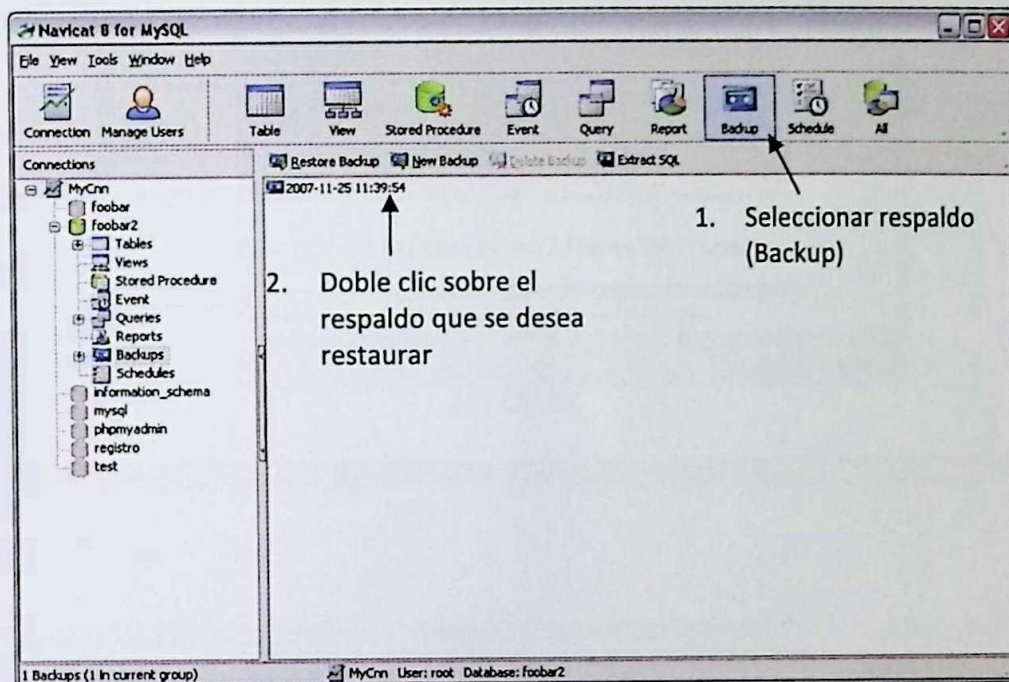
2. Seleccionar abrir archivo de respaldo (Open Backup File)

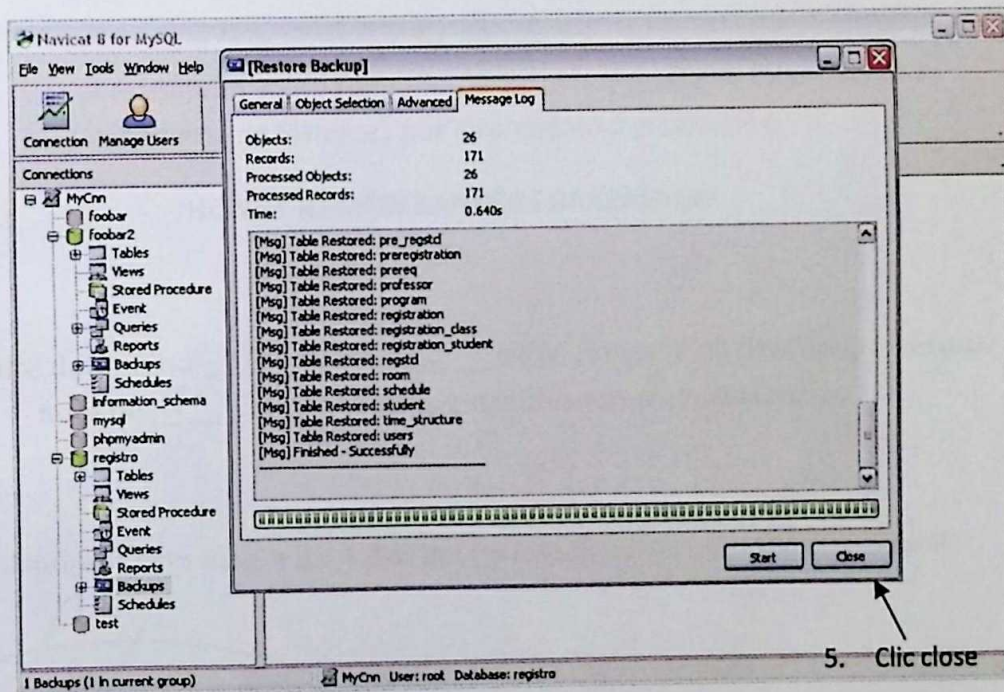
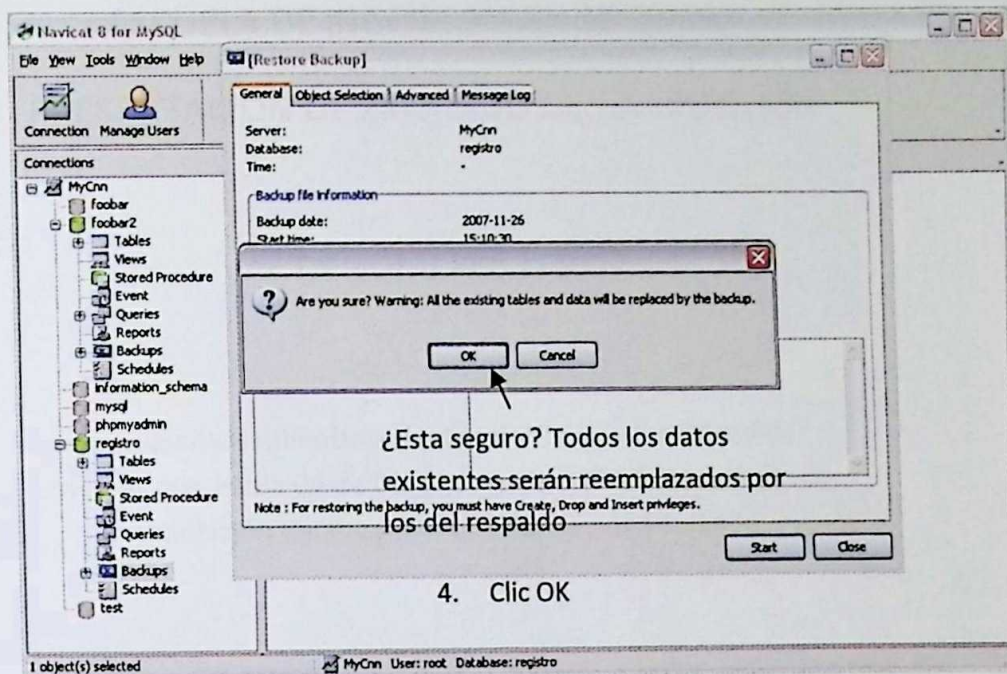




9.1.2.2 Restauración de respaldos con Navicat

Para realizar restauraciones de respaldos con Navicat, se deben llevar a cabo los siguientes pasos.





UNIVERSIDAD ADVENTISTA DE CENTRO AMÉRICA

ESCUELA DE INGENIERÍA EN SISTEMAS

PRESENTACIÓN DE PROYECTO DE GRADUACIÓN

ACTA NÚMERO: 1-2007

Los suscritos miembros del Comité Evaluador, hacemos constar que el día 04 de Diciembre de 2007, fue realizada la exposición del Proyecto de Graduación titulado:

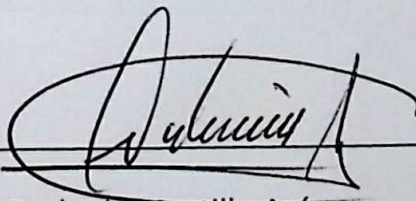
“Desarrollo de los módulos de matrícula y horario para la aplicación Web de Registro”

Presentado como requisito para optar al grado de Licenciatura en Ingeniería de Sistemas por el candidato a graduación:

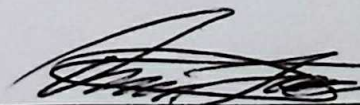
JHONNY RAMÓN BANEGAS BARRALAGA

Este comité determinó APROBADO dicho Proyecto de Graduación con una nota de 91, en virtud de lo cual firmamos la presente acta.

Dado en Alajuela, Costa Rica, a los 4 días del mes de Diciembre del año dos mil siete.


Ing. Dodanim Castillo Aráuz




Ingeniería en Sistemas
Computacionales
UNADECA - COSTA RICA

Máster Osman Longa