

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN-LEÓN
FACULTAD DE CIENCIAS
DEPARTAMENTO DE COMPUTACIÓN



**TESIS PARA OPTAR AL TÍTULO DE INGENIERO EN SISTEMAS
DE INFORMACIÓN**

TEMA:

**“AUTOMATIZACIÓN DEL MÓDULO PLANIFICACIÓN
ACADÉMICA PARA LA FACULTAD DE CIENCIAS UNAN-LEÓN
IMPLEMENTADO EN SISTEMAS DE CRÉDITOS ACADÉMICOS A
TRAVÉS DE SOFTWARE LIBRE”.**

Presentado por:

- Br. Francisco Javier Castro Olivas.
- Br. Karina Julieta Esquivel Alvarado.

TUTOR:

Msc. Álvaro Altamirano

Julio 2006

León, Nicaragua



CONTENIDO:

1. INTRODUCCIÓN.....	1
2. ANTECEDENTES.....	2
3. JUSTIFICACIÓN.....	4
4. OBJETIVOS.....	5
5. MARCO TEÓRICO.....	6
5.1. Sistema de Créditos.....	6
5.2. Sistema.....	11
5.3. El World Wide Web.....	11
5.4. Interfaces.....	11
5.5. Interfaz de usuario.....	11
5.6. Elementos de las Interfaces del sistema.....	12
5.7. Un sitio con páginas Web dinámicas.....	12
5.8. Clave foránea.....	12
5.9. Base de Datos.....	12
5.10. Software Libre	12
5.11. Estructura Cliente-Servidor.....	14
5.12. Software y herramientas para la implementación del sistema.....	15
5.12.1. PHP.....	15
5.12.1.1. Historia.....	15
5.12.1.2. Características.....	17
5.12.1.3. ¿Cómo funciona PHP?.....	18
5.12.1.4. Usos de PHP.....	19
5.12.1.5. Ventajas del uso de PHP.....	19
5.12.1.6. Desventajas del uso de PHP.....	20
5.12.2. Herramientas HTML.....	21
5.12.3. Servidor Apache.....	23
5.12.3.1. ¿Qué es el servidor Apache?.....	23
5.12.3.2. Características del Servidor Apache.....	25
5.12.3.3. Ventajas del uso del servidor Apache.....	26



5.12.4. Internet Information Server ISS.....	27
5.12.5. Apache vs. IIS: Dos servidores a elegir.....	28
5.12.6. Servidor Web.....	30
5.12.7. JAVASCRIPT.....	32
5.12.8. El Archipiélago Eclipse.....	33
5.12.8.1. ¿Qué es la plataforma Eclipse?.....	33
5.12.8.2. Eclipse y el software open source. Un matrimonio de conveniencia bien avenido.....	33
5.12.8.3. La arquitectura del SDK de Eclipse: una vista aérea.....	35
5.12.9. PostgreSQL.....	40
5.12.9.1. Inicio rápido con PostgreSQL.....	41
5.12.9.2. Integridad referencial.....	42
5.12.9.3. Configuración de PostgreSQL.....	42
6. DISEÑO METODOLÓGICO.....	44
6.1. MATERIALES.....	44
6.2. MÉTODOS.....	45
7. ANÁLISIS.....	48
7.1. DIAGRAMA DE FLUJOS DE DATOS.....	112
7.1.1. Diccionario de Datos.....	119
7.2. DIAGRAMA ENTIDAD RELACIÓN.....	121
7.3. DISEÑO.....	122
8. CONCLUSIONES.....	126
9. RECOMENDACIONES.....	127
10. BIBLIOGRAFÍA.....	128
ANEXOS	



AGRADECIMIENTO

En esta ocasión deseo expresar mi gratitud a todas las personas que me han brindado su amistad y cooperación justo en el momento en que las necesité.

GRACIAS:

Madre, porque has estado conmigo en las buenas y en las malas brindándome sabios consejos en el momento oportuno, para conducirme correctamente en el camino de la vida.

Roberto Castro, mi hermano, porque cuando te he necesitado siempre me has extendido la mano para brindarme tú ayuda.

Indiana por tu amor, apoyo incondicional y hacerme el hombre mas feliz al traer a este mundo a mi bella hija Alison Indiana Castro Espinoza.

Prof. José Padilla Palma, por su amistad y consejos.

Erick Quintana por el equipo, sin el mismo no hubiese sido posible la realización de este trabajo.

Y a todas aquellas otras personas que me han ofrecido su amistad en tan corto tiempo. A todos, **GRACIAS, Muchas gracias.**

Francisco Javier Castro Olivas



DEDICATORIA

A:

Alison Indiana Castro Espinoza
Mi Hija.

“Quien da alegrías y fuerzas a mi vida para seguir adelante”

Francisco Javier Castro Olivas



AGRADECIMIENTO

En este trabajo monográfico de alguna u otra manera me han ayudado muchas personas, pero en especial quiero agradecer a las que han estado siempre conmigo y que me han dado consejos, confianza y apoyo siempre que lo necesité, sin ellos no hubiese podido alcanzar una de mis metas que es culminar esta tesis.

Sin lugar a dudas mi primer agradecimiento, es para **Dios**, por darme la vida, nunca me ha dejado sola, siempre me ha iluminado y por haberme dado la oportunidad de estudiar una carrera, también puso en mi vida a personas que me han apoyado tanto.

A lo más hermoso que Dios nos dió “una Madre”, a tí que siempre me has apoyado en todas mis decisiones, tú que siempre me diste ánimo y fortaleza con tus ejemplos y palabras, dándome a entender que con esfuerzo, trabajo y fe nada es imposible, gracias **Maribel de Jesús Alvarado Pichardo** por darme lo mejor de tí y confiar en mí.

A mi abuela **Elba Luz Pichardo**, a quien considero mi segunda madre porque me ha apoyado tanto en cada momento de mi vida.

Al **Br. Erick Quintana Bárcenas**, Presidente 2005-2007 de la Asociación de Estudiantes de Ciencias por su incondicional apoyo en la realización de nuestro trabajo monográfico.

Para terminar, a una persona muy especial que ha hecho que mi vida sea un sueño, siempre apoyándome y entendiéndome aún en los momentos más difíciles, a ti mi más grande cariño.

Karina Julieta Esquivel Alvarado



DEDICATORIA

A:

MARIBEL DE JESÚS ALVARADO PICHARDO

Mi Madre.

“Por los años de esfuerzo y dedicación en mi educación”

Karina Julieta Esquivel Alvarado



1. INTRODUCCIÓN

No será extraño, dentro de poco tiempo, que los estudiantes universitarios del país se pregunten sobre el número de créditos que cursarán durante el semestre académico, ya que la UNAN-LEÓN pondrá en marcha el Sistema de Créditos Académicos en la Educación Superior que busca, en lo fundamental, dar mayor flexibilidad e interacción académica en y entre universidades, establecer criterios para la homologación de cursos y definir en todo el país la intensidad del trabajo académico de los estudiantes.

Con dicha implementación aparece la necesidad de desarrollar un software que permita la gestión y planificación de la información concerniente a la organización de horarios, aulas, profesores, etc. así como también de los pénsum académicos de cada una de las carreras ofertadas en la Universidad y en lo fundamental nuestra aplicación estará enfocada a la automatización del módulo de la Planificación Académica para la Facultad de Ciencias como un primer paso.

Una característica a destacar en nuestro proyecto es la utilización de Software Libre para el desarrollo del sistema ya que la necesidad de automatizar procesos, la existencia de una ley que protege los derechos de autor aunado con las condiciones económicas en la que nos encontramos actualmente, nos ha motivado de forma significativa a la utilización de Software libre como una alternativa eficiente que permitirá disminuir los costos de mantenimiento por el pago de licencia de software propietario (comerciales).



2. ANTECEDENTES

Antes de la década de los 80, las carreras universitarias en Nicaragua, estaban organizadas en su totalidad, por un sistema de créditos. Estos se contabilizaban de acuerdo a las horas teóricas y prácticas de cada asignatura, siendo el crédito equivalente a la suma de las horas teóricas y en horas prácticas significaba medio crédito por cada hora.

A pesar de reconocer que el sistema de créditos académicos, en óptimas condiciones de aplicación, ofrece una serie de ventajas ligadas a la flexibilidad, en la década de los 80 dada la escasez de recursos humanos, financieros e informáticos no existía un Software que mantuviese un control automatizado de la información generada con su aplicación en la enseñanza universitaria de la época, esto agravado por la decisión de masificar la Educación Superior, el Consejo Nacional de la Educación Superior (CNES) recomendó la introducción del sistema de bloques en sustitución del sistema de créditos en la Educación Superior Nicaragüense para lo cual se desarrolló e implementó en la UNAN-León a partir de 1983 una aplicación que dió respuesta a la introducción de dicho sistema.

El CNES justificó esta decisión señalando que el sistema de créditos no permitía planificar, a corto plazo, la cantidad de profesionales que serían formados, y que además dificultaba la planificación de las actividades académicas (horarios, aulas, profesores, etc.) resaltando las desventajas relacionadas con sus costos, recursos humanos y materiales, que no permitían la repetición de todas las asignaturas en cada semestre, lo cual elimina una de las mayores ventajas del sistema de crédito y sobre todo que la repetición de los alumnos en las asignaturas aplazadas, encarece el costo y disminuye la eficiencia académica. Así, para introducir, tal cambio, se apoyan: principalmente en el propósito de fortalecer la dirección centralizada del desarrollo de la educación superior y de esta suerte, planificar mejor la oferta y demanda del graduado.



A favor del sistema de bloques se argumentó que permitiría mayor orden en los registros académicos de los estudiantes y facilitaría la organización de horarios, aulas, profesores, etc. y el seguimiento del número esperado de graduados. Ambos sistemas poseen ventajas y desventajas, sin embargo, el contexto cambiante y las tendencias educativas, han llevado a las autoridades universitarias, a tomar la decisión de estudiar la posibilidad de transitar nuevamente hacia un sistema de créditos.

Nuestro esfuerzo se enfoca a desarrollar el sistema para el módulo de la Planificación Académica de la Facultad de Ciencias bajo el Sistema de Créditos ya que el Software existente está enfocado al sistema de evaluación académica por bloque, programado en software comercial como Visual FoxPro, Visual Basic, Microsoft Access.



3. JUSTIFICACIÓN

La Educación Superior se encuentra enfrentada a grandes retos, ya que debe responder a las necesidades de una realidad compleja y en constante cambio. Muchos son los argumentos que soportan la intención de establecer los sistemas de créditos en la UNAN- León, razón por la cual nace nuestra inquietud de realizar este trabajo para contribuir con el desarrollo a través de Software Libre del módulo “Planificación Académica” enfocado a la Facultad de Ciencias con el objetivo de facilitar la organización y planificación de los horarios de aulas, laboratorios, profesores así como la información que generan los pénsum académicos de cada una de las carreras que se ofertan en la Facultad.



4. OBJETIVOS

OBJETIVO GENERAL:

- ❖ Automatizar el módulo de planificación académica de la Facultad de Ciencias (UNAN-León) implementado en sistemas de créditos académicos, a través de Software Libre.

OBJETIVOS ESPECÍFICOS:

- ❖ Facilitar la gestión de los Pénsum académicos de la Facultad de Ciencias.
- ❖ Facilitar la planificación eficaz de los horarios de las aulas de clase, laboratorios y docentes de la Facultad de Ciencias.
- ❖ Instalar y configurar los Software a utilizar: Linux Debian, PostgreSQL, Servidor Apache, Eclipse, Quanta Plus, PHP.Net.
- ❖ Crear una interfaz de usuario amigable e intuitiva que permita facilidad de uso del sistema y que provea ayuda al usuario.
- ❖ Generar reportes de: pénsum académicos, horarios de ocupación de las aulas de clase y laboratorios, así como de la carga académica de los docentes.



5. MARCO TEÓRICO

5.1 Sistema de Créditos: Es un “sistema que facilita la valoración y comparación de los resultados del aprendizaje en el contexto de las distintos programas y entornos de aprendizaje” incluyendo el aprendizaje en el aula y fuera de ella. Se ha convertido en un mecanismo importante para garantizar la flexibilidad en la formación de los estudiantes.

El sistema de Créditos ofrece las siguientes ventajas:

1. Fomenta la autonomía del estudiante para elegir sus propias actividades formativas, que respondan a sus intereses y motivaciones.
2. Ofrece mayor libertad y autonomía al estudiante.
3. Permite mayor flexibilidad en cuanto a:
 - El tiempo, ya que el estudiante puede ajustarse a sus condiciones y capacidades.
 - El Currículo, ya que el estudiante podrá seleccionar los componentes curriculares según su interés.
 - La posibilidad de egreso, pues se determinan el número y tipo de créditos requeridos para egresar en distintos niveles.
 - La movilidad, dando al estudiante la oportunidad de tomar clases, cursos, talleres o prácticas en otras instituciones y permitiendo los traslados, la convalidación y el reconocimiento de los estudios realizados.
4. Pueden disminuir los costos de operación al compartir los recursos existentes.
5. Aumenta la oferta académica en función de las operaciones terminales disponibles.
6. Facilita la situación de los alumnos ante los cambios que puedan introducirse en un currículo.
7. Limita la frondosidad curricular.
8. Favorece la integración universitaria en el sistema centroamericano e iberoamericano.
9. Facilita la homologación de títulos y estudios.



“El instrumento más ampliamente aceptado a nivel mundial para el reconocimiento de aprendizajes y logros es el crédito académico”.

El crédito se define como una unidad valorativa de trabajo académico del estudiante. Para calcular el número de créditos de cada componente curricular, debe tomarse en cuenta el tiempo presencial, el estudio independiente y demás actividades que integran el proceso de enseñanza-aprendizaje, de acuerdo con los objetivos educativos y el tiempo requerido para alcanzarlos. Los créditos debe expresarse en números enteros.

De acuerdo a la definición anterior, el valor del crédito se puede expresar de la siguiente manera: un crédito equivalente a 3 horas por semana de trabajo del estudiante, durante 16 semanas (dado que el semestre en la UNAN-León se compone de 16 semanas) o sea, un crédito equivale a un total de 48 horas de trabajo en Componente Curricular por semestre.

De aquí se deriva que:

$$\text{No de Créditos} = \frac{\text{No. de horas por semana} \times \text{No. de semanas por semestre (16)}}{48 \text{ Horas}}$$

¿Cómo calcular el total de créditos para programas académicos en América Latina?

Para calcular el valor total de créditos SICA de un programa académico en América Latina se propone el siguiente procedimiento. En primer lugar se toma en consideración una carga semanal normal de un estudiante, que valore su tiempo de dedicación al programa, y respete sus posibilidades de tiempo libre. Dicho valor se multiplica por las semanas académicas de estudio previstas por el programa. El resultado de la multiplicación anterior se multiplica nuevamente por el número de semestres académicos previstos por el programa, para finalmente llegar a un número total de horas de trabajo por parte del estudiante previsible en el programa. El resultado anterior se divide entre la valoración del SICA para llegar a un número total de créditos SICA para América Latina del mencionado programa.



**Planificación Académica para la Facultad de Ciencias
Tesis para optar al Título de Ingeniero en Sistemas de Información**

A continuación se presenta un ejemplo utilizando la equivalencia de 1 crédito a 32 horas de trabajo total por parte del estudiante. En este caso, se presume el período académico es semestral, situación que bien podría ser distinta para casos particulares, sin embargo, corresponde a la mayoría de programas en América Latina.

Tabla No.1. Cálculo del número total de horas de un Programa Académico típico²

Carga semanal promedio y razonable para que un estudiante lleve adelante sus procesos de aprendizaje (Docencia directa y trabajo independiente)	48 horas
Total semanas de trabajo por parte de un estudiante por semestre	15-20 semanas por semestre
Total Semestres de trabajo por parte de un estudiante	8-10 semestres
Total previsible de semanas en un programa académico	10-13 semestres Medicina 120-200 semanas
Total previsible de horas de trabajo del estudiante para un programa	150-260 semanas Medicina 5760-9600 horas
	7200-12480 horas Medicina

**Tabla No.2. Cálculo del número total de créditos SICA de un Programa Académico típico
(1 crédito SICA=32 horas de trabajo del estudiante).³**

1 CREDITO SICA = 32 HORAS DE TRABAJO TOTAL DEL ESTUDIANTE⁴	
Créditos Totales del programa en una definición SICA-AL de 1crédito=32 horas	180-300
Créditos Anuales del programa en una definición SICA-AL de 1crédito=32 horas	225-390 Medicina 45-80 ⁵
	45-80 ⁶ Medicina

Como puede observarse el SICA para América Latina permite una fácil aproximación de algunos modelos de créditos académicos en AL, pero adicionalmente permite la aproximación al modelo ECTS. La propuesta se asemejaría al modelo ECTS y permitiría una fácil homologación al sistema europeo de educación superior.

Este mismo ejercicio podría desarrollarse para los programas de Maestrías y Doctorados típicos para tener un estimado del rango de créditos aplicable.



¿Cómo calcular los créditos de una asignatura para programas académicos en América Latina?

Para este cálculo es necesario considerar durante un período académico la totalidad de las horas de trabajo por parte del estudiante, incluyendo el número total de horas presenciales para la unidad del curso (número de horas semanales x número de semanas). Dichas horas presenciales incluyen clases magistrales, talleres, seminarios, trabajo de laboratorio, tutorías, prácticas, u otras estrategias en las que existe un trabajo acompañado al estudiante. Por otro lado, se consideran las horas de trabajo independiente, que incluyen: estudio por parte del estudiante, investigación o trabajo personal, trabajo en grupos, u otro similar en los que no hay un trabajo acompañado al estudiante.

Dicho valor se divide entre la valoración del SICA7 a partir de lo cual se puede estimar el número de créditos previstos para dicha asignatura.

Tabla No.3. Cálculo del número de créditos SICA cuando 1 crédito SICA=32 horas de trabajo del estudiante.

Asignatura: XXXXXX	
Número total de horas presenciales acompañadas por un docente en el periodo académico para la Asignatura XXXXXX.	64
Número total de horas de trabajo independiente por parte del estudiante en el periodo académico para la Asignatura XXXXXX	128
Total de Horas de trabajo por parte del estudiante en el periodo académico para la Asignatura XXXXXX	192
Créditos de la Asignatura XXXXXX	6 créditos

Tabla No.3. Cálculo del número de créditos SICA cuando 1 crédito SICA=32 horas de trabajo del estudiante.

El SICA ofrece la posibilidad de contar con una cuantificación del trabajo académico (6 créditos) en un mismo orden de magnitud, lo que facilita su comparación entre distintas instituciones y países. Si la diferencia en los órdenes de magnitud entre diferentes sistemas es muy grande, la percepción es que existen muchas dificultades para su comparación, aunque sólo sea un problema de conversión numérica.



Pero el proceso de identificación del número de créditos no puede reducirse a la conversión mecánica de las horas que cada componente curricular tiene plasmadas en el plan de estudio actual, sino que amerita un mayor análisis en el que se tome en cuenta la complejidad del componente.

Para este análisis se requiere reconocer el tipo de actividades que se realizan en cada uno de los componentes curriculares y ubicarlas dentro de las siguientes categorías:

- **Actividades eminentemente teóricas:** Estas requieren que el estudiante dedique un determinado tiempo para realizar estudio o trabajo adicional, es el caso de las clases teóricas, seminarios, foros, entre otras. Para estas actividades, por cada hora presencial deberán calcularse dos horas de trabajo independiente (no presencial).
- **Actividades eminentemente prácticas:** Son las que no requieren trabajo adicional del estudiante, en esta categoría se encuentran las actividades físicas, artístico-culturales, comunitarias, etc. En este caso, sólo se considerarán las horas presenciales.
- **Actividades teórico-prácticas:** Éstas requieren que el estudiante dedique períodos variables de tiempo para el trabajo independiente, es el caso de las prácticas clínicas, de laboratorio, etc. Para este tipo de actividades, el docente deberá valorar la cantidad de tiempo, que requieren los estudiantes para el estudio o trabajo adicional y asignará por cada hora presencial, de 0.5 a 2 horas de auto-estudio.

Para completar su plan de estudio, un estudiante deberá cursar un número determinado de componentes obligatorios, electivos, optativos:

- **Obligatorios:** Los que no pueden faltar en su formación profesional. 60% de total de los créditos.
- **Electivos:** Cursos de formación general, cursos que permiten al estudiante profundizar determinados contenidos o dar una orientación específica a su carrera, que el estudiante deberá seleccionar entre las opciones que se le presenten. 30%.
- **Optativas:** Cursos que el estudiante podrá seleccionar de entre todos los componentes que se ofrecen en la Universidad u otras instituciones de Educación



Superior, pudiendo ser incluso cursos de otras carreras diferentes a la que cursa. 10%.

5.2 Sistema: Es una combinación de medios (como personas, materiales, equipos, software, instalaciones, datos, etc.), integrados de forma tal que puedan desarrollar una determinada función en respuesta a una necesidad concreta". También se puede definir como un "Conjunto de elementos interrelacionados entre sí, de forma tal que, un elemento afecta el comportamiento de todo el conjunto".

5.3 El World Wide Web (W. W. W.): Es un sistema hipermedia distribuido, accesible a través de Internet, que permite navegar con facilidad por una gran cantidad de información. Las empresas están utilizando el Web e Internet para comunicarse con clientes y proveedores mediante la publicación de contenidos tales como información técnicas de sus productos, de sus servicios y publicidad. La educación, también se puede impartir como un proveedor de servicio por lo que pueden utilizar esta tecnología para comunicarse con sus "alumnos" y "profesores" y demás usuarios que formen parte del sistema.

5.4 Interfaces: Son el resultado de la programación que nos permite definir un conjunto de miembros que describen una determinada funcionalidad. Lo importante es que tan sólo la describen, pero no proporcionan implementación alguna de los mismos ya que ésta es una tarea de las clases de objetos, que son las que pueden contener código. Las interfaces se dividen en estáticas que no tienen cambios y son difíciles de modificar y las interfaces dinámicas que cambian de acuerdo a los requerimientos establecidos. Normalmente las interfaces de usuarios son utilizadas para proporcionar información o resultados que genera un sistema de cómputo.

5.5 Interfaz de usuario: Es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina, también es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario, es responsable de solicitar comandos al usuario, y de desplegar los resultados de la aplicación de una manera comprensible.



5.6 Elementos de las Interfaces del sistema: En un interfaz intervienen 3 elementos importantes de los que al interactuarse entre ellos se crea una comunicación:

- Humano
- Computadora
- Interacción

5.7 Un sitio con páginas Web dinámicas: Es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

5.8 Clave foránea: En una tabla de una base de datos, se corresponde con un campo de otra tabla que es clave primaria en ella. Las claves foráneas se utilizan para establecer relaciones entre las tablas de una Base de Datos.

5.9 Base de Datos: es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

5.10 Software Libre: se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.



Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar.

El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debes tener la libertad de distribuir estos formatos si encontraras o desarrollaras la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre.

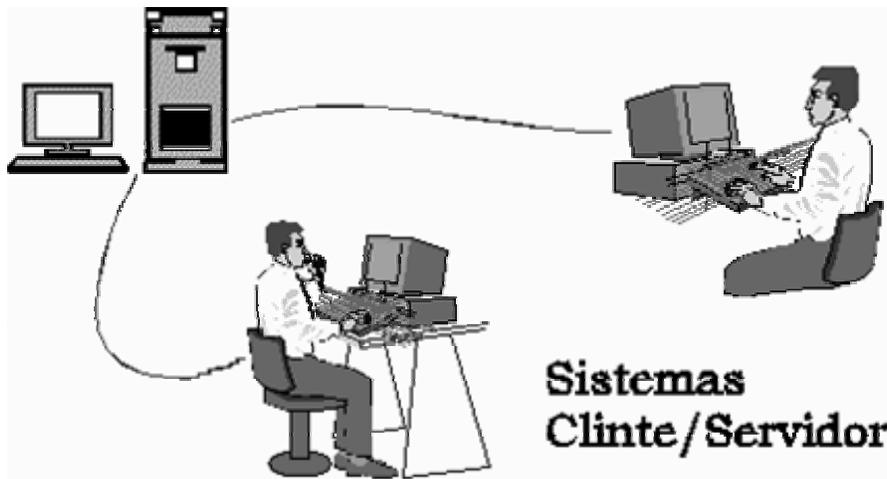
Para que estas libertades sean reales, deben ser irrevocables mientras no hagas nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no le hayas dado motivos, el software no es libre.



Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir software libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, *copyleft* ["izquierdo de copia"] (expresado muy simplemente) es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

“Software libre” no significa “no comercial”. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

5.11 Estructura Cliente-Servidor: En las redes basadas en estructuras cliente-servidor, los servidores ponen a disposición de sus clientes recursos, servicios y aplicaciones. Dependiendo de qué recursos ofrece el servidor y cuales se mantienen en los clientes se pueden hacer distinciones entre distintas estructuras cliente-servidor.





5.12 SOFTWARE Y HERRAMIENTAS PARA LA IMPLEMENTACIÓN DEL SISTEMA

5.12.1 PHP

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

La definición oficial del lenguaje PHP dice que esto son las siglas de "PHP: Hypertext Preprocessor". Se trata de un lenguaje cuyos programas se ejecutan en la parte del servidor, y cuyo código escribiremos incrustado con el código HTML. Dado que se ejecuta en el servidor, será necesario pues que tengamos un servidor Web para poder comprobar nuestros programas.

PHP es el lenguaje más parecido a C para quienes hayan estudiado programación C, para quienes no hayan estudiado este lenguaje decirles que este tipo de programación consiste en un conjunto de sentencias, variables y palabras clave con el que se pueden hacer multitud de cosas a nivel del servidor, desde escribir un código HTML u otro en función del valor de una variable hasta el manejo de bases de datos, por citar algunos ejemplos.

5.12.1.1 Historia

PHP (Hypertext Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), que desarrolla páginas Web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código abierto, es el más popular y extendido en la Web.



PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones y fue desarrollado originalmente por Rasmus Ledford en 1994 como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y consiguió un relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Cuando Rasmus tuvo la necesidad de crear páginas dinámicas que trabajasen con formularios, creó una serie de etiquetas a las que denominó “Form Interpreters”, y lo sacó al público con el nombre de PHP/FI en 1995. Luego salió la versión mejorada, llamada PHP/FI 2.0.

Zeev Suraski y Andi Gutmans programaron el analizador sintáctico incluyendo nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales, como MySQL y Postgre SQL, así como un módulo para Apache. Con estas mejoras surgió PHP3 en 1997. Este analizador define la sintaxis y semántica de la versión PHP3 y la siguiente: PHP4.

PHP3 carecía del uso de sesiones, algo muy común en las páginas Web de cierta complejidad. En el año 2000, PHP3 evolucionó a PHP4, que utiliza el motor Zend (desarrollado por Zeev y Andi encargado de interpretar el código fuente de los scripts de PHP), desarrollado para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su mayor independencia del servidor Web y su rapidez, ya que primero se compila y luego se ejecuta, mientras que antes se ejecutaba a la vez que se interpretaba el código.

La última versión es PHP5, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET.



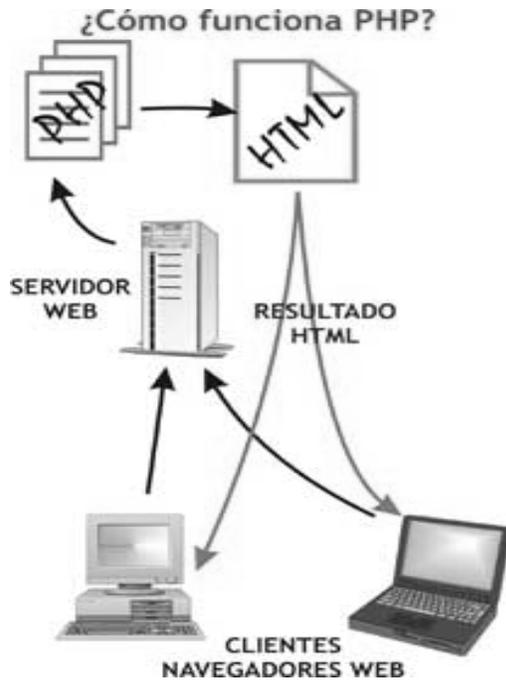
5.12.1.2 Características

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Las principales características de PHP son:
 - Rápidez.
 - Su facilidad de aprendizaje.
 - Su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos.
 - Se distribuye de forma gratuita bajo una licencia abierta.



5.12.1.3 ¿Cómo funciona PHP?



A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una Base de Datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap).

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP.

El modo de operación de PHP es el siguiente:

- El Navegador realiza una petición al servidor (se escribe la URL).
- Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.
- Por último el Navegador muestra la respuesta del servidor.



5.12.1.4 Usos de PHP

Los principales usos del PHP son los siguientes:

- Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS X.

5.12.1.5 Ventajas del uso de PHP

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con JavaScript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.
- PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.



- Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
- El código PHP es mucho más legible que el de PERL, todo el que haya programado PERL podrá corroborar esta afirmación.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF, etc.)
- Al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente.
- Esta siendo utilizado con éxito en varios millones de sitios Web.
- Hay multitud de aplicaciones PHP para resolver problemas concretos (tiendas virtuales, periódicos, etc.) listas para usar.
- Es multiplataforma, funciona en todas las plataformas que soporten Apache como servidor.
- Es software libre. Se puede obtener en la Web y su código esta disponible bajo la licencia GPL.

5.12.1.6 Desventajas del uso de PHP

- Todo el trabajo lo realiza el servidor y no deja nada al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La orientación a objetos es aún muy deficiente para aplicaciones grandes.
- Seguridad: PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.
- PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta



combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

5.12.2 Herramientas HTML

HTML (HyperText Markup Language), es un lenguaje para la definición de estilos lógicos en documentos de hipertexto, siendo el medio principal para la difusión de información en World Wide Web. HTML es un estándar que nos sirve para crear páginas Web, fue el primer lenguaje en surgir, posteriormente surgieron el XML y el XHTML.

El HTML es una aplicación SGML (Lenguaje de Etiqueta Generalizado Estándar) conforme al estándar internacional ISO 8879.

Desde inicios del HTML, ha habido una constante invención de nuevos elementos para ser usados dentro de HTML (como estándar) y para adaptar HTML a mercados verticales, altamente especializados. Esta cantidad de nuevos elementos ha llevado a problemas de compatibilidad de los documentos en las distintas plataformas. Su interpretación depende del software utilizado. (Netscape, Internet Explorer, Opera, Firefox, etc....) y puede ser utilizado por cualquier lenguaje y/o sistema operativo.

Los documentos HTML se dividen en tres partes bien diferenciadas:

- La Cabecera de tipo de documento, la usa el software para saber la versión de HTML que se está usando (no visible).
- La Cabecera del documento (document HEADER), usada para dar información sobre el documento (no visible).



- El Cuerpo del documento (document BODY), es la parte principal del documento y la parte que el usuario ve.

Estas tres partes pueden verse en la plantilla mostrada a continuación.

```
<HTML>
```

```
  <HEAD>
```

```
  </HEAD>
```

```
<BODY>
```

```
</BODY>
```

```
</HTML>
```

No hay que compilar el código para ver si funciona. Compilar significa que el programa verifica que el código escrito por el usuario sea parte del lenguaje de programación (C, Pascal, Cobolt, etc.) y luego escribe en lenguaje entendible para los computadores, programa ejecutable. Un programa ejecutable no necesita de otro programa para funcionar (C, Pascal, Cobol, etc.), funcionando en forma independiente como es el caso del word.exe o el excel.exe, que son los programas ejecutables del Word y de Excel.

El código que se escribe no lanza un error, el browser despliega la información como la interpreta, en el peor caso no muestra la información.

“El lenguaje es relativamente pequeño y fácil de usar, permitiendo a los usuarios aprenderlo en un corto tiempo y comenzar a practicarlo en forma inmediata”.



5.12.3 Servidor Apache

La historia de Apache se remonta a febrero de 1995, donde empieza el proyecto del grupo Apache, el cual esta basado en el servidor Apache httpd de la aplicación original de NCSA. El desarrollo de esta aplicación original se estancó por algún tiempo tras la marcha de Rob McCool por lo que varios Web master siguieron creando sus parches para sus servidores Web hasta que se contactaron vía e-mail para seguir en conjunto el mantenimiento del servidor Web, fue ahí cuando formaron el grupo Apache.

Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores.

Apache surgió a partir del servidor de HTTP más famoso y difundido en su época: NCSA. Desde entonces se convirtió en un poderoso rival de todos los servidores Unix utilizados hasta la fecha por su eficiencia, funcionalidad y rapidez. Es por ello que se conoce como el rey de los servidores Web. Se desarrolla de forma estable y segura gracias a la cooperación y los esfuerzos de un grupo de personas conocidas como grupo Apache (Apache Group), los cuales se comunican a través de Internet y del Web. Juntos se dedican a perfeccionar el servidor y su documentación regidos por la ASF (Apache Software Foundation).

Apache es el servidor Web líder en el mercado. Su costo es gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

5.12.3.1 ¿Qué es el servidor Apache?

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un



conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor parcheado). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

El servidor Apache (<http://www.apache.org>) es el servicio que se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de Internet http. En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios.

El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor Web, este término podría referirse a la máquina que almacena y maneja los sitios Web, y en este sentido es utilizada por las compañías que ofrecen hospedaje. Alternativamente, el servidor Web podría referirse al software, como el servidor de HTTP de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas Web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.



Como su nombre implica, un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor. En la WWW, un servidor Web es un ordenador que usa el protocolo http para enviar páginas Web al ordenador de un usuario, cuando el usuario las solicita.

Los servidores Web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar el Internet. Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo. Los servidores se conectan a la red mediante una interfaz.

5.12.3.2 Características del servidor Apache

Dentro de las características de Apache tenemos las siguientes.

- Es un servidor Web potente, flexible y ajustado al HTTP/1.1
- Es altamente configurable y extensible.
- Puede ser ajustado a través de la definición de módulos empleando su propio API (Application Programming Interface).
- Provee todo su código fuente de forma libre y se distribuye bajo una licencia no restrictiva.
- Se ejecuta en diversas plataformas operativas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes: IRIX, Solaris, HPUX, SCO, FreeBSD, NetBSD, AIX, Digital Unix, etc.
- Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
- Apache significa "A PATCHy sErver", o sea se basa en un código y un conjunto de ficheros "parches". Otros desarrolladores relacionan su nombre con el de las tribus nativas americanas de Apaches.
- Bases de datos DBM para autenticación.



- Permiten establecer fácilmente la protección de documentos a través de passwords para una gran cantidad de usuarios sin dañar el funcionamiento del servidor.
- Respuestas adaptables a los errores o problemas.
- Se pueden definir ficheros o scripts de tipo CGI1.3 que respondan ante la ocurrencia de errores internos o en las solicitudes realizadas.
- Directiva para definir múltiples índices.
- Se utiliza cuando se solicitan directorios por parte de los clientes a partir de lo cual se puede buscar en estos y devolver un documento índice cuyo nombre puede ser por ejemplo: index.HTML, index.cgi o default.HTML.
- Ilimitadas y flexibles posibilidades de redireccionamiento y definición de alias para los URLs.
- Apache no tiene un límite establecido para definir alias y redireccionamientos que pueden ser declarados en sus ficheros de configuración.
- Negociación del contenido de las respuestas.
- Apache es capaz de ofrecer la mejor representación de la información accedida de acuerdo con las capacidades del cliente solicitante.
- Soporte de hosts virtuales.
- Es la habilidad del servidor de distinguir entre los pedidos hechos a diferentes direcciones IP o nombres de dominio definidos en la misma máquina.
- Configuración flexible de las trazas generadas.
- Es posible adaptar el formato de las trazas obtenidas así como redireccionarlas a través de tuberías (Unix) en aras de filtrarlas. De esta forma se puede lograr por ejemplo dividir dinámicamente las trazas de los hosts virtuales en distintos ficheros.

5.12.3.3 Ventajas del uso del Servidor Apache

Algunas de las ventajas son las siguientes:

- Se pueden extender las características de Apache hasta donde nuestra imaginación y conocimientos lleguen.



- El Apache soporta Dinamic Shared Object (DSO). Gracias a ello se pueden construir módulos que le den nuevas funcionalidades que son cargadas en tiempos de ejecución.
- Apache puede facilitar información en varios formatos para que un determinado cliente pueda interpretarla.
- Apache permite la creación de sitios Web dinámicos mediante:
 - El uso de CGI's.
 - El uso de Server Side Includes (SSI).
 - El uso de lenguajes de Scripting como PHP, JavaScript, Python.
 - El uso de Java y páginas jsp.

5.12.4 Internet Information Server IIS

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: **FTP, SMTP, NNTP y HTTP/HTTPS**. Servidor conectado a Internet que puede contener paginas web, servicio de correo electrónico etc.

Este servicio convierte a un computador en un servidor de internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.



Existen otros servidores web que pueden usarse como alternativa a este que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software Libre y otros muchos.

Versiones

- * IIS 1.0, Windows NT 3.51 Service Pack 3
- * IIS 2.0, Windows NT 4.0
- * IIS 3.0, Windows NT 4.0 Service Pack 3
- * IIS 4.0, Windows NT 4.0 Option Pack
- * IIS 5.0, Windows 2000
- * IIS 5.1, Windows XP Professional
- * IIS 6.0, Windows Server 2003 y Windows XP Professional x64 Edition
- * IIS 7.0, Windows Vista y Windows Server "Longhorn"

5.12.5 Apache vs. IIS: Dos servidores a elegir

El mercado está dividido (simplificando un poco) entre IIS (Internet Information Server) de Microsoft, y Apache, un proyecto libre de la Fundación Apache, gratuito y de código abierto. Si atendemos a las estadísticas, el mundo de los usuarios se decanta claramente por lo segundo.

“Apache es usado por aproximadamente 65% de los servidores, mientras que IIS es utilizado por aproximadamente el 30%”.

- IIS va por su versión 6, y, por supuesto, sólo funciona bajo servidores Microsoft. Debe ser usado bajo licencia, esto es, hay que pagar por él, con esto te ofrecen un servicio técnico y cierto tipo de garantías, como soporte ante fallos de seguridad y funcionamiento, disponiendo de parches con cierta tranquilidad.
- Apache es un proyecto libre, y por tanto, gratuito. Existen dos vertientes que desarrollan paralelamente, las versiones 1.3.x y 2.0.x. De hecho la más usada es la 1.3.x. No necesita de licencias, y no se recibe soporte técnico. Eso sí, está demostrado que ante los problemas de seguridad, actúan más rápido que Microsoft, poniendo a disposición de los usuarios hotfixes que solucionan el problema en menos tiempo.



Pero, centrándonos en la seguridad, sabemos que la responsabilidad ante un desastre (violación de la seguridad, entrada en los sistemas, defacement, denegación de servicio...) se reparte a partes iguales entre el servidor Web, el sistema operativo y el administrador del sistema. La seguridad en el servidor Web está íntimamente ligada a la del sistema operativo, sobretodo en sistemas Windows. Estadísticamente, el número de incidentes de seguridad sufridos por sistemas funcionando con productos de Microsoft es muy superior al de los ataques perpetrados contra sistemas de código abierto como Linux o Unix. En torno al 60% de los desfiguramientos de páginas Web se producen en servidores con Windows NT o 2000 como sistema operativo en contraste con el uso desigual de cada uno. El porqué de este dato puede tener varias explicaciones. Tradicionalmente se tiene la idea de que la compañía de Bill Gates comete grandes errores relacionados con la seguridad y que suelen ser lentos e incluso ineficaces a la hora de resolverlos. Esto hace que los hackers maliciosos lo vean como objetivo claro y apetecible para cometer sus fechorías. Esta tendencia varía con el tiempo. Ha llegado a ser tan "fácil" pone en entredicho la seguridad de un sistema Microsoft, que parece que los "profesionales" del ramo han decidido ignorarlos, y proponerse nuevos retos que motiven y enriquezcan más sus conocimientos.

También se puede hablar de la leyenda de los tipos de administradores. Se dice que si usan sistemas de código abierto, de alguna forma y sólo en teoría, más "difíciles" de manejar y configurar, es que conocen mejor su sistema y pueden aprovechar de manera más eficaz las innumerables ventajas del código abierto. Los usuarios de Windows, deben permanecer más atentos al "parcheo" de sus sistemas, y su configuración, al ser más intuitiva y sencilla, puede dar pie a despistes fatales.

Si nos centramos en el uso que se hace en Internet de los servidores Web, Apache resulta ser el claro ganador ante todas las otras alternativas, pues ha demostrado con creces desde su nacimiento su estabilidad, solidez y rendimiento superiores. En todo caso, Apache supone una clara alternativa para el que pretenda trastear con este mundo de los servidores Web, pues sólo tiene que descargarlo y comenzar a usarlo en su propio sistema, con la tranquilidad de no estar quebrantando ningún tipo de ley. Por supuesto también está disponible para sistemas Windows, con una facilidad de manejo que asombraría a más de uno. Nada más instalarlo ya está listo para ser usado, luego, si se quiere poner a punto su configuración, sólo es necesario modificar un fichero de texto que viene perfectamente documentado. Todo esto en menos de dos megas.

IIS, sin embargo, es un poco más intuitivo (cuadros de diálogo y ventanas) pero a costa de una integración total en el sistema que hace difícil distinguir qué opciones afectan al servidor y cuáles al sistema operativo. También hay que decir que ocupa muchos más megas en el disco duro, y su uso sin licencias es ilegal.

Si el factor decisivo para decantarse por uno u otro pudiese ser el soporte técnico, Apache lo solventa con una gran comunidad altruista disponible en foros, canales de IRC y servidores de noticias, donde existen miles de usuarios dispuestos a echar una mano cuando surge algún problema.



5.12.6 Servidor Web

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso. En este artículo nos referiremos siempre a la aplicación.

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. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear `www.wikipedia.org` en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio Web clásico podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:



* **Aplicaciones en el lado del cliente:** el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

* **Aplicaciones en el lado del servidor:** el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional.

Algunos conceptos relacionados con las aplicaciones Web son:

- * PHP
- * ASP
- * Perl
- * CGI
- * .NET
- * JSP (Tecnología Java)

Algunos servidores Web importantes son:

- * **Apache**
- * **IIS**
- * **Cherokee**

Otros servidores, más simples pero más rápidos, son:

- * lighttpd
- * thttpd



5.12.7 JAVASCRIPT

JavaScript es una de las múltiples aplicaciones que han surgido para extender las capacidades del Lenguaje HTML. JavaScript es un lenguaje script orientado a documento. No sirve para crear programas, sino para agregarles características a nuestra Web.

Las normas para poder escribir cualquier código de JavaScript se basan en algunos puntos básicos y que debemos cumplir siempre. Estas normas son las siguientes:

- Todo el código (sentencias) esta dentro de funciones.
- Las funciones se desarrollan entre las etiquetas `<script>` y `</script>`.
- Las etiquetas “`<script>`” deben colocarse entre las etiquetas `<head>` y `</head>`.
- Las etiquetas “`<title>`” no pueden estar colocadas entre las de “`<script>`”.
- La llamada a la función se hace a través de un evento de un elemento del documento.

Las funciones son un conjunto de sentencias (bloque de código) que especifica al programa las operaciones a realizar. Son útiles para evitar la repetición de líneas y modular el código. Para trabajar con ellas hay que desarrollarlas y llamarlas cuando lo necesitemos.

Sintaxis del desarrollo:

```
function nombre_funcion([var1,var2,varN])  
{  
sentencia(s);  
}
```

Sintaxis de la llamada:

```
<elemento evento=nombre_funcion([val1,val2,valN]);>  
nombre_funcion(valor1,valor2,valorN);
```

En el primero de los casos la llamada se realiza desde un elemento del documento. En el segundo caso la llamada se realiza desde el interior de otra función que también es posible.



5.12.8 El Archipiélago Eclipse

5.12.8.1 ¿Qué es la plataforma Eclipse?

Una definición se puede resumir así: "[es] una plataforma universal para integrar herramientas de desarrollo" con una "arquitectura abierta, basada en plug-ins". Plataforma universal, pues emplea una estructura abierta de plug-ins (extensiones; to plug in significa conectar y plug, enchufe o conector) que permite expandir las capacidades de la plataforma base hasta el infinito. Arquitectura abierta, en definitiva, porque es un producto de código fuente abierto u open source.

Desde el punto de vista del usuario que le eche un vistazo por vez primera, la plataforma Eclipse resulta ser un IDE (entorno de desarrollo integrado) de código fuente abierto, la mayor parte del cual ha sido escrito en Java.

5.12.8.2 Eclipse y el software open source. Un matrimonio de conveniencia bien avenida

Eclipse se distribuye actualmente bajo la licencia CPL (Common Public License o Licencia) versión 1.0 de IBM, aprobada por la organización Open Source Initiative (OSI). A diferencia de otros proyectos open source (o, más exactamente, free software) que no permiten que se deriven de ellos trabajos propietarios o cerrados, Eclipse puede extenderse al estar bajo esta licencia CPL- mediante la inclusión de plug-ins propietarios o ser usado como base para la creación de nuevas herramientas y, tras reempaquetarse y compilarse el código resultante, el producto final puede venderse de forma comercial, manteniéndose público el código de Eclipse utilizado o modificado, pero sin la obligación de poner a disposición del público el nuevo código añadido (este último puede ir bajo licencia que se desee).

Como es bien sabido, el software propietario o cerrado se caracteriza porque su redistribución y modificación está prohibida o requiere autorización previa; la mayor parte del software comercial es propietario, pero no cabe identificar ambos tipos de software.



Se pueden obtener beneficios económicos de Eclipse (al igual que de cualquier otro proyecto de código fuente abierto o de software libre). Al igual que cualquier licencia autorizada o admitida por la OSI, la licencia CPL exige el cumplimiento de una serie de requisitos, algunos de los cuales figuran a continuación:

- Distribución gratuita: Cualquier software bajo licencia CPL puede distribuirse libremente, permitiéndose la venta sin que se requiera el pago de royalties o compensaciones de cualquier tipo.
- Cualquier programa bajo licencia CPL debe permitir la distribución en forma de código fuente y en forma compilada. Si el producto no incluye el código fuente, deberá incluirse en él la manera de conseguirlo.
- Cualquier programa bajo licencia CPL debe permitir la producción de trabajos derivados a partir de él y la introducción de modificaciones en el programa original.
- Un programa difundido bajo licencia CPL puede ser distribuido por cualquiera en forma compilada o de código fuente.

Un elevado porcentaje del éxito de Eclipse y de las mejoras continuas que experimenta se debe a la naturaleza de su licencia: la licencia CPL de IBM supone ventajas comerciales frente a licencias como la GNU GPL, las cuales impiden que se deriven o incorporen trabajos propietarios como es bien sabido, los objetivos de las empresas con ánimo de lucro, aunque a algunas les cause sarpullidos reconocerlo públicamente, se fundamentan en dos reglas:

- 1) Gane dinero y mantenga su propiedad;
- 2) Nunca olvide la primera regla (eso sí, cada una las implementa como puede)-.

Muchas empresas (grandes, PYMEs) pueden desarrollar plug-ins propietarios o sus propias herramientas derivadas de Eclipse y obtener beneficios de su trabajo sin ver mermadas sus ganancias por el pago de royalties, y los desarrolladores pueden planear con rapidez sus propias extensiones, modificaciones o mejoras, a la vista del código fuente de Eclipse y de los productos derivados bajo licencia CPL. Individuos y empresas pueden trabajar en simbiosis, lograr sus objetivos, contribuir a la mejora continua de Eclipse y ofrecer mejores productos (más competitivos en prestaciones y precio) a los consumidores finales.



Al usuario final poco le importa que el gato sea blanco, negro, pardo o el pedigrí de sus progenitores: lo importante es que cace ratones. Y que los cace bien.

Aparte de las distintas licencias de Linux y Eclipse, hay también otro rasgo diferenciador entre ambos proyectos que contribuye a la vertiginosa expansión de Eclipse: poca gente (comparativamente hablando) tiene conocimientos de programación de sistemas operativos; sin embargo, cualquier desarrollador usuario de Eclipse -y hay muchos más desarrolladores que expertos en sistemas operativos- es un potencial colaborador del proyecto Eclipse.

Eclipse puede considerarse, en justicia, como un IDE para Java, una plataforma de integración de herramientas de desarrollo y un framework de aplicaciones.

5.12.8.3 La arquitectura del SDK de Eclipse: una vista aérea

El Standard Development Kit (Kit de desarrollo estándar) de Eclipse se compone de tres elementos:

La Plataforma Eclipse (cuya arquitectura interna se describirá más adelante).

El JDT (Java Development Tooling, las herramientas de desarrollo Java).

El PDE (Plug-in Development Environment, el entorno de desarrollo de plug-ins).

Tal y como ya se explicó, su desarrollo y mejora está en manos del proyecto Eclipse (subproyecto de Eclipse).

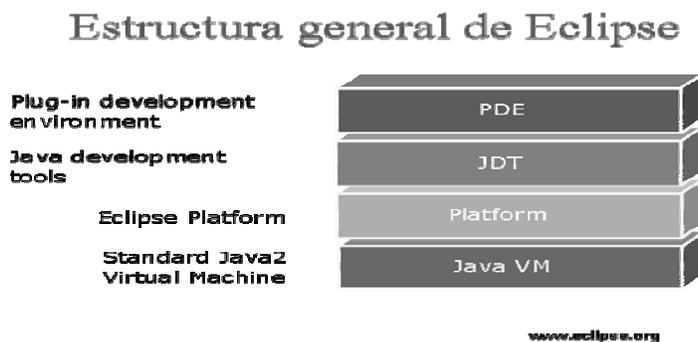


Fig. 4. Estructura general de Eclipse. Extraído de la documentación oficial de Eclipse



En esencia, la plataforma Eclipse es una plataforma para el desarrollo general de herramientas (recordemos: "un IDE para cualquier cosa y para nada en particular"). Por sí sola, la plataforma resulta de escasa utilidad para el usuario último pues se halla capacitada para trabajar con cualquier tipo de fichero (no necesariamente con ficheros asociados a lenguajes de programación, sino también con ficheros generados por aplicaciones como Word, ficheros de vídeo, de gráficos, etcétera), pero carece del conocimiento específico de cómo tratarlos.

Es decir, Eclipse puede mostrar un fichero C, por ejemplo, pero desconoce la gramática y sintaxis de C (palabras reservadas, bloques, etc.). La palabra main no significa más que la palabra vino para la plataforma aislada, pues no proporciona facilidades específicas para la depuración, edición, etc. La utilidad real de la plataforma por sí sola para el programador de C -o de cualquier otro lenguaje, incluido Java- no es mayor que la de un editor de texto plano (aunque con un extraordinario entorno gráfico alrededor).

Para el desarrollador de plug-ins y de IDEs, sin embargo, la situación adquiere un aspecto muy distinto: la plataforma por sí sola le proporciona un conjunto de frameworks, un conjunto de reglas de integración con la plataforma, una interfaz gráfica francamente espléndida, soporte para el control de versiones, infraestructura para la depuración independiente del lenguaje usado y el control de las bibliotecas gráficas, entre otras muchas características. Los desarrolladores de plug-ins e IDEs pueden usar todas estas funcionalidades ya incorporadas para desarrollar sus propias herramientas que expandan la plataforma.

Cuando se usa la plataforma Eclipse con plug-ins, empieza a vislumbrarse la potencia que ofrece a los usuarios no desarrolladores de plug-ins o IDEs. Los plug-ins explican a la plataforma cómo se deben tratar y gestionar los distintos tipos de archivos, y aumentan la funcionalidad del sistema resultante (o, dicho de otro modo, extienden o amplían la plataforma).



Para añadir nuevas capacidades o funcionalidades a la plataforma Eclipse se usan los puntos de extensión. Los puntos de extensión son, según la documentación oficial de Eclipse, "lugares bien definidos del sistema donde otras herramientas (llamadas plug-ins) pueden añadir funcionalidad".

De conformidad con la terminología orientada a objetos, un punto de extensión no deja de ser una interfaz que deberá ser implementada por cualquier desarrollador interesado en extender la plataforma. Conviene destacar un aspecto importante: el mecanismo de los puntos de extensión define el único modo de añadir nuevas funcionalidades a la plataforma.

Los plug-ins no sólo extienden o amplían la plataforma base, también pueden extender, a su vez, otros plug-ins que hayan definido sus propios puntos de extensión. Un plug-in puede hacer públicos interfaces que otros plug-ins pueden implementar.

Las implementaciones de los interfaces (llamadas extensiones) mostrados por los puntos de extensión se realizan típicamente en Java, aunque algunos puntos de extensión pueden acomodar extensiones proporcionadas por ficheros ejecutables nativos o componentes ActiveX; incluso pueden programarse en lenguajes de script.

El principal obstáculo con el cual se enfrentan las extensiones no realizadas en Java es la falta de acceso a la funcionalidad completa de la plataforma Eclipse. Por otro lado, los plug-ins sólo se cargan cuando son necesarios; así se evita disminuir innecesariamente el rendimiento de Eclipse. Tal y como se detalló, esta propiedad traza una clara separación, en cuanto a consumo de recursos, entre los IDEs comerciales y Eclipse. A diferencia de estos, Eclipse solo carga en memoria los plug-ins cuando los necesita.

Por ejemplo, el JDT agrupa un conjunto de plug-ins que extienden la plataforma al proporcionar características para la edición, compilación, depuración y ejecución de código Java (explica a la plataforma cómo entender los ficheros Java, en definitiva).



El JDT viene incluido en el SDK de Eclipse, pero resulta factible desarrollar otros plug-ins que permitan a la plataforma trabajar con otros lenguajes. Se encuentran ya disponibles plug-ins del consorcio Eclipse.org que proporcionan IDEs para C/C++ y COBOL.



Fig. 5. Arquitectura de los plug-ins de Eclipse. Traducido de la documentación oficial de Eclipse.

El Java Development Tooling (JDT) es, tal y como ya se ha escrito arriba, un conjunto de plug-ins que extienden la plataforma al proporcionar características para la edición, compilación, depuración y ejecución de código Java.

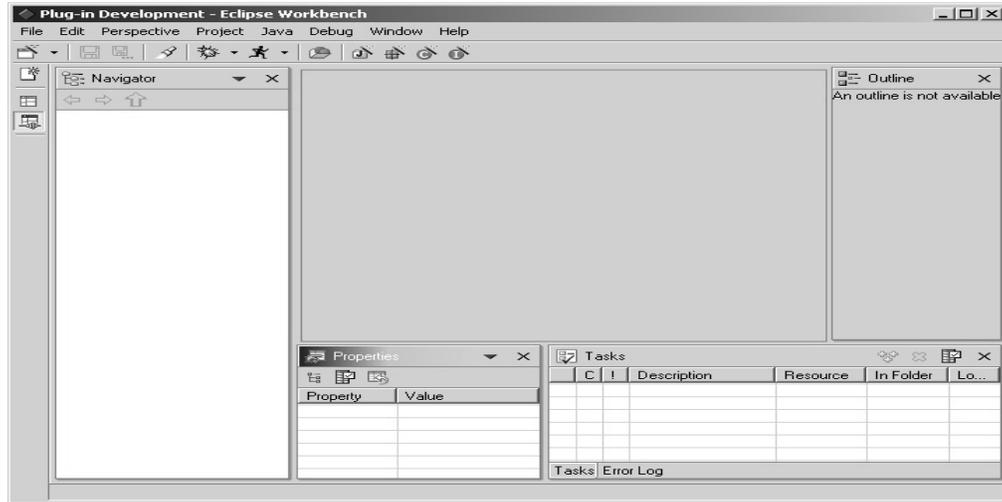


Fig. 6. Vista del PDE. Extraído de la documentación oficial de Eclipse

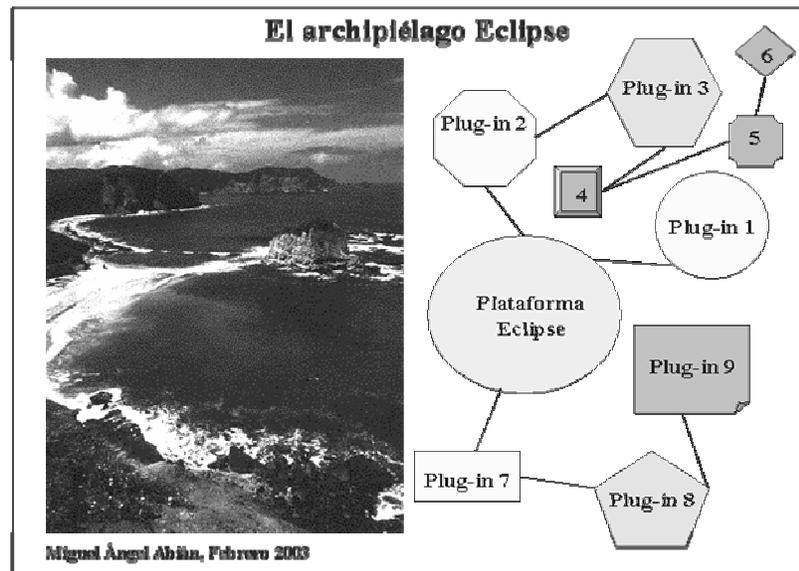


Fig. 7. Eclipse como archipiélago



5.12.9 PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y brevemente llamado Postgres95) está derivado del paquete Postgres escrito en la Universidad de Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

En 1996, se hizo evidente que el nombre "Postgres95" no resistiría el paso del tiempo. Elegimos un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL. Al mismo tiempo, hicimos que los números de versión partieran de la 6.0, volviendo a la secuencia seguida originalmente por el proyecto Postgres.

Durante el desarrollo de Postgres95 se hizo hincapié en identificar y entender los problemas en el código del motor de datos. Con PostgreSQL, el énfasis ha pasado a aumentar características y capacidades, aunque el trabajo continúa en todas las áreas.

Las principales mejoras en PostgreSQL incluyen:

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys).
- Disparadores (triggers).
- Vistas.
- Integridad transaccional.
- Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe).
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.



5.12.9.1 Inicio rápido con PostgreSQL

Para crear una base de datos, se utiliza el comando que viene con la distribución de PostgreSQL.

```
$ createdb 'nombre'
```

```
$ dropdb 'nombre'
```

```
$ createuser 'nombre'
```

```
$ dropuser 'nombre'
```

Los comandos anteriores permiten crear una base de datos, borrar, crear usuarios y borrar usuarios respectivamente. Una vez conectado en la base de datos, podemos efectuar las siguientes operaciones básicas.

```
prueba=> CREATE TABLE categorias ( Nombre varchar(20) );
```

```
prueba=> SELECT * FROM categorias;
```

```
prueba=> INSERT INTO categorias VALUES ('Misc');
```

```
prueba=> UPDATE categorias SET Nombre = 'Comestible' WHERE Nombre = 'Misc';
```

```
prueba=> TRUNCATE categorias;
```

Estos son algunos comandos que permiten crear una tabla 'categorias' con un solo atributo (en este caso 'Nombre'), luego realizar una consulta simple de la misma, insertar valores (tuplas o registros), cambiar el valor ingresado y por último borrar todo el contenido de la tabla.



5.12.9.2 Integridad referencial

PostgreSQL controla cualquier problema que pueda ocurrir por violación de la integridad referencial de una base de datos.

El siguiente ejemplo muestra un 'Foreign Key' utilizado para relacionar dos tablas y para evitar cualquier problema.

```
CREATE TABLE categorias ( Nombre varchar(20) PRIMARY KEY, Desc text );
```

```
CREATE TABLE productos (Nombre varchar(30) PRIMARY KEY, Categoria varchar(20)  
REFERENCES categoria, Desc text );
```

5.12.9.3 Configuración de PostgreSQL

1. `psql -t template 1 -U postgres`
2. **Configurar archivo : /etc/postgresql/pg_hba.conf**

```
# PostgreSQL Client Authentication Configuration File  
# =====  
.....  
#  
# TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD  
# Database administrative login by UNIX sockets  
# (Cambiar ident sameus por trust )  
  
#local all postgres ident sameus  
local all postgres trust  
#  
# All other connections by UNIX sockets  
local all all md5  
#  
# All IPv4 connections from localhost  
host all all 127.0.0.1 255.255.255.255 md5  
#  
# All IPv6 localhost connections  
host all all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff md5  
host all all ::ffff:127.0.0.1/128 md5  
#  
# reject all other connection attempts  
host all all 0.0.0.0 0.0.0.0 md5
```



3. **Recargar postgresql para reiniciar el servicio:** /etc/init.d/postgresql reload
4. **Ejecutar:** psql -t template 1 -U postgres
5. **Asignarle una contraseña al postgres:** alter user postgres with password 'planifacad';
6. **Crear usuario de base de datos:** create user administrador with password 'kjea' createdb;
7. **Presionar \q para salir**
8. **Configurar archivo :** /etc/postgresql/pg_hba.conf para pedir una contraseña al usuario postgres : vi /etc/postgres/pg_hba.conf

```
# PostgreSQL Client Authentication Configuration File
# =====
.....
#
# TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD
# Database administrative login by UNIX sockets
# (Cambiar todos los md5 por trust )

#local all postgres ident sameus
local all postgres trust
#
# All other connections by UNIX sockets
local all all trust
#
# All IPv4 connections from localhost
host all all 127.0.0.1 255.255.255.255 trust
#
# All IPv6 localhost connections
host all all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff trust
host all all ::ffff:127.0.0.1/128 trust
#
# reject all other connection attempts
host all all 0.0.0.0 0.0.0.0 trust
```

9. **Recargar postgresql para reiniciar el servicio:** /etc/init.d/postgresql reload

10. **Acceder al potgresql:** psql -t template 1 -U postgres

Se pedirá la contraseña del usuario postgres y si la contraseña es correcta aparece el siguiente prompt : template1=#



6. DISEÑO METODOLÓGICO

La ingeniería del software se centra en el hecho, establecimiento y uso de los principios robustos de la ingeniería. Orientados a obtener software económicos y fiables para que funcionen de manera eficiente en máquinas reales. Este proyecto consta de herramientas de alta calidad permiten mayor rapidez en el procesamiento de los datos.

La ingeniería del software está formada por 3 elementos claves:

- Métodos.
- Herramientas.
- Procedimientos.

6.1 Materiales

- a. **Hardware:** En nuestro proyecto a nivel de hardware emplearemos las siguientes herramientas:
 - 2 PC con las siguientes características:
 - ✓ Memoria RAM de 128 MB
 - ✓ 40 GB en Disco Duro
 - ✓ Procesador Intel® Celeron™ CPU
 - ✓ Velocidad de procesamiento 1200Mhz
- b. **Software:** Las herramientas software que emplearemos en nuestra aplicación son:
 - ✓ Microsoft Word.
 - ✓ Easy Case Proffesional Versión 4.21.016
 - ✓ Linux Debian (Sistema Operativo)
 - ✓ PostgreSQL (Sistema Gestor de Base de Datos)
 - ✓ Servidor Apache TomCat
 - ✓ Eclipse
 - ✓ Quanta Plus (Editor)
 - ✓ PHP.Net (Lenguaje de programación)



6.2 Métodos

Indican cómo construir técnicamente el software con la ayuda de diferentes tipos de modelos (estructurado, cascada, espiral, etc.) los cuales proporcionan una serie de pasos que se deben seguir durante el proceso de desarrollo.

Los modelos del ciclo de vida realizan actividades comunes como:

- Gestión del proyecto por medio de la descomposición del mismo en etapas.
- Uso de alguna metodología de trabajo en cada etapa.
- Verificación y validación de cada fase de desarrollo.

Descripción del ciclo de vida

El ciclo de vida exige un enfoque sistemático y secuencial acerca del desarrollo del software que indica con el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

Los modelos de desarrollo abordan el estudio de las actividades del sistema en las siguientes etapas:

- Planificación.
- Análisis.
- Diseño del sistema.
- Construcción y elaboración del sistema.

El modelo de desarrollo de sistemas que será implementado en nuestro sistema es el **MODELO EN CASCADA**.



Modelo en Cascada

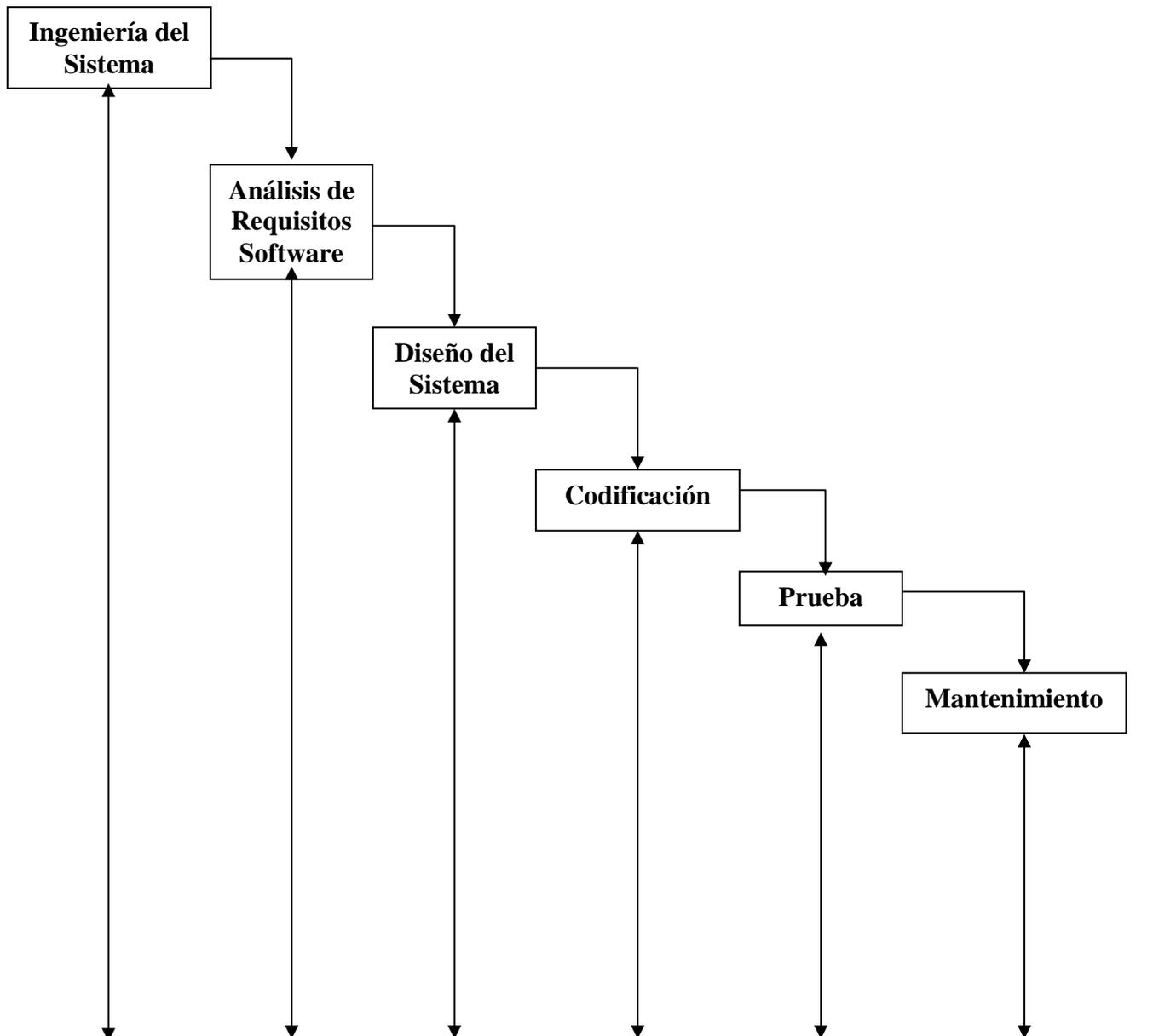
Descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

Actividades del Ciclo de Vida en Cascada

- **Ingeniería del sistema:** estudio de sistemas para obtener una convicción del análisis y diseño. Abarca el cómo trabaja un sistema existente y preparar modificaciones al sistema para cambiar el comportamiento del mismo.
- **Análisis:** Análisis del ERS, especificación de tareas.
- **Diseño:** Definición de las estructuras de datos, arquitectura e interfaces.
- **Codificación:** Conversión del diseño a un lenguaje de programación.
- **Prueba:** El software debe ser probado para descubrir los defectos que puedan existir en la función, en la lógica y en la implementación.
- **Mantenimiento:** La fase de mantenimiento se centra en el cambio. Esta fase aplica los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existe. Durante la fase de mantenimiento se centran tres tipos de cambios: corrección, adaptación y mejora.



Gráficamente el modelo de ciclo de vida en cascada se representa de la siguiente manera:





7. ANÁLISIS

ESPECIFICACIÓN DE REQUISITOS SOFTWARE (ERS)

1. Introducción:

1.1 Propósito

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación “MÓDULO PLANIFICACIÓN ACADÉMICA DE LA FACULTAD DE CIENCIAS UNAN-LEÓN” IMPLEMENTADO EN SISTEMAS DE CRÉDITOS ACADÉMICOS A TRAVÉS DE SOFTWARE LIBRE” que consiste en automatizar los procesos de gestión y planificación del área académica de dicha facultad.

Este documento se dirige a la Secretaría Académica de la Facultad de Ciencias y a los usuarios finales que deberán estudiarlo para comprender su funcionamiento y hacer un uso correcto y eficiente de la aplicación.

1.2 Alcance

El nombre con el que se conocerá a esta aplicación será: “**PLANIFICACIÓN ACADÉMICA DE LA FACULTAD DE CIENCIAS**”.

El producto realizará las siguientes funciones:

- ✓ Registrar datos correspondientes a cada una de los departamentos de la Facultad de Ciencias.
- ✓ Modificar datos de un departamento de la Facultad de Ciencias.
- ✓ Registrar datos de los usuarios del sistema.
- ✓ Modificar datos de los usuarios del sistema.
- ✓ Registrar datos correspondientes a cada una de las carreras ofertadas en la Facultad de Ciencias.
- ✓ Modificar datos de una carrera de la Facultad de Ciencias.



- ✓ Registrar datos correspondientes a los p^éns^um acad^émicos de cada una de las carreras de la Facultad de Ciencias.
- ✓ Modificar datos del p^éns^um acad^émico de una carrera en espec^ífico de la Facultad de Ciencias.
- ✓ Registrar datos correspondientes a cada una de las asignaturas impartidas en la Facultad de Ciencias.
- ✓ Modificar datos de una asignatura en espec^ífico.
- ✓ Registrar datos correspondientes a cada una de las aulas de clase y/o laboratorios de la Facultad de Ciencias.
- ✓ Modificar datos de un aula de clase y/o laboratorio de la Facultad de Ciencias.
- ✓ Registrar datos correspondientes a los docentes que laboran en la Facultad de Ciencias.
- ✓ Modificar datos de un docente de la Facultad de Ciencias.
- ✓ Registrar datos correspondientes a la divisi^ón de los grupos de clase por cada una de las asignaturas ofertadas en los departamentos de la Facultad de Ciencias.
- ✓ Modificar datos correspondientes a la divisi^ón de los grupos de clase por cada una de las asignaturas ofertadas en los departamentos de la Facultad de Ciencias.
- ✓ Registrar datos correspondientes de la asignaci^ón de aulas de clases y/o laboratorios a cada departamento de la Facultad de Ciencias.
- ✓ Modificar datos de la asignaci^ón de un aula de clase y/o laboratorio de la Facultad.
- ✓ Registrar datos correspondientes al horario de ocupaci^ón de cada una de las aulas de clases asignadas a cada departamento de la Facultad de Ciencias.
- ✓ Generaci^ón de los siguientes informes:
 - Reporte con los datos de un departamento en espec^ífico.
 - Reporte con los datos de todos los departamentos de la Facultad de Ciencias.
 - Reporte con los datos de una carrera en espec^ífico.



- Reporte con los datos de todas las carreras ofertadas en la Facultad de Ciencias.
- Pénsum académicos de cada una de las carreras ofertadas en la Facultad de Ciencias.
- Reporte con los datos de un docente en específico.
- Reporte con los datos de los docentes que imparten clases para un departamento específico de la Facultad de Ciencias.
- Reporte con los datos de todos los docentes de la Facultad de Ciencias.
- Reporte con los datos de un aula de clase y/o laboratorio en específico de la Facultad de Ciencias.
- Reporte con los datos de todas las aulas de clase y/o laboratorio asignadas a un departamento específico de la Facultad de Ciencias.
- Reporte con los datos de todas las aulas de clase y/o laboratorios de la Facultad de Ciencias.
- Reporte con los datos de la asignación de aulas de clase y/o laboratorios a cada departamento de la Facultad de Ciencias.
- Asignación de un aula de clase y/o laboratorio en específico.
- Horario de ocupación de un aula de clase y/o laboratorio de la Facultad de Ciencias.
- Reporte con los datos de una asignatura impartida en la Facultad de Ciencias.
- Reporte con los datos de todas las asignaturas impartidas en la Facultad de Ciencias.
- Reporte con los datos de las asignaturas impartidas por cada uno de los departamentos de la Facultad de Ciencias.
- Reporte con los datos de un determinado grupo de clase de la Facultad de Ciencias.
- Reporte con los datos de todos los grupos de clase de la Facultad de Ciencias.
- Reporte con los datos de los grupos de clase por cada una de las asignaturas que se imparten en la Facultad de Ciencias.



- Reporte con los datos de la carga académica de cada docente que pertenece a la Facultad de Ciencias.

1.3 Definiciones, acrónimos y abreviaturas:

- 1. Departamento:** Entidad que contendrá información descriptiva referente ha determinado departamento de los cuales está dividido administrativamente la Facultad de Ciencias. De aquí en adelante se conocerá como “Departamento”.
- 2. Carrera:** Entidad que contendrá información descriptiva referente ha determinada carrera de las que se ofertan en la Facultad de Ciencias. De aquí en adelante se conocerá como “Carrera”.
- 3. Docente:** Entidad que contendrá información descriptiva referente ha cada docente que labora para la Facultad de Ciencias. De aquí en adelante se conocerá como “Docente”.
- 4. Asignatura:** Entidad que contendrá información descriptiva referente ha cada una de las asignaturas que se imparten en la Facultad de Ciencias. De aquí en adelante se conocerá como “Asignatura”.
- 5. Pénsum Académico:** Registrará la información de los pénsum académicos de cada una de las carreras que se ofertan en la Facultad de Ciencias. De aquí en adelante se conocerá como “PensumAcademico”.
- 6. Aula:** Entidad que contendrá información descriptiva referente ha determinada aula de clase o laboratorio de la Facultad de Ciencias. De aquí en adelante se conocerá como “Aula”.
- 7. Usuarios:** Entidad que contendrá información descriptiva referente ha cada uno de los usuarios del sistema. De aquí en adelante se conocerá como “Usuarios”.
- 8. Aulas Asignadas:** Registrará la información de la asignación de aulas de clase y/o laboratorios a cada uno de los departamentos de la Facultad de Ciencias. De aquí en adelante se conocerá como “AulasAsignadas”.



9. **GrupoClase:** Registrará la información de la división de los grupos de clase por asignatura de la Facultad de Ciencias. Cada registro de esta entidad representará un renglón en la sección de detalles de “Planificacion”. De aquí en adelante se conocerá como “GrupoClase”.
10. **Planificacion:** Esta entidad llevará el registro de la planificación de cada aula y grupo de clase de la Facultad de Ciencias. De aquí en adelante se conocerá como “Planificacion”.
11. **Registrar datos de los departamentos de la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Departamento”. De aquí en adelante “NUEVO DEPARTAMENTO”.
12. **Modificar datos de los departamentos de la Facultad de Ciencias:** Acción de modificar los datos de un determinado producto registrado en la entidad “Departamento”. De aquí en adelante “EDITAR REGISTRO DEPARTAMENTO”.
13. **Borrar un Departamento de la entidad Departamento:** Acción de borrar un registro en la entidad “Departamento”. De aquí en adelante “BORRAR REGISTRO DEPARTAMENTO”.
14. **Buscar un registro Departamento:** Acción de buscar un registro en la entidad “Departamento”. De aquí en adelante “BUSCAR REGISTRO DEPARTAMENTO”.
15. **Registrar datos de las carreras que se ofertan en la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Carrera”. De aquí en adelante “NUEVA CARRERA”.
16. **Modificar datos de las carreras que se ofertan en la Facultad de Ciencias:** Acción de modificar los datos de una determinada carrera registrada en la entidad “Carrera”. De aquí en adelante “EDITAR REGISTRO CARRERA”.
17. **Borrar una carrera de la entidad Carrera:** Acción de borrar un registro en la entidad “Carrera”. De aquí en adelante “BORRAR REGISTRO CARRERA”.
18. **Buscar un registro de la entidad Carrera:** Acción de buscar un registro en la entidad “Carrera”. De aquí en adelante “BUSCAR REGISTRO CARRERA”.



- 19. Registrar datos de las asignaturas que se imparten en la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Asignatura”. De aquí en adelante “NUEVA ASIGNATURA”.
- 20. Modificar datos de las asignaturas que se imparten en la Facultad de Ciencias:** Acción de modificar los datos de un determinado docente registrado en la entidad “Asignatura”. De aquí en adelante “EDITAR REGISTRO ASIGNATURA”.
- 21. Borrar un registro asignatura de la entidad Asignatura:** Acción de borrar un registro en la entidad “Asignatura”. De aquí en adelante “BORRAR REGISTRO ASIGNATURA”.
- 22. Buscar un registro Asignatura:** Acción de buscar un registro en la entidad “Asignatura”. De aquí en adelante “BUSCAR REGISTRO ASIGNATURA”.
- 23. Registrar Pénsum académico de una carrera determinada:** Acción de añadir un registro en la entidad “PensumAcademico”. De aquí en adelante “NUEVO PENSUM ACADEMICO”.
- 24. Modificar datos del pénsum académico de una determinada carrera de las que se ofertan en la Facultad Ciencias:** Acción de modificar los datos del pénsum académico de una determinada carrera de las que se ofertan en la Facultad de Ciencias registrado en la entidad “PensumAcademico”. De aquí en adelante “EDITAR REGISTRO PENSUM ACADEMICO”.
- 25. Borrar un registro PensumAcademico de la entidad PensumAcademico:** Acción de borrar un registro en la entidad “PensumAcademico”. De aquí en adelante “BORRAR REGISTRO PENSUM ACADEMICO”.
- 26. Buscar un registro PensumAcademico:** Acción de buscar y mostrar un registro de la entidad “PensumAcademico”. De aquí en adelante “BUSCAR REGISTRO PENSUM ACADEMICO”.
- 27. Registrar datos de las aulas de clase | laboratorios de la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Aula”. De aquí en adelante “NUEVA AULA”.



- 28. Modificar datos de las aulas de clase | laboratorios de la Facultad de Ciencias:** Acción de modificar los datos de un determinado registro de la entidad “Aula”. De aquí en adelante “EDITAR REGISTRO AULA”.
- 29. Borrar un aula de clase | laboratorio de la entidad Aula:** Acción de borrar un registro de la entidad “Aula”. De aquí en adelante “BORRAR REGISTRO AULA”.
- 30. Buscar un registro de la entidad Aula:** Acción de buscar y mostrar un registro de la entidad “Aula”. De aquí en adelante “BUSCAR REGISTRO AULA”.
- 31. Registrar datos de la asignación de aulas de clase y/o laboratorios a determinado departamento de la Facultad de Ciencias:** Acción de añadir un registro en la entidad “AulasAsignadas”. De aquí en adelante “NUEVA ASIGNACION AULA”.
- 32. Borrar una registro asignación de la entidad AulasAsignadas:** Acción de borrar un registro de la entidad “AulasAsignadas”. De aquí en adelante “BORRAR REGISTRO ASIGNACION AULA”.
- 33. Buscar un registro de la entidad AulasAsignadas:** Acción de buscar y mostrar un registro de la entidad “AulasAsignadas”. De aquí en adelante “BUSCAR REGISTRO ASIGNACION AULA”.
- 34. Registrar datos de los docentes que laboran para la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Docente”. De aquí en adelante “NUEVO DOCENTE”.
- 35. Modificar datos de los docentes que laboran para la Facultad de Ciencias:** Acción de modificar los datos de un determinado docente registrado en la entidad “Docente”. De aquí en adelante “EDITAR REGISTRO DOCENTE”.
- 36. Borrar un registro de la entidad Docente:** Acción de borrar un registro en la entidad “Docente”. De aquí en adelante “BORRAR REGISTRO DOCENTE”.
- 37. Buscar un registro Docente:** Acción de buscar y mostrar un registro de la entidad “Docente”. De aquí en adelante “BUSCAR REGISTRO DOCENTE”.



- 38. Registrar datos de la división de los grupos de clase por asignatura de la de la Facultad de Ciencias:** Acción de añadir un registro en la entidad “GrupoClase”. De aquí en adelante “NUEVA DIVISIÓN GRUPO CLASE”.
- 39. Modificar datos de un grupo de clase de la Facultad de Ciencias:** Acción de modificar los datos de un determinado grupo de clase registrado en la entidad “GrupoClase”. De aquí en adelante “EDITAR REGISTRO GRUPO CLASE”.
- 40. Borrar un registro de la entidad GrupoClase:** Acción de borrar un registro en la entidad “GrupoClase”. De aquí en adelante “BORRAR REGISTRO GRUPO CLASE”.
- 41. Buscar un registro GrupoClase:** Acción de buscar y mostrar un registro de la entidad “GrupoClase”. De aquí en adelante “BUSCAR REGISTRO GRUPO CLASE”.
- 42. Registrar datos del horario de ocupación de una determinada aula de la Facultad de Ciencias:** Acción de añadir un registro en la entidad “Planificacion”. De aquí en adelante “NUEVA PLANIFICACION”.
- 43. Borrar un registro de la entidad Planificacion:** Acción de borrar un registro en la entidad “Planificacion”. De aquí en adelante “BORRAR REGISTRO PLANIFICACION”.
- 44. Buscar un registro Planificacion:** Acción de buscar y mostrar un registro de la entidad “Planificacion”. De aquí en adelante “BUSCAR REGISTRO PLANIFICACION”.
- 45. Permitir ingresar al sistema.** Acción de permitir el ingreso al sistema a través de la validación del nombre de usuario y contraseña. De aquí en adelante “INGRESAR SISTEMA”
- 46. Registrar datos de los usuarios del sistema.** Acción de añadir un registro en la entidad “Planificacion”. De aquí en adelante “NUEVO USUARIO”.
- 47. Modificar datos de un usuario del sistema:** Acción de modificar los datos de un determinado usuario del sistema registrado en la entidad “Usuarios”. De aquí en adelante “EDITAR REGISTRO USUARIO”.
- 48. Borrar un registro de la entidad Usuario:** Acción de borrar un registro en la entidad “Usuarios”. De aquí en adelante “BORRAR REGISTRO USUARIO”.



- 49. Reporte con los datos de un departamento en específico de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “Departamento”. De aquí en adelante “REPORTE DEPARTAMENTO”.
- 50. Generar reporte con los datos de todos los departamentos de la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “Departamento”. De aquí en adelante “REPORTE DEPARTAMENTOS FACULTAD”.
- 51. Reporte con los datos de una carrera en específico de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “Carrera”. De aquí en adelante “REPORTE CARRERA”.
- 52. Generar reporte con los datos de todas las carreras de la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “Carrera”. De aquí en adelante “REPORTE CARRERAS FACULTAD”.
- 53. Generar reporte del Pénsum Académico correspondiente a determinada carrera de las que se ofertan en la Facultad de Ciencias.** Acción de presentar un registro de la entidad “PensumAcademico”. De aquí en adelante “REPORTE PENSUM ACADEMICO”.
- 54. Reporte con los datos de un docente en específico de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “Docente”. De aquí en adelante “REPORTE DOCENTE”.
- 55. Generar reporte con los datos de todos los docentes que pertenecen a determinado departamento de la Facultad de Ciencias.** Acción de presentar determinados registros de la entidad “Docente”. De aquí en adelante “REPORTE DOCENTES DEPARTAMENTO”.
- 56. Generar reporte con los datos de todos los docentes que laboran para la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “Docente”. De aquí en adelante “REPORTE DOCENTES FACULTAD”.
- 57. Reporte con los datos de un aula en específico de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “Aula”. De aquí en adelante “REPORTE AULA”.



- 58. Generar reporte con los datos de todas las aulas que pertenecen a determinado departamento de la Facultad de Ciencias.** Acción de presentar determinados registros de la entidad “Aula”. De aquí en adelante “REPORTE AULAS DEPARTAMENTO”.
- 59. Generar reporte con los datos de todas las aulas que pertenecen a la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “Aula”. De aquí en adelante “REPORTE AULAS FACULTAD”.
- 60. Generar reporte con los datos de todas las asignaturas que imparte determinado departamento de la Facultad de Ciencias.** Acción de presentar determinados registros de la entidad “Asignatura”. De aquí en adelante “REPORTE ASIGNATURAS DEPARTAMENTO”.
- 61. Generar reporte con los datos de todas las asignaturas que se imparten en la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “Asignatura”. De aquí en adelante “REPORTE ASIGNATURAS FACULTAD”.
- 62. Generar reporte con los datos de la asignación de aulas de clase y/o laboratorios a cada departamento de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “AulasAsignadas”. De aquí en adelante “REPORTE ASIGNACION AULA”.
- 63. Reporte con los datos de un grupo de clase de la Facultad de Ciencias.** Acción de presentar un registro de la entidad “GrupoClase”. De aquí en adelante “REPORTE GRUPO CLASE”.
- 64. Generar reporte con los datos de todos los grupos de clase por cada una de las asignaturas que se imparten en la Facultad de Ciencias.** Acción de presentar determinados registros de la entidad “GrupoClase”. De aquí en adelante “REPORTE GRUPOS CLASE POR ASIGNATURA”.
- 65. Generar reporte con los datos de todos los grupos de clase de la Facultad de Ciencias.** Acción de presentar todos los registros de la entidad “GrupoClase”. De aquí en adelante “REPORTE GRUPOS DE CLASE FACULTAD”.



66. Generar reporte con los datos de la carga académica de cada docente que labora en la Facultad de Ciencias. De aquí en adelante “REPORTE CARGA ACADEMICA DOCENTE”.

1.4 Referencias:

- Folleto Especificación de Requisitos Software. Suministrado por Msc. Danilo Padilla.
- Introducción de un sistema de créditos académicos para las carreras de pregrado en la UNAN-León. Elaborado por Nidia Roa Gamboa.
- Direcciones de Internet:
 - ✓ www.milonic.com
 - ✓ www.nvu.com
 - ✓ www.postgresql.org
 - ✓ www.apache.org
 - ✓ www.php.net

1.5 Visión General:

Se realizará una descripción general del producto a desarrollar para pasar posteriormente a estudiar cada uno de los Requisitos Específicos.

2. DESCRIPCIÓN GENERAL:

2.1 Relaciones del producto:

La aplicación se desarrollará en un equipo con las siguientes características:

- ❖ 40GB en Disco Duro.
- ❖ 128 MB RAM.
- ❖ Intel(R) Celeron(TM) CPU 1200MHz.
- ❖ Sistema Operativo: Linux Debian.



El equipo en que se implantará el sistema final es:

- ❖ 40GB en Disco Duro.
- ❖ 128 MB RAM.
- ❖ Intel(R) Celeron(TM) CPU 1200MHz.
- ❖ Sistema Operativo: Linux Debian.

2.2 Funciones del producto:

El producto software debe realizar las siguientes funciones:

- Registrar datos de los usuarios del sistema.
- Registrar datos de cada uno de los departamentos de la Facultad.
- Registrar datos de cada una de las carreras de la Facultad, para ello el usuario debe llenar el formulario con la información requerida.
- Registrar datos del p nsun acad mico de cada una de las carreras ofertadas en la Facultad de Ciencias, para ello el usuario debe llenar el formulario con la informaci n requerida.
- Registrar datos de cada una de las asignaturas impartidas en la Facultad, para ello el usuario debe llenar el formulario con la informaci n requerida.
- Registrar los datos correspondientes a cada una de las aulas de clases y/o laboratorios de la Facultad, para ello el usuario debe llenar el formulario con la informaci n requerida.
- Registrar los datos correspondientes a cada uno de los docentes que laboran para la Facultad, para ello el usuario debe llenar el formulario con la informaci n requerida.
- Registrar datos de la asignaci n de aulas de clases y/o laboratorios a cada departamento de la Facultad de Ciencias.
- Registrar el horario de ocupaci n de cada una de las aulas de clases y laboratorios asignadas a cada departamento de la Facultad de Ciencias.
- Registrar datos de la divisi n de grupos de clase para cada una de las asignaturas impartidas en la Facultad de Ciencias.
- Generar reportes para cada una de las funciones del producto final.



2.3 Características del usuario:

El usuario final de la aplicación debe tener conocimientos informáticos acerca del manejo básico de un sistema computarizado.

2.4 Restricciones Generales:

La Secretaria Académica usuaria de nuestra aplicación deberá tener como requisitos mínimos:

- Computadores con: 20GB de Disco Duro, 128MB de Memoria RAM, Procesadores con velocidad de 1200Mhz.
- Sistema Operativo Linux Debian.
- Red de Área Local.

3. REQUISITOS ESPECÍFICOS:

3.1 Requisitos Funcionales:

3.1.1 NUEVO DEPARTAMENTO

3.1.1.1 Especificación:

3.1.1.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema los datos correspondientes a un determinado departamento de la Facultad de Ciencias, haciendo uso de la entidad Departamento.



3.1.1.1.2 Entradas:

Por pantalla:

- ⇒ nombredartamento: Este campo representa el nombre de cada uno de los departamentos de la Facultad de Ciencias.
- ⇒ nombredirector: Este campo representa el nombre del director del departamento.
- ⇒ anyofundacion: Este campo representa el año de la fundación del departamento.
- ⇒ extension: Representa el número de extensión telefónica que se le asigna a cada departamento dentro de la red telefónica de la Facultad de Ciencias.
- ⇒ observacion: Representa cualquier otra información de relevancia que se quiera registrar para documentar a cada departamento.

3.1.1.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrá ingresar los campos: nombredartamento, nombredirector, noextension, observacion. El campo anyofundacion podrá ser seleccionado de una lista.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón Guardar.

3.1.1.1.4 Salidas

Se añadirá un nuevo registro a la entidad Departamento.

3.1.2 EDITAR REGISTRO DEPARTAMENTO

3.1.2.1 Especificación:

3.1.2.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad Departamento.



3.1.2.1.2 Entradas

Por el sistema:

⇒ nombredepartamento: Este campo representa el nombre del departamento al cual se le desean modificar datos.

3.1.2.1.3 Proceso

Se mostrará al usuario una interfaz donde deberá seleccionar de una lista el nombre del departamento que desee editar para que los datos asociados a este sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que estos sean permanentes en la entidad Departamento pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.2.1.4 Salidas

Se actualizará el registro modificado en la entidad Departamento.

3.1.3 BORRAR REGISTRO DEPARTAMENTO

3.1.3.1 Especificación:

3.1.3.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Departamento.

3.1.3.2 Entradas

Por el sistema:

⇒ nombredepartamento: Este campo representa el nombre del departamento que desea eliminar de forma permanente de la Base de Datos.



3.1.3.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del departamento que desee eliminar para que los datos asociados a este sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.3.4 Salidas

Se eliminará un registro de la entidad Departamento.

3.1.4 BUSCAR REGISTRO DEPARTAMENTO

3.1.4.1 Especificación:

3.1.4.1.1 Introducción:

Esta función permitirá al usuario mostrar sobre un formulario determinado registro existente en la entidad Departamento especificando para ello el Nombre del Departamento.

3.1.4.1.2 Entradas

Por el sistema:

⇒ nombredeldepartamento: Este campo representa el nombre del departamento que desea visualizar.

3.1.4.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el nombre del departamento que desea buscar.



Los datos asociados al Departamento serán visualizados en un nuevo formulario luego de haber pulsado el botón Buscar donde el usuario podrá consultar cualquier información que necesite de este.

3.1.4.1.4 Salidas

Se visualizará en un formulario la información relacionada con el Departamento especificado en el campo de búsqueda “Departamento”.

3.1.5 NUEVA CARRERA

3.1.5.1 Especificación:

3.1.5.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema los datos correspondientes a determinada carrera de la Facultad de Ciencias, haciendo uso de la entidad Carrera y Departamento.

3.1.5.1.2 Entradas

Por pantalla:

- ⇒ nombrecarrera: Este campo representa el nombre de cada una de las carreras que se ofertan en la Facultad de Ciencias.
- ⇒ anyofundacion: Este campo representa el año de la fundación de la carrera.
- ⇒ cantcreditoblig: Representa el número de créditos obligatorios que el estudiante deberá cursar para cumplir con su respectivo plan de estudios.
- ⇒ cantcreditopt: Representa el número de créditos optativos que el estudiante deberá cursar para cumplir con su respectivo plan de estudios.
- ⇒ cantcreditoselectivos: Representa el número de créditos electivos que el estudiante deberá cursar para cumplir con su respectivo plan de estudios.
- ⇒ observacion: Representa cualquier otra información de relevancia que se quiera registrar para documentar a la carrera.



Por el sistema:

⇒ nombredeldepartamento: Representa el nombre del departamento al cual pertenece la carrera que se desea ingresar.

3.1.5.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrá ingresar los campos: nombrecarrera, anyofundacion, cantcreditobligatorios, cantcreditoptativos, cantcreditoselectivos, totalcreditoscarrera, observacion.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón Guardar.

3.1.5.1.4 Salidas

Se añadirá un nuevo registro a la entidad Carrera.

3.1.6 EDITAR REGISTRO CARRERA

3.1.6.1 Especificación:

3.1.6.1.1 Introducción:

Esta función permitirá al usuario editar un registro existente en la entidad Carrera.

3.1.6.1.2 Entradas

Por el sistema:

⇒ nombrecarrera: Este campo representa el nombre de la carrera que desea editar.

3.1.6.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre de la carrera que desee editar para que los datos asociados a esta sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior



podremos realizar los cambios necesarios y pulsar el botón Guardar para que estos sean permanentes en la entidad Carrera o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.6.1.4 Salidas

Se actualizará el registro modificado en la entidad Carrera.

3.1.7 BORRAR REGISTRO CARRERA

3.1.7.1 Especificación:

3.1.7.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Carrera.

3.1.7.2 Entradas

Por el sistema:

⇒ nombrecarrera: Este campo representa el nombre de la carrera que desea eliminar de forma permanente de la Base de Datos.

3.1.7.3 Proceso

Se mostrará al usuario un formulario donde deberá especificar el nombre de la carrera que desee eliminar para que los datos asociados a ésta sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.7.4 Salidas

Se eliminará un registro de la entidad Carrera.



3.1.8 BUSCAR REGISTRO CARRERA

3.1.8.1 Especificación:

3.1.8.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad Carrera especificando para ello el nombre de la carrera.

3.1.8.1.2 Entradas

Por el sistema:

⇒ nombrecarrera: Representa el nombre de la carrera de la cual se desea visualizar la información asociada.

3.1.8.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el nombre de la carrera a buscar. Los datos asociados a la carrera serán visualizados en un nuevo formulario donde el usuario podrá consultar la información que desee siempre que haya pulsado el botón Buscar.

3.1.8.1.4 Salidas

Se visualizará en un formulario la información relacionada con la carrera especificada en el campo de búsqueda “Carrera”.

3.1.9 NUEVA ASIGNATURA

3.1.9.1 Especificación:

3.1.9.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema los datos correspondientes a cada una de las asignaturas impartidas en la Facultad de Ciencias, haciendo uso de la entidad Asignatura, Departamento.



3.1.9.1.2 Entradas

Por pantalla:

- ⇒ nombreasignatura: Este campo representa el nombre de la asignatura que forma parte del p nsun acad mico de cada una de las carreras que se ofertan en la Facultad de Ciencias.
- ⇒ no_creditos: Este campo representa el n mero de cr ditos que debe cursar para la asignatura.
- ⇒ observacion: Representa cualquier otra informaci n de relevancia que se quiera registrar para documentar a cada asignatura.

Por el sistema:

- ⇒ nombredepartamento: Representa el nombre del departamento que oferta dicha asignatura.
- ⇒ codigoasignatura: Representa el c digo a trav s del cual se conocer  de forma  nica a cada asignatura. Este c digo ser  generado a partir de los campos: nombreasignatura y no_creditos, es decir se tomar n 4 caracteres de cada palabra del nombre de la asignatura (excluyendo: de, las, los etc.) + el valor que tenga el campo no_cr ditos. Por ejemplo: El c digo de la asignatura Arquitectura de Computadores I ser : ArquCompI20.

3.1.9.1.3 Proceso

Se mostrar  al usuario la interfaz a trav s de la cual podr  ingresar los campos nombreasignatura, observaci n. Los campos no_creditos, nombredepartamento, podr n ser seleccionados de una lista. Para guardar los datos introducidos en la base de datos se deber  pulsar el bot n Guardar.

3.1.9.1.4 Salidas

Se a adir  un nuevo registro a la entidad Asignatura.



3.1.10 EDITAR REGISTRO ASIGNATURA

3.1.10.1 Especificación:

3.1.10.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad Asignatura.

3.1.10.1.2 Entradas

Por el sistema:

⇒ nombreasignatura: Este campo representa el nombre de asignatura que se desea editar.

3.1.10.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre de la asignatura que se desea modificar para que los datos asociados a esta sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que estos sean permanentes en la entidad Asignatura pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.10.1.4 Salidas

Se actualizará el registro modificado en la entidad Asignatura.

3.1.11 BORRAR REGISTRO ASIGNATURA

3.1.11.1 Especificación:

3.1.11.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Asignatura.



3.1.11.2 Entradas

Por el sistema:

⇒ nombre asignatura: Este campo representa el nombre de la asignatura que se desea eliminar de forma permanente de la Base de datos.

3.1.11.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre de la asignatura que se desea eliminar para que los datos asociados a ésta sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación pulsaremos el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.11.4 Salidas

Se eliminará un registro de la entidad Asignatura.

3.1.12 BUSCAR REGISTRO ASIGNATURA

3.1.12.1 Especificación:

3.1.12.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad Asignatura especificando para ello el nombre de la asignatura.

3.1.12.1.2 Entradas

Por el sistema:

⇒ nombre asignatura: Representa el nombre de la asignatura a buscar.



3.1.12.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el nombre de la asignatura a buscar. Los datos asociados a la asignatura serán visualizados en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de esta siempre que haya pulsado el botón Buscar.

3.1.12.1.4 Salidas

Se visualizará en un formulario la información relacionada con la asignatura especificada en el campo de búsqueda “Nombre Asignatura”.

3.1.13 NUEVO PENSUM ACADEMICO

3.1.13.1 Especificación:

3.1.13.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema las asignaturas correspondientes al plan de estudios de una determinada carrera, haciendo uso de las entidades Carrera, Asignatura y PensumAcademico.

3.1.13.1.2 Entradas:

Por el sistema:

- ⇒ **codigoasignatura:** Representa el código de la asignatura que forma parte de dicho pènsun acadèmico de carrera.
- ⇒ **nombrecarrera:** Representa el nombre de la carrera a la cual se le estè asignando pènsun.
- ⇒ **tipo:** Este campo representa el tipo de asignatura que puede ser: electiva, optativa, obligatoria.
- ⇒ **prerequisito:** Representa el nombre de la asignatura que el estudiante deberè haber cursado antes de la asignatura indicada en el campo nombreasignatura.



3.1.13.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrán seleccionar de listas los campos: nombreasignatura, nombrecarrera, tipo, prerequisite.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón Guardar.

3.1.13.1.4 Salidas

Se añadirá un nuevo registro a la entidad PensumAcademico.

3.1.14 EDITAR REGISTRO PENSUM ACADEMICO

3.1.14.1 Especificación:

3.1.14.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad PénsumAcademico, indicando para ello el nombre de la carrera y la asignatura

3.1.14.1.2 Entradas

Por el Sistema:

- ⇒ nombrecarrera: Representa el nombre de la carrera a la cual desea modificar dicho pénsum académico.
- ⇒ nombreasignatura: Representa el nombre de la asignatura que desea editar del pénsum de dicha carrera.

3.1.14.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre de la carrera y la asignatura que se desea editar, para que los datos asociados a dicha asignatura sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que



estos sean permanentes en la entidad PensumAcademico pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.14.1.4 Salidas

Se actualizará el registro modificado en la entidad PensumAcademico.

3.1.15 BORRAR REGISTRO PENSUM ACADEMICO

3.1.15.1 Especificación:

3.1.15.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad PensumAcademico asociado a determinada carrera de las que se ofertan en la Facultad de Ciencias.

3.1.15.2 Entradas:

Por el sistema:

- ⇒ nombrecarrera: Representa la carrera a la cual desea borrar determinada asignatura asociada a su pénsum académico.
- ⇒ nombreasignatura: Representa el nombre de la asignatura que se desea eliminar del pénsum académico de determinada carrera.

3.1.15.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de listas el nombre de la carrera y la asignatura que desee borrar del pénsum académico de la carrera seleccionada, para que los datos asociados a éste sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.



3.1.15.4 Salidas

Se eliminará un registro de la entidad PensumAcademico.

3.1.16 BUSCAR REGISTRO PÉNSUM ACADÉMICO

3.1.16.1 Especificación:

3.1.16.1.1 Introducción:

Esta función permitirá al usuario visualizar determinado registro existente en la entidad PensumAcademico especificando para ello el nombre de la carrera de la cual desee mostrar información del Pénsum Académico asociado.

3.1.16.1.2 Entradas

Por el sistema:

- ⇒ nombrecarrera: Representa el nombre de la carrera de la cual desea visualizar el pénsum académico asociado.

3.1.16.1.3 Proceso:

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el nombre de la carrera de la cual desee buscar el pénsum académico. La información asociada al pénsum académico y a la carrera seleccionada será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier dato que necesite de este siempre que haya pulsado el botón Buscar.

3.1.16.1.4 Salidas

Se visualizará en un formulario la información relacionada con el pénsum académico especificado en el campo de búsqueda “Nombre Carrera”.



3.1.17 NUEVA AULA

3.1.17.1 Especificación:

3.1.17.1.1 Introducción:

Esta función permitirá al usuario almacenar en la base de datos información correspondiente a cada aula (Aula de clase/Laboratorio) perteneciente a la Facultad de Ciencias, haciendo uso de la entidad Aula.

3.1.17.1.2 Entradas

Por pantalla:

- ⇒ nombreaula: Este campo representa el nombre con el que se identificará a cada una de las aulas de clase y/o laboratorios de la Facultad de Ciencias.
- ⇒ ubicacion: Representa la ubicación exacta (específica) del aula de clase.
- ⇒ capacidad: Representa el número máximo de personas que pueden ser alojados en un aula de clase o laboratorio para recibir clases.
- ⇒ no Equipos: Representa la cantidad de equipos (computadoras, beaker, tubos de ensayo, erlemeyer, pipetas, etc.) en el caso de que el aula sea del tipo laboratorio.
- ⇒ observacion: Representa cualquier otra información de relevancia que se quiera registrar para el aula.

Por el sistema:

- ⇒ códigoaula: Representa el código a través del cual se conocerá de forma única a cada aula de la Facultad de Ciencias. Este código será generado a partir del campo: nombreaula, es decir se tomará el primer carácter de cada palabra del nombre del aula (excluyendo: de, las, los etc.). Por ejemplo: El código del aula Mariano Fiallos Gil 1 será MFG1.
- ⇒ tipo_aula: Este campo representa el tipo de aula. En nuestro sistema consideraremos a los laboratorios como un tipo de aula.



3.1.17.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrá ingresar los datos: nombreaula, capacidad, no_equipos, observación, ubicacion. El campo tipoaula podrá ser seleccionado de una lista. Para guardar los datos introducidos en la base de datos se deberá pulsar el botón Guardar.

3.1.17.1.4 Salidas

Se añadirá un nuevo registro a la entidad Aula.

3.1.18 EDITAR REGISTRO AULA

3.1.18.1 Especificación:

3.1.18.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad Aula. El campo codigoaula no podrá ser modificado por razones de reglas de integridad referencial.

3.1.18.1.2 Entradas

Por el sistema:

⇒ nombreaula: Este campo representa el nombre del aula a la cual se desea modificar información.

3.1.18.1.3 Proceso:

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del aula de clase o laboratorio que desea editar para que los datos asociados a esta sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que estos sean



permanentes en la entidad Aula pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.18.1.4 Salidas:

Se actualizará el registro modificado en la entidad Aula.

3.1.19 BORRAR REGISTRO AULA

3.1.19.1 Especificación:

3.1.19.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Aula.

3.1.19.2 Entradas

Por el sistema:

⇒ nombreaula: Representa el nombre del aula que se desea borrar.

3.1.19.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del aula de clase o laboratorio que desea eliminar para que los datos asociados a esta sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.19.4 Salidas

Se eliminará un registro de la entidad Aula.



3.1.20 BUSCAR REGISTRO AULA

3.1.20.1 Especificación:

3.1.20.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad Aula especificando para ello el nombre del aula.

3.1.20.1.2 Entradas

Por el sistema:

⇒ nombreaula: Representa el nombre del aula a buscar.

3.1.20.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del aula de clase o laboratorio a buscar. La información asociada al aula de clase o laboratorio será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de esta siempre que haya pulsado el botón Buscar.

3.1.20.1.4 Salidas

Se visualizará en un formulario la información relacionada con el aula especificada en el campo de búsqueda “Nombre Aula”.

3.1.21 NUEVA ASIGNACION AULA

3.1.21.1 Especificación:

3.1.21.1.1 Introducción:

Esta función permitirá al usuario almacenar en la base de datos información correspondiente a la asignación de aulas de clase y/o laboratorios a cada uno de los departamentos de la Facultad de Ciencias, haciendo uso de la entidad AulasAsignadas, Aula y Departamento.



3.1.21.1.2 Entradas

Por el sistema:

- ⇒ nombreaula: Representa el nombre del aula que se desea asignar a determinado departamento de la Facultad de Ciencias.
- ⇒ nombredepartamento: Representa el nombre del departamento al cual se le asignará un aula (aula de clase y/o laboratorio).
- ⇒ hora_inicio: Representa la hora de inicio de la asignación del aula en cuestión.
- ⇒ hora_fin: Representa la hora de fin de la asignación del aula en cuestión.

3.1.21.1.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de listas el nombre del departamento y el nombre de aula que desea asignar ha dicho departamento, así como también la hora de inicio y fin de la asignación del aula en cuestión. Dicho registro será añadido a la entidad AulasAsignadas al pulsar el botón Guardar.

3.1.21.1.4 Salidas

Se añadirá un nuevo registro a la entidad AulasAsignadas.

3.1.22 BORRAR REGISTRO AULAS ASIGNADAS

3.1.22.1 Especificación:

3.1.22.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad AulasAsignadas.



3.1.22.2 Entradas

Por el sistema:

- ⇒ nombredepartamento: Este campo representa el nombre del departamento al cual se le desea eliminar determinada asignación de aula.
- ⇒ nombreaula: Este campo representa el nombre del aula.
- ⇒ hora_inicio: Este campo representa la hora de inicio de la asignación.
- ⇒ hora_fin: Este campo representa la hora de fin de la asignación.

3.1.22.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de listas: el nombre del aula de clase o laboratorio, el nombre del departamento, la hora de inicio y hora de fin que desea eliminar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.22.4 Salidas

Se eliminará un registro de la entidad AulasAsignadas.

3.1.23 BUSCAR REGISTRO AULAS ASIGNADAS

3.1.23.1 Especificación:

3.1.23.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad AulasAsignadas especificando para ello el nombre del aula.



3.1.23.1.2 Entradas

Por el sistema:

⇒ nombreaula: Representa el nombre del aula de la cual desea obtener información acerca de su horario de ocupación.

3.1.23.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del aula de clase o laboratorio a buscar. La información asociada al aula de clase o laboratorio será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de esta siempre que haya pulsado el botón Buscar.

3.1.23.1.4 Salidas

Se visualizará en un formulario la información relacionada con el aula especificada en el campo de búsqueda “Nombre Aula”.

3.1.24 NUEVO DOCENTE

3.1.24.1 Especificación:

3.1.24.1.1 Introducción:

Esta función permitirá al usuario almacenar en la base de datos información correspondiente a cada docente que labora para la Facultad de Ciencias, haciendo uso de la entidad Docente y Departamento.

3.1.24.2 Entradas

Por pantalla:

⇒ noinss: Este campo representa el número de inss de cada docente que lo identificará de manera única a cada docente que labora para la Facultad de Ciencias.



- ⇒ nombredocente: Este campo representa el nombre completo de cada docente.
- ⇒ telefono: Este campo representa el número telefónico de cada docente.
- ⇒ horas_semana: Este campo representa el número de horas de clase a impartir por semana.
- ⇒ correo: Este campo representa la dirección de correo electrónico de cada docente.

Por el sistema:

- ⇒ nombreddepartamento: Representa el nombre del departamento al cual está asignado dicho docente.
- ⇒ categoría: Representa la categoría (Máster, Licenciado, Phd) de cada docente.
- ⇒ tipo: Representa el tipo (Titular, Adjunto, Horario) de contratación de cada docente.

3.1.24.3 Proceso

Se mostrará al usuario un formulario a través de la cual podrá ingresar los datos correspondientes a cada docente que trabaja en la Facultad de Ciencias. Los campos: categoríadocente, tipodocente podrán ser seleccionados de una lista. Dicho registro será añadido a la entidad Docente al pulsar el botón Guardar.

3.1.24.4 Salidas

Se añadirá un nuevo registro a la entidad Docente.



3.1.25 EDITAR REGISTRO DOCENTE

3.1.25.1 Especificación:

3.1.25.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad Docente. El campo noinss no podrá ser modificado por razones de reglas de integridad referencial.

3.1.25.1.2 Entradas

Por el sistema:

⇒ nombredocente: Este campo representa el nombre del docente al cual desea modificar datos.

3.1.25.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del docente al cual desea modificar información, para que los datos asociados a este sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que estos sean permanentes en la entidad Docente pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.25.1.4 Salidas

Se actualizará el registro modificado en la entidad Docente.



3.1.26 BORRAR REGISTRO DOCENTE

3.1.26.1 Especificación:

3.1.26.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Docente.

3.1.26.2 Entradas

Por el sistema:

⇒ nombredocente: Este campo representa el nombre del docente que desea eliminar de forma permanente de la Base de Datos.

3.1.26.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del docente que desea eliminar para que los datos asociados a éste sean visualizados en un formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.26.4 Salidas

Se eliminará un registro de la entidad Docente.

3.1.27 BUSCAR REGISTRO DOCENTE

3.1.27.1 Especificación:

3.1.27.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad Docente especificando para ello el nombre del docente.



3.1.27.1.2 Entradas

Por pantalla:

⇒ nombredocente: Este campo representa el nombre del docente del cual se desea obtener información.

3.1.27.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre del docente que desea buscar. La información asociada al docente será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de este siempre que haya pulsado el botón Buscar.

3.1.27.1.4 Salidas

Se visualizará en un formulario la información relacionada con el docente especificado en el campo de búsqueda “Nombre Docente”.

3.1.28 NUEVA DIVISION GRUPO CLASE

3.1.28.1 Especificación:

3.1.28.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema los datos correspondientes a la división del o los grupos de clase por asignatura, haciendo uso de la entidad Asignatura, Docente, GrupoClase.

3.1.28.1.2 Entradas

Por el sistema:

⇒ nombregroupo: Este campo representa el nombre que tendrá cada grupo de clase.



- ⇒ noestudiantes: Representa el número de estudiantes que conforman cada grupo de clase.
- ⇒ anyolectivo: Este campo representa el año lectivo (calendario) para el cual estará activo dicho grupo de clase.
- ⇒ semestreacadémico: Este campo representa el valor del semestre académico para el cual estará activa la división del grupo de clase.
- ⇒ codigoasignatura: Representa el código a través del cual se conocerá de forma única a cada asignatura.
- ⇒ noinss: Representa una secuencia de caracteres que identificarán de manera única a cada docente que labora para la Facultad de Ciencias.
- ⇒ tipogrupo: Este campo representa el tipo de grupo de clase, el cual puede ser Teórico o de Laboratorio.

3.1.28.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrá seleccionar de listas los campos correspondientes a la división de grupos de clase por asignatura que son: tipogrupo, codigogrupo, nombregrupo, anyolectivo, semestreacadémico. Para guardar los datos introducidos en la Base de Datos se deberá pulsar el botón Guardar.

3.1.28.1.4 Salidas

Se añadirá un nuevo registro a la entidad GrupoClase.



3.1.29 EDITAR REGISTRO GRUPO CLASE

3.1.29.1 Especificación:

3.1.29.1.1 Introducción:

Esta función permitirá al usuario modificar un registro existente en la entidad GrupoClase. El campo código grupo no podrá ser modificado por razones de reglas de integridad referencial.

3.1.29.1.2 Entradas

Por el sistema:

⇒ código grupo: Este campo representa el código del grupo al cual se desea modificar información.

3.1.29.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el código del grupo al cual se desea modificar información, para que los datos asociados a dicho grupo sean mostrados sobre el formulario, luego de haber pulsado el botón Buscar. Una vez ejecutado lo anterior podremos realizar los cambios necesarios y hacer que estos sean permanentes en la entidad GrupoClase pulsando para ello el botón Guardar o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.29.1.4 Salidas

Se actualizará el registro modificado en la entidad GrupoClase.

3.1.30 BORRAR REGISTRO GRUPO CLASE

3.1.30.1 Especificación:

3.1.30.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad GrupoClase.



3.1.30.2 Entradas

Por el sistema:

⇒ **códigogrupos**: Este campo representa el código del grupo que se desea borrar de la Base de datos.

3.1.30.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el código del grupo que se desea borrar, para que los datos asociados a dicho grupo de clase sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación pulsaremos el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.30.4 Salidas

Se eliminará un registro de la entidad GrupoClase.

3.1.31 BUSCAR REGISTRO GRUPO CLASE

3.1.31.1 Especificación:

3.1.31.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad GrupoClase especificando para ello el código de grupo.

3.1.31.1.2 Entradas

Por el sistema:

⇒ **codigogrupos**: Este campo representa el código del grupo de clase del cual se desea visualizar información.



3.1.31.1.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el código del grupo de clase a buscar.

La información asociada al grupo de clase será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de este siempre que haya pulsado el botón Buscar.

3.1.31.1.4 Salidas

Se visualizará en un formulario la información relacionada con el grupo de clase especificado en el campo de búsqueda “Grupo”.

3.1.32 NUEVA PLANIFICACION

3.1.32.1 Especificación:

3.1.32.1.1 Introducción:

Esta función permitirá al usuario registrar en el sistema los datos correspondientes al horario de clase de determinado grupo de la Facultad de Ciencias, haciendo uso de la entidad Aula, GrupoClase y Planificación.

3.1.32.1.2 Entradas

Por el sistema:

- ⇒ **codigogrupo:** Representa el código del grupo al cual se le asignará horario de clase.
- ⇒ **codigoaula:** Representa el código del aula donde se le impartirá clases al grupo especificado en el campo **codigogrupo**.
- ⇒ **día:** Representa una secuencia de caracteres que identificarán los 5 días (Lunes...Viernes) de la semana laborales en la Facultad de Ciencias.



⇒ horainicio: Este campo representa la hora de inicio de las clases, las cuales actualmente tienen un período de 2 horas. Ejemplo: 7:00 am - 9:00 am, 9:00 am-11:00 am....

3.1.32.1.3 Proceso

Se mostrará al usuario la interfaz a través de la cual podrá seleccionar de listas los campos correspondientes al horario de clase para cada grupo de clase de la Facultad de Ciencias: codigogrupos, codigoaula, día, horainicio. Para guardar los datos introducidos en la base de datos se deberá pulsar el botón Guardar.

3.1.32.1.4 Salidas

Se añadirá un nuevo registro a la entidad Planificacion.

3.1.33 BORRAR REGISTRO PLANIFICACION

3.1.33.1 Especificación:

3.1.33.1.1 Introducción:

Esta función permitirá al usuario borrar un registro existente en la entidad Planificacion.

3.1.33.2 Entradas

Por el sistema:

- ⇒ codigogrupos: Representa el código del grupo al cual se le eliminará determinada asignación de horario de clase.
- ⇒ codigoaula: Representa el código del aula a la cual se le eliminará determinado hora de ocupación.
- ⇒ día: Representa el día de la asignación que se quiere eliminar.
- ⇒ horainicio: Representa la hora de asignación que se quiere eliminar.



3.1.33.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista: el código del grupo, código de aula, día y hora de inicio. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.33.4 Salidas

Se eliminará un registro de la entidad Planificacion.

3.1.34 BUSCAR REGISTRO PLANIFICACION

3.1.34.1 Especificación:

3.1.34.1.1 Introducción:

Esta función permitirá al usuario visualizar un registro existente en la entidad Planificacion especificando para ello el nombre del aula.

3.1.34.1.2 Entradas

Por el sistema:

⇒ nombreaula: Representa el nombre del aula

3.1.34.1.3 Proceso

Se mostrará al usuario un formulario (interfaz) donde deberá especificar: el nombre del aula de la cual desea el horario de ocupación semanal. La información asociada al aula seleccionada será visualizada en un nuevo formulario donde el usuario podrá consultar cualquier información que necesite de esta siempre que haya pulsado el botón Buscar.



3.1.34.1.4 Salidas

Se visualizará en un formulario la información relacionada con el horario de ocupación del aula especificada en el campo de búsqueda “Nombre Aula”.

3.1.35 INGRESAR SISTEMA

3.1.35.1 Especificación:

3.1.35.1.1 Introducción:

Esta función permitirá a los usuarios registrados ingresar al sistema siempre y cuando el escriban su ID de usuario y contraseña válidos.

3.1.35.2 Entradas

Por el sistema:

- ⇒ nombreusuario: Este campo representa el ID de cada usuario que sea válido en el sistema.
- ⇒ password: Representa la contraseña asociada al usuario que está intentando ingresar al sistema.

3.1.35.3 Proceso

Se mostrará al usuario la interfaz principal de acceso al sistema, donde deberá introducir el nombre de usuario y su contraseña, en la cual el sistema verificará si los datos introducidos son válidos, de ser así el usuario podrá ingresar, de lo contrario se presentará un mensaje indicándole que el ID de usuario o contraseña son incorrectos.

3.1.35.4 Salidas

Ninguna



3.1.36 NUEVO USUARIO

3.1.36.1 Especificación:

3.1.36.1.1 Introducción:

Esta función permitirá al usuario con privilegios de administrador dar de alta en el sistema un nuevo ID de usuario y sus contraseña.

3.1.36.1.2 Entradas

Por pantalla:

- ⇒ nombreusuario: Representa el ID de usuario a registrar.
- ⇒ Password: Representa la contraseña asociada al usuario que se esta intentando dar de alta.

Por el sistema:

- ⇒ tipousuario: Representa el tipo de usuario al que se está intentando dar de alta., Puede ser: Administrador, Invitado, Director, cada uno de estos tiene diferentes privilegios sobre el sistema.

3.1.36.1.3 Proceso

Se mostrará al usuario el formulario de introducción del nombre de usuario y su contraseña.; del usuario que se está intentando dar de alta. El campo tipousuario podrá ser seleccionado de una lista. El sistema verificará que dicho ID de usuario no exista de lo contrario el proceso no podrá ser completado hasta que se proponga un nuevo ID de usuario que no esté registrado en el sistema.

3.1.36.1.4 Salidas

Se añadirá un nuevo registro a la entidad Usuarios.



3.1.37 EDITAR REGISTRO USUARIO

3.1.37.1 Especificación:

3.1.37.1.1 Introducción:

Esta función permitirá al usuario actual modificar su contraseña.

3.1.37.1.2 Entradas

Por pantalla:

⇒ password: Este campo representa la contraseña asociada al usuario que se está intentando modificar.

Por el sistema:

⇒ nombreusuario: Este campo representa el ID de usuario.

3.1.37.1.3 Proceso:

Se mostrará al usuario la interfaz de modificación de la contraseña del usuario que está haciendo uso del sistema.

3.1.37.1.4 Salidas:

Se actualizará el registro modificado en la entidad Usuarios.

3.1.38 BORRAR REGISTRO USUARIO

3.1.38.1 Especificación:

3.1.38.1.1 Introducción:

Esta función permitirá al usuario con privilegios de administrador dar de baja a determinado usuario.



3.1.38.2 Entradas

Por el sistema:

⇒ nombreusuario: Representa el nombre usuario que se desea eliminar.

3.1.38.3 Proceso

Se mostrará al usuario un formulario donde deberá seleccionar de una lista el nombre de usuario que desea eliminar para que los datos asociados a esta sean visualizados en el formulario luego de haber pulsado el botón Borrar. A continuación podremos pulsar el botón Borrar para eliminar de forma permanente el registro de la base de datos o bien pulsar el botón Cancelar para no ejecutar ningún cambio.

3.1.38.4 Salidas

Se eliminará un registro de la entidad Usuarios.

3.1.39 REPORTE DEPARTAMENTO

3.1.39.1 Especificación:

3.1.39.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a determinado departamento de la Facultad de Ciencias. Esta función estará soportada por la entidad Departamento.

3.1.39.2 Entradas

Por el sistema:

⇒ nombreddepartamento: Este campo representa el nombre del departamento del cual desea visualizar la información asociada.



3.1.39.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre del Departamento del cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener la información general asociada a dicho departamento.

3.1.39.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: NombreDepartamento, NombreDirector, AnyoFundacion, Extensión, Observación.

3.1.40 REPORTE DEPARTAMENTOS FACULTAD

3.1.40.1 Especificación:

3.1.40.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todos los departamentos de la Facultad de Ciencias. Esta función estará soportada por la entidad Departamento.

3.1.40.2 Entradas

Ninguna.

3.1.40.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Departamentos de la Facultad de Ciencias** para obtener la información general de todos los departamentos de la Facultad de Ciencias.



3.1.40.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: nombreddepartamento, nombredirector, anyofundacion, extensión, observación para cada departamento.

3.1.41 REPORTE CARRERA

3.1.41.1 Especificación:

3.1.41.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a determinada carrera de la Facultad de Ciencias. Esta función estará soportada por la entidad Carrera.

3.1.41.2 Entradas

Por el sistema:

⇒ nombrecarrera: Este campo representa el nombre de la carrera de la cual se desea obtener reporte.

3.1.41.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre de la carrera de la cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener el informe de dicha carrera.

3.1.41.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: nombrecarrera, cantcreditoptativos, cantcreditoblig, cantcreditoelectivos, anyofundacion, totalcreditoscarrera, observación, nombreddepartamento.



3.1.42 REPORTE CARRERAS FACULTAD

3.1.42.1 Especificación:

3.1.42.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todas las carreras de la Facultad de Ciencias. Esta función estará soportada por la entidad Carrera.

3.1.42.2 Entradas

Ninguna.

3.1.42.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Carreras de la Facultad de Ciencias** para obtener la información general de todas las carreras de la Facultad de Ciencias.

3.1.42.4 Salidas

Se mostrará por pantalla una página que contendrá los datos de la carrera: nombrecarrera, cantcreditoopt, cantcreditoblig, cantcreditelectivos, anyofundacion, totalcreditoscarrera, observación, nombredepartamento.

3.1.43 REPORTE PENSUM ACADEMICO

3.1.43.1 Especificación:

3.1.43.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada el Pénsum Académico correspondiente a determinada carrera de la Facultad de Ciencias. Esta función estará soportada por las entidades Carrera, Asignatura y PenumAcademico.



3.1.43.2 Entradas

Por el sistema:

⇒ nombrecarrera: Representa el nombre de la carrera de la cual se desea conocer el pénsum académico.

3.1.43.3 Proceso

Se mostrará al usuario el formulario donde deberá introducir el nombre correspondiente a la carrera de la cual desee conocer su Pénsum Académico. A continuación deberá pulsar el botón Buscar para obtener el informe del Pénsum académico asociado a dicho nombre de carrera.

3.1.43.4 Salidas

Se mostrará por pantalla un reporte que contendrá: los datos de la carrera seleccionada, nombreasignatura, tipoasignatura, prerequisito.

3.1.44 REPORTE DOCENTE

3.1.44.1 Especificación:

3.1.44.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen a un Docente de la Facultad de Ciencias. Esta función tendrá como soporte la entidad Docente.

3.1.44.2 Entradas

Por el sistema:

⇒ nombredocente: Este campo representa el nombre del docente del cual se desea obtener un reporte.



3.1.44.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre del Docente del cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener la información general asociada a dicho docente.

3.1.44.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos del docente: noinss, nombredocente, categoría, tipo, telefono, correo.

3.1.45 REPORTE DOCENTES DEPARTAMENTO

3.1.45.1 Especificación:

3.1.45.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todos los docentes que pertenecen a determinado departamento de la Facultad de Ciencias. Esta función estará soportada por la entidad Docente, Departamento.

3.1.45.2 Entradas

Por el sistema:

⇒ nombredepartamento: Este campo representa el nombre de departamento del cual se desea obtener un reporte con los docentes que pertenecen a dicho departamento.

3.1.45.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Docentes por Departamento**, aparecerá un formulario donde podrá seleccionar de una lista el nombre del departamento del cual



desea obtener la información general de todos los docentes que pertenecen a dicho departamento de la Facultad de Ciencias.

3.1.45.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: datos del departamento, noins, nombredocente, categoría, tipo, telefono, correo.

3.1.46 REPORTE DOCENTES FACULTAD

3.1.46.1 Especificación:

3.1.46.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todos los docentes que laboran en la Facultad de Ciencias. Esta función estará soportada por la entidad Docente.

3.1.46.2 Entradas

Ninguna.

3.1.46.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Docentes de la Facultad de Ciencias** para obtener la información general de todos los docentes de la Facultad de Ciencias.

3.1.46.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos para cada docente: noins, nombredocente, categoría, tipo, telefono, correo.



3.1.47 REPORTE AULA

3.1.47.1 Especificación:

3.1.47.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen y ubican perfectamente un aula de clase | laboratorio de la Facultad de Ciencias, para ello se ejecutarán la(s) consultas correspondientes. Esta función tendrá como soporte la entidad Aula.

3.1.47.2 Entradas

Por el sistema:

⇒ nombreaula: Este campo representa el nombre del aula de la cual se desea obtener un reporte.

3.1.47.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre del aula de la cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener la información general asociada a dicha aula de clase o laboratorio de la Facultad de Ciencias.

3.1.47.4 Salidas

Se mostrará por pantalla un reporte que contendrá los datos del aula seleccionada: codigoaula, nombreaula, tipo_aula, no Equipos, ubicacion, capacidad, observación.



3.1.48 REPORTE AULAS DEPARTAMENTO

3.1.48.1 Especificación:

3.1.48.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todas las aulas de clase y/o laboratorios que pertenecen a determinado departamento de la Facultad de Ciencias. Esta función estará soportada por la entidad Aula, AulasAsignadas.

3.1.48.2 Entradas

Por el sistema:

⇒ nombreddepartamento: Este campo representa el nombre de departamento del cual se desea obtener un reporte con las aulas que han sido asignadas a dicho departamento.

3.1.48.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Aulas por Departamento**, aparecerá un formulario donde podrá seleccionar de una lista el nombre del departamento del cual desea obtener la información general de todas las aulas que han sido asignadas a dicho departamento de la Facultad de Ciencias.

3.1.48.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: datos del departamento, codigoaula, nombreaula, tipo_aula, no Equipos, ubicacion, capacidad, observación.



3.1.49 REPORTE AULAS FACULTAD

3.1.49.1 Especificación:

3.1.49.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todas las aulas de la Facultad de Ciencias. Esta función estará soportada por la entidad Aula.

3.1.49.2 Entradas

Ninguna.

3.1.49.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Aulas de la Facultad de Ciencias** para obtener la información general de todas las aulas de la Facultad de Ciencias.

3.1.49.4 Salidas

Se mostrará por pantalla un reporte que contendrá los siguientes datos para cada registro aula: `codigoaula`, `nombreaula`, `tipo_aula`, `no_equipos`, `ubicacion`, `capacidad`, `observación`.

3.1.50 REPORTE ASIGNATURA

3.1.50.1 Especificación:

3.1.50.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen cada una de las asignaturas que se imparten en la Facultad de Ciencias, para ello se ejecutarán la(s) consultas correspondientes. Esta función tendrá como soporte la entidad Asignatura.



3.1.50.2 Entradas

Por el sistema:

⇒ nombreasignatura: Este campo representa el nombre de la asignatura de la cual se desea obtener un reporte.

3.1.50.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre de la asignatura de la cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener la información general asociada a dicha asignatura impartida en la Facultad de Ciencias.

3.1.50.4 Salidas

Se mostrará por pantalla un reporte que contendrá los datos de la asignatura seleccionada: codigoasignatura, nombreasignatura, no_creditos, observación.

3.1.51 REPORTE ASIGNATURAS DEPARTAMENTO

3.1.51.1 Especificación:

3.1.51.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todas las asignaturas que imparte determinado departamento de la Facultad de Ciencias. Esta función estará soportada por la entidad Asignatura, Departamento

3.1.51.2 Entradas

Por el sistema:

⇒ nombredepartamento: Este campo representa el nombre de departamento del cual se desea obtener un reporte con las asignaturas que imparte dicho departamento.



3.1.51.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Asignaturas por Departamento**, aparecerá un formulario donde podrá seleccionar de una lista el nombre del departamento del cual desea obtener la información general de todas las asignaturas que imparte dicho departamento de la Facultad de Ciencias.

3.1.51.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: datos del departamento, codigoasignatura, nombreasignatura, no_creditos, observación.

3.1.52 REPORTE ASIGNATURAS FACULTAD

3.1.52.1 Especificación:

3.1.52.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todas las asignaturas impartidas en la Facultad de Ciencias. Esta función estará soportada por la entidad Asignatura.

3.1.52.2 Entradas

Ninguna.

3.1.52.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Asignaturas de la Facultad de Ciencias** para obtener la información general de todas las asignaturas impartidas en la Facultad de Ciencias.



3.1.52.4 Salidas

Se mostrará por pantalla un reporte que contendrá los siguientes datos para cada registro asignatura: nombredepartamento, codigoasignatura, nombreasignatura, no_creditos, observación.

3.1.53 REPORTE ASIGNACION AULA

3.1.53.1 Especificación:

3.1.53.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen el horario de asignación semanal de un aula para determinado departamento de la Facultad de Ciencias. Esta función tendrá como soporte la entidad Departamento, Aula, AulasAsignadas.

3.1.53.1.2 Entradas

Por el sistema:

⇒ nombreaula: Representa el nombre de aula de la cual se desea conocer el horario de asignación semanal.

3.1.53.2 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el nombre correspondiente al Aula de la cual desea conocer la descripción del horario de asignación semanal que tiene dicha aula de clase o laboratorio para determinado departamento de la Facultad de Ciencias.

3.1.53.3 Salidas

Se mostrará por pantalla una página que contendrá: los datos del aula seleccionada, nombredepartamento, hora_inicio, hora_fin.



3.1.54 REPORTE GRUPO CLASE

3.1.54.1 Especificación:

3.1.54.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen a cada grupo de clase de la Facultad de Ciencias, para ello se ejecutarán la(s) consultas correspondientes. Esta función tendrá como soporte la entidad GrupoClase.

3.1.54.2 Entradas

Por el sistema:

⇒ **codigogrupo:** Este campo representa el código del grupo del cual se desea obtener un reporte.

3.1.54.3 Proceso

Se mostrará al usuario el formulario donde deberá seleccionar de una lista el código de grupo del cual desee conocer sus datos. A continuación deberá pulsar el botón Buscar para obtener la información general asociada a dicho grupo de clase de la Facultad de Ciencias.

3.1.54.4 Salidas

Se mostrará por pantalla un reporte que contendrá los datos del grupo seleccionado: **codigoasignatura**, **no_inss**, **codigogrupo**, **nombregrupo**, **noestudiantes**, **tipo**, **anyo_lectivo**, **semestreacademico**.



3.1.55 REPORTE GRUPOS CLASE POR ASIGNATURA

3.1.55.1 Especificación:

3.1.55.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todos los grupos de clase que se formen por cada una de las asignaturas que se imparten en la Facultad de Ciencias. Esta función estará soportada por la entidad GrupoClase, Asignatura, Docente

3.1.55.2 Entradas

Por el sistema:

⇒ nombreasignatura: Este campo representa el nombre de la asignatura de la cual se desea obtener un reporte con los datos de los grupos de clase formados para dicha asignatura.

3.1.55.3 Proceso

⇒ Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Grupos de Clase por Asignatura**, aparecerá un formulario donde podrá seleccionar de una lista el nombre de la asignatura de la cual desea obtener la información general de todos de los grupos de clase formados para dicha asignatura.

3.1.55.4 Salidas

Se mostrará por pantalla una página que contendrá los siguientes datos: codigogrupo, nombregrupo, noestudiantes, tipo, anyo_lectivo, semestreadademico, codigoasignatura, no_inss.



3.1.56 REPORTE GRUPOS DE CLASE FACULTAD

3.1.56.1 Especificación:

3.1.56.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a todos los grupos de clase de la Facultad de Ciencias. Esta función estará soportada por la entidad GrupoClase.

3.1.56.2 Entradas

Ninguna.

3.1.56.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Grupos de Clase de la Facultad de Ciencias** para obtener la información general de los grupos de clase de la Facultad de Ciencias.

3.1.56.4 Salidas

Se mostrará por pantalla un reporte que contendrá los siguientes datos para cada registro grupo de clase: codigogrupo, nombregrupo, noestudiantes, tipo, anyo_lectivo, semestreacademico, codigoasignatura, no_inss.

3.1.57 REPORTE CARGA ACADEMICA DOCENTE

3.1.57.1 Especificación:

3.1.57.1.1 Introducción:

Esta función permitirá al usuario visualizar de forma detallada los atributos que describen la carga académica de un Docente de la Facultad de Ciencias, para ello se ejecutarán la o las consultas correspondientes. Esta función tendrá como soporte la entidad Docente, GrupoClase, Planificacion.



3.1.57.2 Entradas

Por el sistema:

⇒ nombredocente: Representa el nombre del docente del cual se desea obtener un reporte con la carga académica asociada.

3.1.57.3 Proceso

Se mostrará al usuario la interfaz principal, donde seleccionará del menú la opción Reportes. A continuación deberá escoger: **Carga Académica de un Docente** aparecerá un formulario donde podrá seleccionar de una lista el nombre del docente del cual desea obtener la información general de la carga académica asociada a dicho docente de la Facultad de Ciencias.

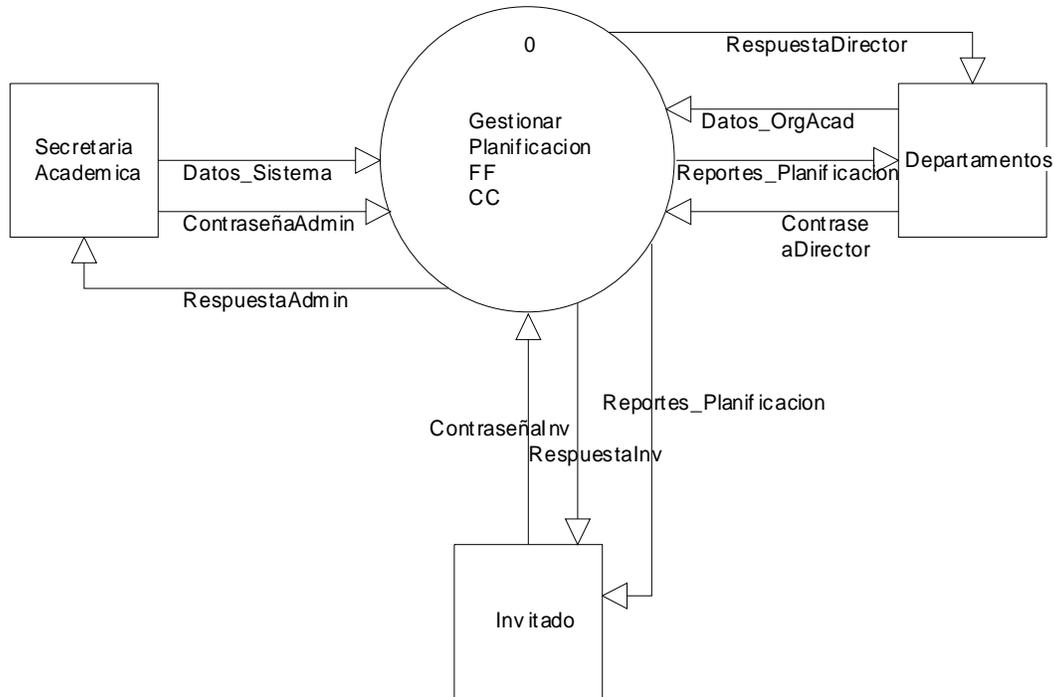
3.1.57.4 Salidas

Se mostrará por pantalla un reporte que contendrá los siguientes datos con la carga horaria del docente seleccionado: día, hora, grupo de clase a impartir, asignatura a impartir.



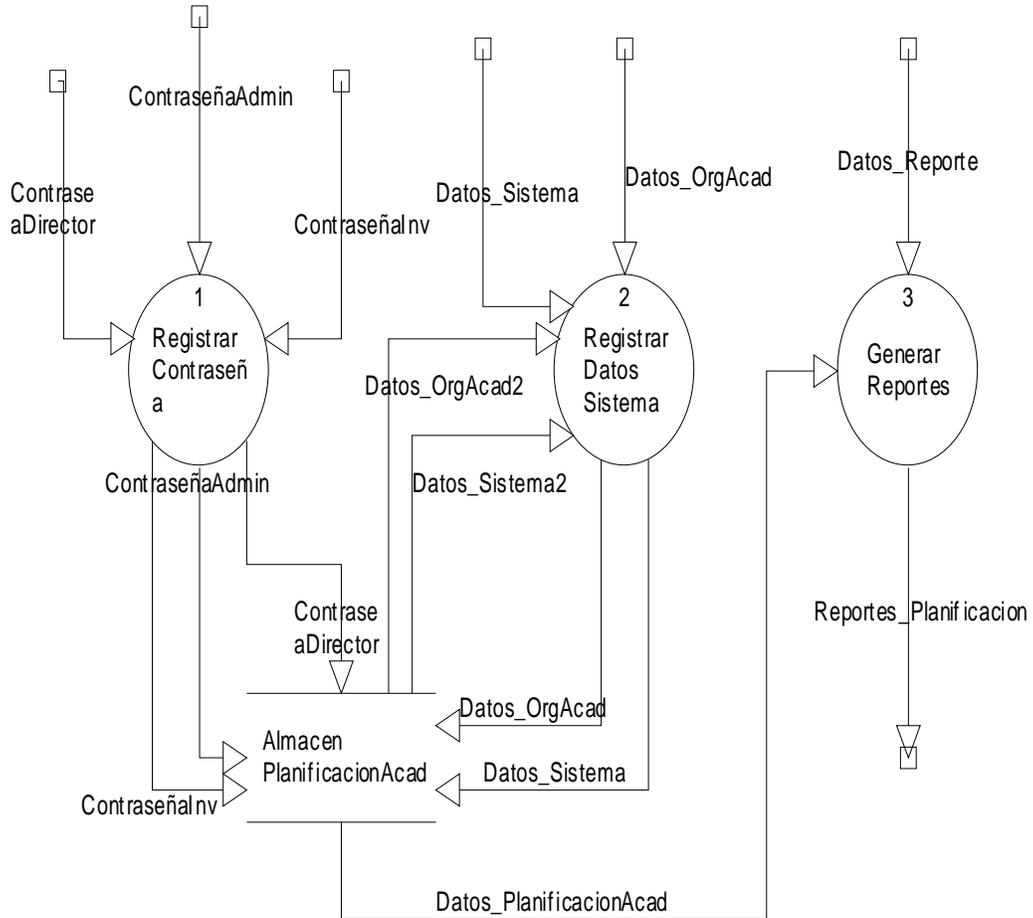
7.1 DIAGRAMA DE FLUJO DE DATOS

NIVEL 0



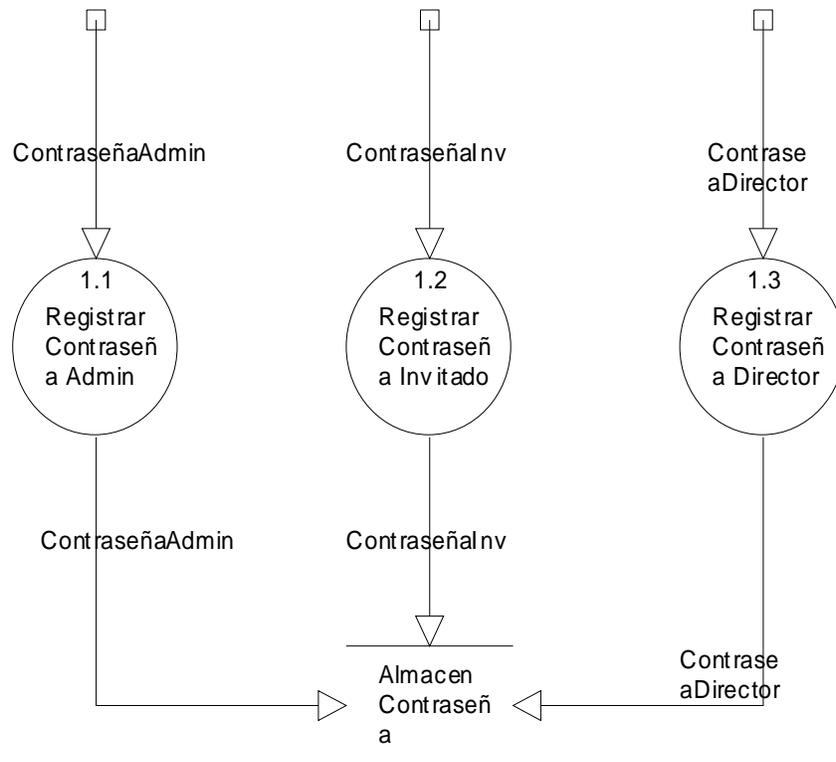


NIVEL 1 DIAGRAMA 1



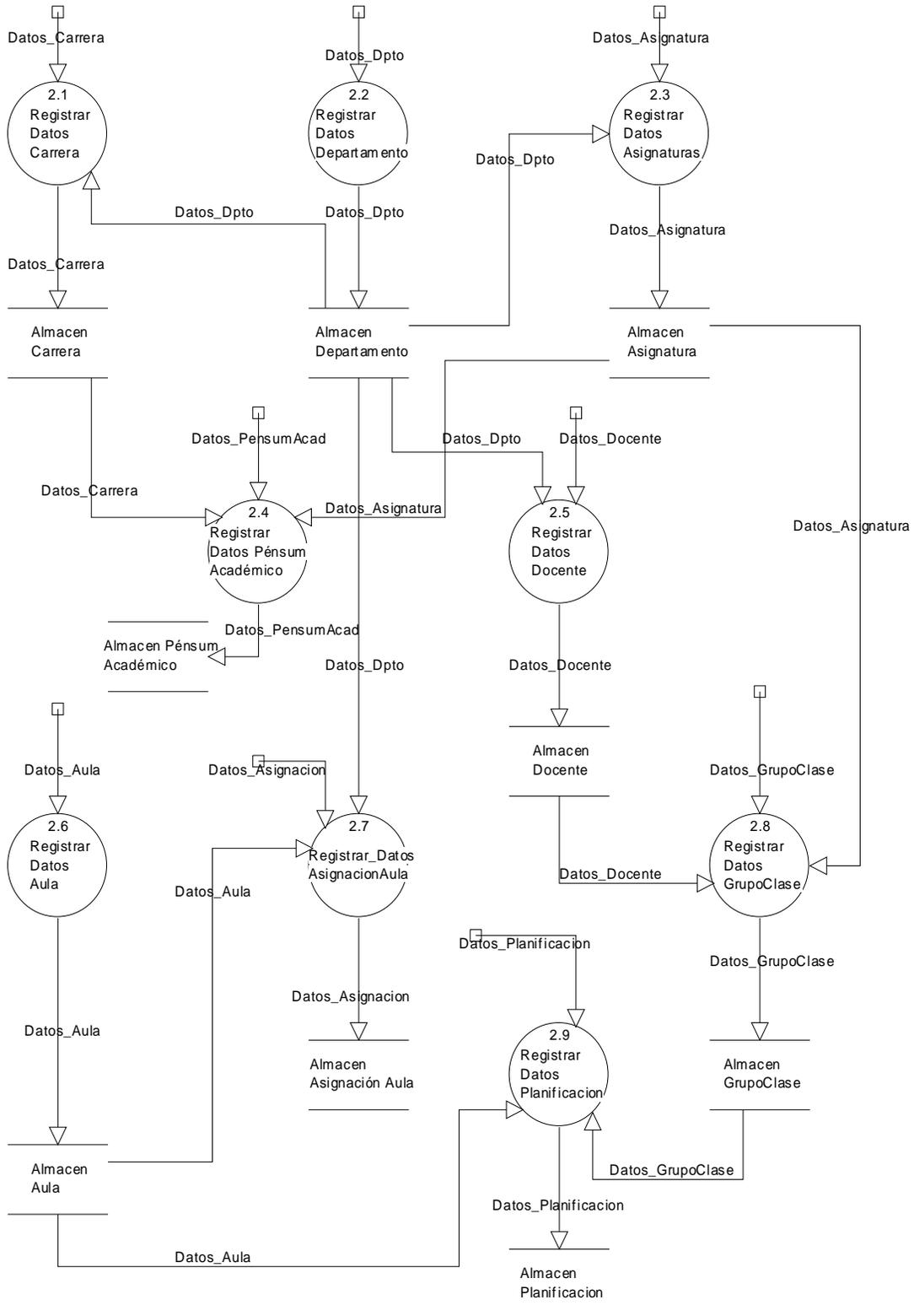


NIVEL 2 DIAGRAMA 1



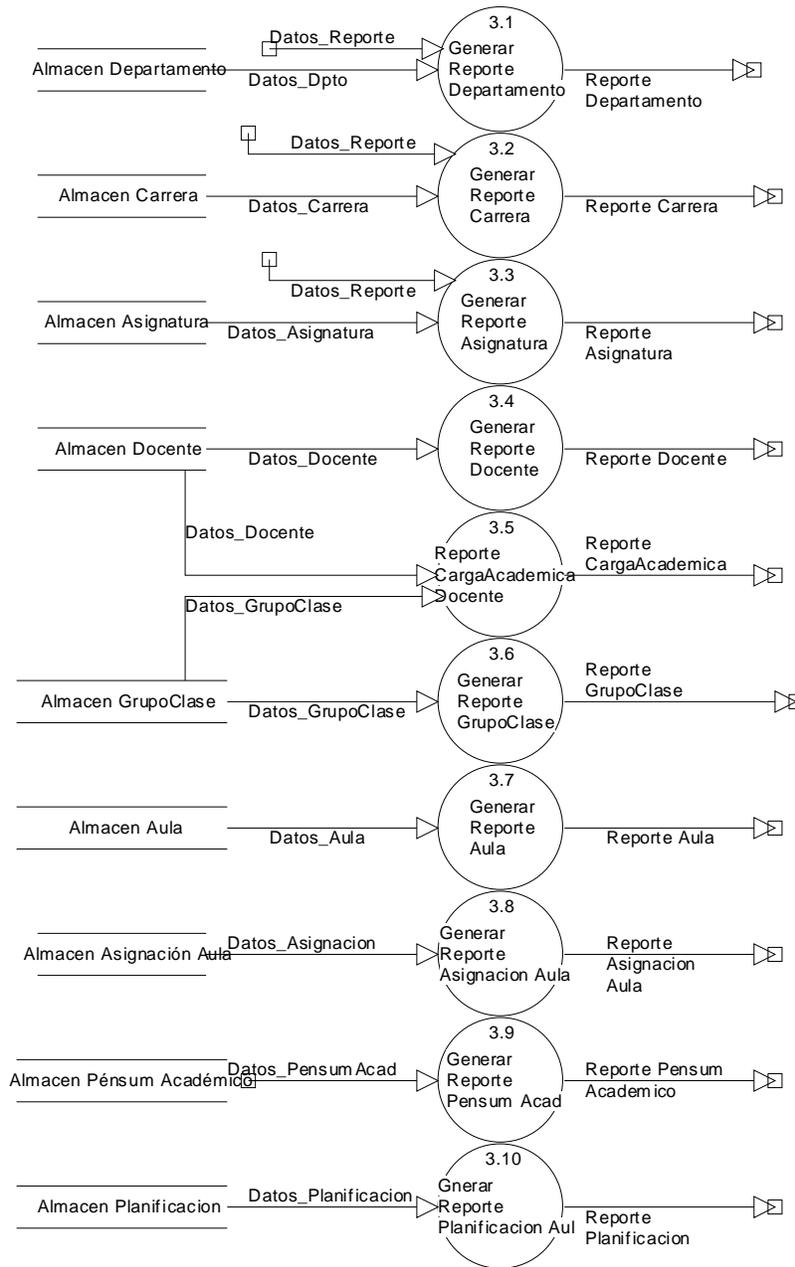


NIVEL 2 DIAGRAMA 2



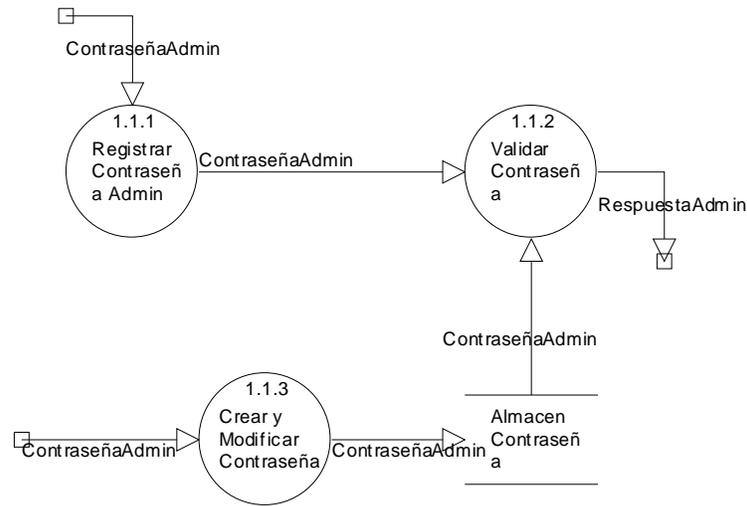


NIVEL 2 DIAGRAMA 3

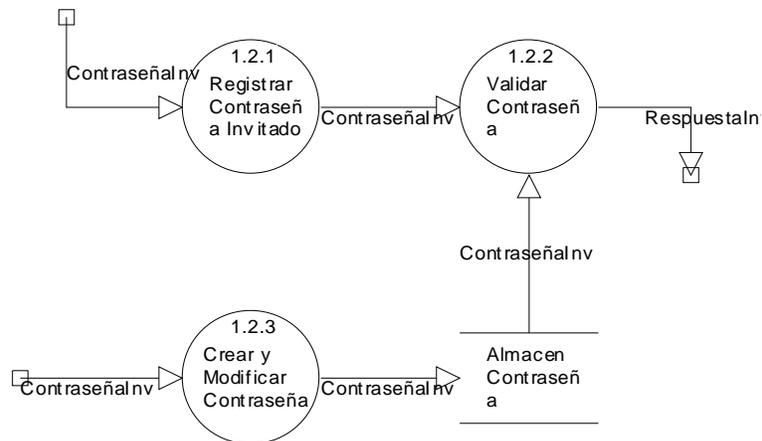




NIVEL 3 DIAGRAMA 1.1

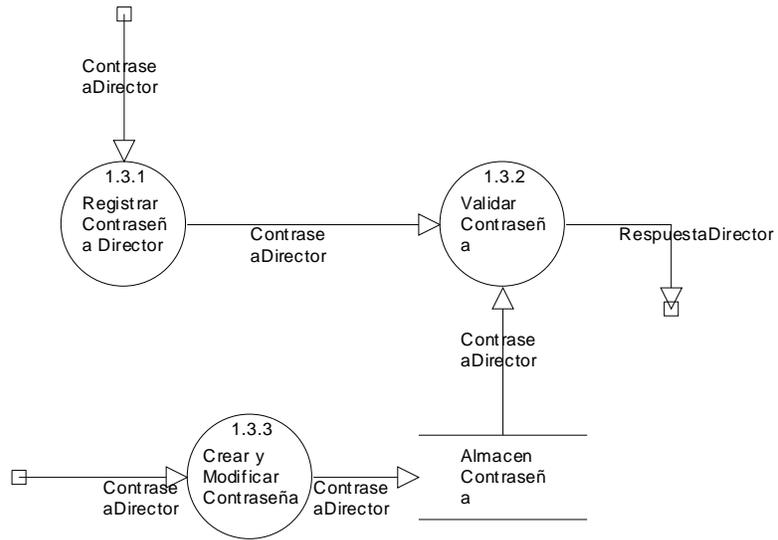


NIVEL 3 DIAGRAMA 1.2





NIVEL 3 DIAGRAMA 1.3





7.1.1 DICCIONARIO DE DATOS

Datos_Sistema = Datos_Dpto + Datos_Carrera + Datos_ Asignatura + Datos_Docente + Datos_Aula + Datos_PensumAcad + Datos_Asignacion + Datos_GrupoClase + Datos_Planificacion + Datos_CursosDocente + Datos_Reporte

Datos_Sistema2 = Datos_Dpto + Datos_Carrera + Datos_ Asignatura + Datos_Docente + Datos_Aula + Datos_PensumAcad + Datos_GrupoClase

Datos_OrgAcad = Datos_GrupoClase + Datos_Planificacion + Datos_CursosDocente + Datos_Reporte

Datos_OrgAcad 2 = Datos_GrupoClase

Datos_Dpto = nombredpto + nombredirector + anyofundacion + extension + observacion

Datos_Carrera = nombrecarrera + anyofundacion + cantcreditoblig + cantcreditopt + cantcreditoselectivo + totalcreditocarrera + observación + nombredpto

Datos_Asignatura = codigoasignatura + nombreasignatura + nocreditos + observación + nombredpto

Datos_PensumAcad = nombrecarrera + codigoasignatura + tipo + prerequisito

Datos_Docente = no_inss + nombredocente + categoria + tipo+ telefono + correo + nombredpto

Datos_CursosDocente = no_inss + nombrecurso + anyo + lugar

Datos_Aula = codigoaula + nombreaula + tipo + no Equipos + capacidad + observación

Datos_Asignacion = codigoaula + nombredpto + hora_inicio + hora_fin

Datos_GrupoClase = codigogrupo + nombregrupo + nocreditos + noestudiantes + tipo + anyolectivo + semestreacademico + codigoasigantura + no_inss

Datos_Planificacion = codigoaula + codigogrupo + dia + horainicio

Datos_Reporte = nombredpto + nombrecarrera + nombredocente + codigogrupo + nombreaula + nombreasignatura

Almacen PlanificacionAcad = ContraseñaAdmin + ContraseñaInv + ContraseñaDirector + Datos_Sistema + Datos_OrgAcad



Almacen PlanificacionAcad = Almacen Carrera + Almacen Departamento + Almacen Asignatura + Almacen Pénsum Academico + Almacen Docente + Almacen Aula + Almacen Asignación Aula + Almacen GrupoClase + Almacen Planificacion

Almacen Contraseña = ContraseñaAdmin + ContraseñaInv + ContraseñaDirector

Almacen Carrera = Datos_Carrera

Almacen Departamento = Datos_Dpto

Almacen Asignatura = Datos_Asignatura

Almacen Pénsum Academico = Datos_PensumAcad

Almacen Docente = Datos_Docente

Almacen Aula = Datos_Aula

Almacen Asignación Aula = Datos_Asignacion

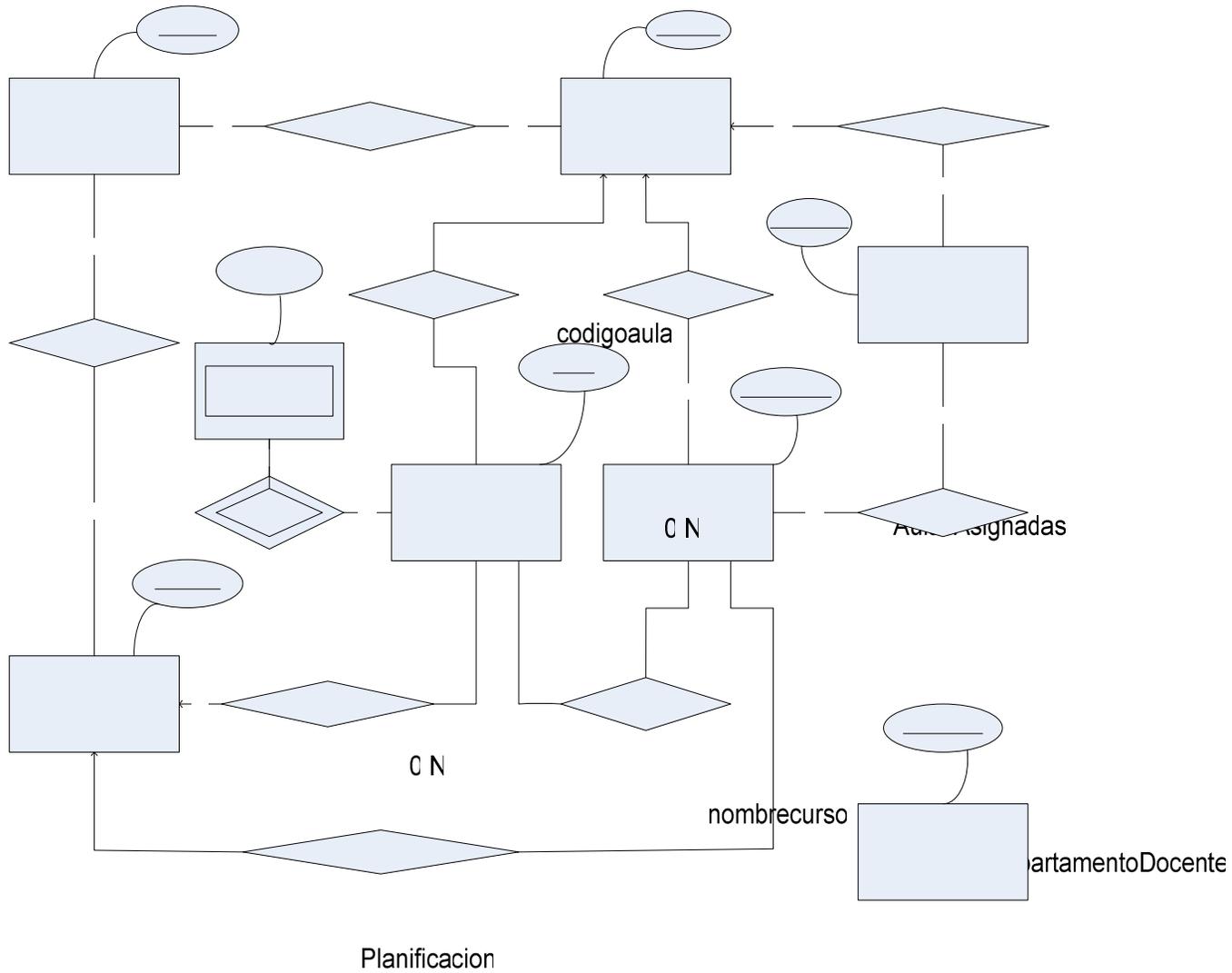
Almacen GrupoClase = Datos_GrupoClase

Almacen Planificacion = Datos_Planificacion

Reportes_Planificacion = Reporte Departamento + Reporte Carrera + Reporte Asignatura + Reporte Docente + Reporte GrupoClase + Reporte PlanificacionAcad + Reporte Aula + Reporte Asignación + Reporte PensumAcademico + Reporte CargaAcademica



7.2 DIAGRAMA ENTIDAD RELACIÓN



Cursos_Realizados

1 1

1 N

1 N

Realiza

1 N

Docente

codigogrupo



7.3 DISEÑO

DISEÑO DE DATOS

1. Tabla aula		
Nombre Atributo	Tipo	Tamaño
codigoaula	varchar	10
nombreaula	varchar	60
ubicacion	varchar	60
tipo_aula	varchar	13
no_equipos	smallint	
capacidad	smallint	
observacion	varchar	60

2. Tabla aulas_asignadas		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
codigoaula	varchar	10
hora_inicio	time	
hora_fin	time	

3. Tabla departamento		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
anyofundacion	smallint	
nombredirector	varchar	60
extension	varchar	4
observacion	varchar	60

4. Tabla departamentocarrera		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
nombrecarrera	varchar	80



5. Tabla carrera		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
nombrecarrera	varchar	80
anyofundacion	smallint	
cantcreditoblig	smallint	
cantcreditopt	smallint	
cantcreditoselectivo	smallint	
totalcreditoscarrera	smallint	
observacion	varchar	60

6. Tabla departamentodocente		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
no_inss	varchar	16

7. Tabla docente		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
no_inss	varchar	16
nombredocente	varchar	80
categoria	varchar	10
tipo	varchar	8
horas_semana	smallint	
telefono	varchar	7
correo	varchar	50

8. Tabla cursos_realizados		
Nombre Atributo	Tipo	Tamaño
no_inss	varchar	16
nombrecurso	varchar	80
anyo	smallint	
lugar	varchar	60

9. Tabla imparte		
Nombre Atributo	Tipo	Tamaño
no_inss	varchar	16
codigoasignatura	varchar	20



10. Tabla asignatura		
Nombre Atributo	Tipo	Tamaño
nombredpto	varchar	36
codigoasignatura	varchar	20
nombreasignatura	varchar	80
no_creditos	smallint	
observacion	varchar	60

11. Tabla departamentoimparte		
Nombre Atributo	Tipo	Tamaño
codigoasignatura	varchar	20
nombredpto	varchar	36

12. Tabla pensum_academico		
Nombre Atributo	Tipo	Tamaño
nombrecarrera	varchar	80
codigoasignatura	varchar	20
tipo	varchar	12
prerequisito	varchar	20

13. Tabla asignaturagrupo		
Nombre Atributo	Tipo	Tamaño
codigoasignatura	varchar	20
codigogrupo	varchar	20

14. Tabla docentegrupo		
Nombre Atributo	Tipo	Tamaño
no_inss	varchar	16
codigogrupo	varchar	20



15. Tabla grupo_clase		
Nombre Atributo	Tipo	Tamaño
codigoasignatura	varchar	20
no_inss	varchar	16
codigogrupo	varchar	20
nombregrupo	char	1
noestudiantes	smallint	
tipo	varchar	11
anyo_lectivo	smallint	
semestreacademico	varchar	2

16. Tabla planificación		
Nombre Atributo	Tipo	Tamaño
codigoaula	varchar	10
codigogrupo	varchar	20
dia	varchar	9
horainicio	time	

17. Tabla usuarios		
Nombre Atributo	Tipo	Tamaño
nombreusuario	varchar	25
password	varchar	25
tipousuario	varchar	15



8. CONCLUSIONES

Al concluir con nuestro trabajo-tesis consideramos que hemos cumplido con los objetivos planteados, al proporcionar un documento que brinda conocimientos básicos sobre la utilización de “Software Libre” como herramienta de desarrollo para la aplicación que facilitará la gestión de la información que genera la Planificación Académica de la Facultad de Ciencias enfocada a la implementación de los Sistemas de Créditos Académicos.

Al llevar a cabo el desarrollo de nuestro Sistema y haber utilizado **Linux Debian** (Sistema Operativo), **PHP** (Lenguaje de Programación), **PostgreSQL** (Sistema Gestor de Base de Datos), **Apache** (Servidor), **Quanta Plus** (Editor de código HTML) y **Eclipse** (Entorno de Desarrollo), concluimos que son herramientas potentes que permiten crear de forma sencilla y rápida sistemas de cualquier índole.

Gracias a la utilización de estas herramientas (PHP, PostgreSQL, Apache, Eclipse, Quanta) se concluyó con el desarrollo de la aplicación para la Planificación Académica de la Facultad de Ciencias a través de Software Libre, propuesta como objetivo fundamental de nuestro trabajo monográfico.



9. RECOMENDACIONES

Al concluir nuestro trabajo monográfico recomendamos:

- Implementar un protocolo de transferencia de datos a través de la red que garantice la integridad de los mismos.
- Promover entre los estudiantes del Departamento de Computación el desarrollo de sistemas a través de Software Libre.
- Implementar las reglas establecidas por la secretaría Académica para la codificación de las asignaturas.



10.BIBLIOGRAFÍA

- <http://es.wikipedia.org/wiki/PostgreSQL>
- <http://es.tldp.org/Postgresql-es/web/navigable/tutorial/x56.html>
- <http://www.gsl.unt.edu.ar/desc/Festival/diapositivas/PostgreSQL.pdf>
- www.postgresql.org
- www.milonic.com
- www.nvu.com
- www.apache.org
- www.php.net
- www.desarrolloweb.com
- http://es.wikipedia.org/wiki/Internet_Information_Server



ANEXOS



PANTALLA PRINCIPAL





PANTALLA (Usuario Administrador)





PANTALLA (Usuario Director)





PANTALLA (Usuario Invitado)





PANTALLA (Registrar Nuevo Departamento)

Nuevo Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/NuevoDepartamento.php

Getting Started Latest Headlines

 *Nuevo Registro: Departamento* 

Nombre Departamento:	Matemática Estadística Actuariales
Año Fundación	2006
Nombre Director:	Msc, Alberto Cerda
Extensión Telefónica:	1109
Observaciones:	Está ubicado Frente al Edificio Central 2 Piso

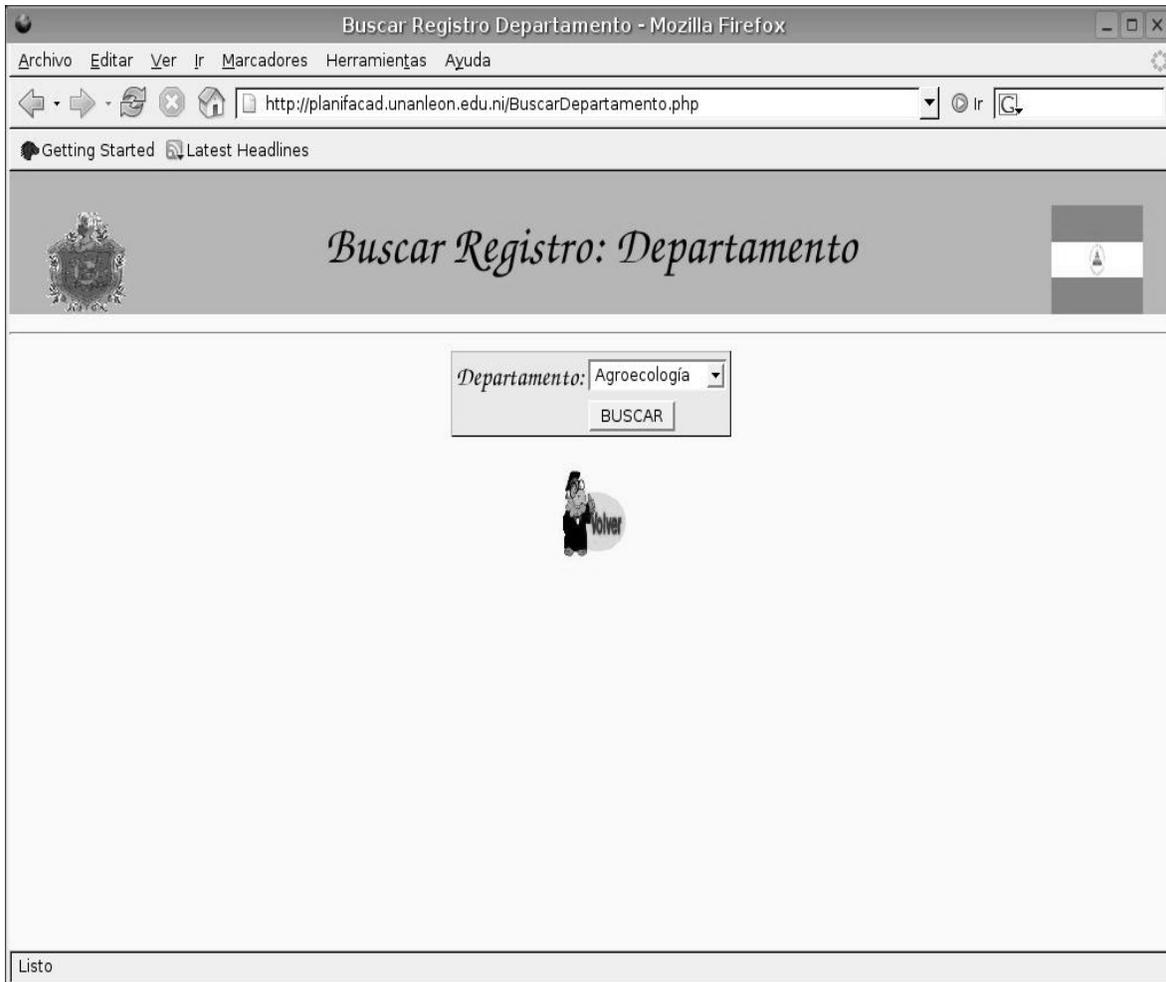
GUARDAR



Listo



PANTALLA (Buscar Registro Departamento)





PANTALLA (Información General de un Departamento)

<i>Departamento</i>	<i>Año Fundación</i>	<i>Director</i>	<i>Extensión</i>	<i>Observaciones</i>
<i>Agroecología</i>	<i>2000</i>	<i>Ing. Eva Gutiérrez G</i>	<i>1779</i>	<i>Ninguna</i>



PANTALLA (Borrar Registro Departamento)

Borrar Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/BorrarRegistroDepartamento.php

Getting Started Latest Headlines

 *Borrar Registro: Departamento* 

Departamento: Matemática Estadística Actuariales

BORRAR



Listo



PANTALLA (Borrar Registro Departamento 2)

Borrando Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/pprincipaldpto.php

Getting Started Latest Headlines

Borrar Registro: Departamento

Nombre Departamento: Matemática Estadística Actuariales

Año Fundación: 1987

Nombre Director: Alberto Cerda

Extensión Telefónica: 1109

Observaciones: Ninguna

BORRAR

[Aplicación JavaScript]

¿Desea Borrar el Registro Actual?

Cancelar Aceptar

Listo



PANTALLA (Borrar Registro Departamento 3)

Borrando Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/pprincipaldpto.php

Getting Started Latest Headlines

Borrar Registro: Departamento

Nombre Departamento: Matemática Estadística Actuariales

Año Fundación: 1987

Nombre Director: Alberto Cerda

Extensión Telefónica: 1109

Observaciones: Ninguna

[Aplicación JavaScript]

! Registro Borrado

Transfiriendo datos desde planifacad.unanleon.edu.ni...



PANTALLA (Editar Registro Departamento 1)

Editar Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/EditarRegistro.php

Getting Started Latest Headlines

 *Editar Registro: Departamento* 

Departamento: Agroecología



Listo



PANTALLA (Editar Registro Departamento 2)

Editar Registro Departamento - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

http://planifacad.unanleon.edu.ni/pprincipaldpto.php

Getting Started Latest Headlines

Editar Registro: Departamento

Nombre Departamento:	<input type="text" value="Agroecología"/>
Año Fundación:	<input type="text" value="2001"/>
Nombre Director:	<input type="text" value="Ing.Eva Gutiérrez G"/>
Extensión Telefónica:	<input type="text" value="1779"/>
Observaciones:	<input type="text" value="Ninguna"/>



Listo



PANTALLA (Editar Registro Departamento 3)

The screenshot shows a Mozilla Firefox browser window titled "Editar Registro Departamento". The address bar contains the URL "http://planifacad.unanleon.edu.ni/pprincipaldpto.php". The page header features the university's coat of arms and the text "Editar Registro: Departamento".

The main content area contains a form with the following fields:

- Nombre Departamento: Agroecología
- Año Fundación: 2001
- Nombre Director: Ing.Eva Gutiérrez G
- Extensión Telefónica: 1779
- Observaciones: Ninguna

Below the form is a "GUARDAR" button. A JavaScript dialog box is overlaid on the form, titled "[Aplicación JavaScript]", with the message "¿Desea Guardar los cambios en el Registro Actual?" and "Cancelar" and "Aceptar" buttons.

At the bottom left of the page, the text "Listo" is visible.



PANTALLA (Editar Registro Departamento 4)

The screenshot shows a Mozilla Firefox browser window titled "Editar Registro Departamento - Mozilla Firefox". The address bar displays the URL "http://planifacac.unanleon.edu.ni/pprincipaldpto.php". The page content features a header with the text "Editar Registro: Departamento" and a logo on the left. Below the header is a form with the following fields:

- Nombre Departamento: Agroecología
- Año Fundación: 1999
- Nombre Director: Ing.Eva Gutiérrez G
- Extensión Telefónica: 1779
- Observaciones: Ninguna

A "GUARDAR" button is located below the "Observaciones" field. A JavaScript alert dialog box is overlaid on the form, titled "[Aplicación JavaScript]", with the message "Registro Modificado" and an "Aceptar" button. At the bottom of the browser window, a status bar indicates "Transfiriendo datos desde planifacac.unanleon.edu.ni...".



CODIFICACIÓN:

//Funciones Departamento

```
<?php
function vermensaje($mensaje)
{
    echo "<script type 'text/javascript'> alert ('$mensaje');</script> ";
}

function atras($mensaje)
{
    echo "<script type 'text/javascript'>window.history.back(); </script> ";
}

function cargaranyo($etiqueta,$combo,$anyo,$vdefecto)
{
    echo "<tr>";
    echo "<td>$etiqueta</td>";
    echo "<td><select name=\"\$combo\">";
    for($n=0;$n<sizeof($anyo);$n++)
    {
        if($vdefecto == $anyo[$n])
            echo "<option value=$anyo[$n] selected>$anyo[$n]</option>";
        else
            echo "<option value=$anyo[$n]>$anyo[$n]</option>";
    }
    echo "</select></td></tr>";
}

function VerificarExistencia($tabla,$campobusqueda,$cadenabuscar)
{
    global $cadenaconexcion;
    $conexion = pg_connect($cadenaconexcion);
    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql = "select * from $tabla where $campobusqueda = '$cadenabuscar'";
    $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
    pg_last_error($conexion));
    $numregistros = pg_num_rows($ResultadoBusqueda);
}
```



```
if($numregistros == 1)
{
    return 1;
}
else
{
    return 0;
}
}

function InsertarNuevoDepartamento()
{
    global $cadenaconexcion;
    global $nombredpto,$anyofundacion,$nombredirector,$observacion,$extension;

    $resul = VerificarExistencia("departamento","nombredpto",$nombredpto);
    if( !$resul )
    {
        $conexion = pg_connect($cadenaconexcion);
        if($conexion == FALSE)
        {
            vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
            exit();
        }
        $sentsql= "BEGIN TRANSACTION";
        $insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .
        pg_last_error($conexion));

        $sentsql = "insert into departamento
        values('$nombredpto','$anyofundacion','$nombredirector','$extension','$observacion
        ')";
        $insertar = pg_query($conexion,$sentsql) or die("Error en la consulta:$sentsql" .
        pg_last_error($conexion));

        $numreginser = pg_affected_rows($insertar);

        if(!$numreginser)
        {
            $error = pg_last_error($conexion);
            $sentsql="ROLLBACK";
            vermensaje("Registro no Guardado...Inténtelo Nuevamente");
            vermensaje("Hubo algunos errores al intentar añadir los datos a la base de
            datos. Error: $error");
            atras();
        }

        else
    }
}
```



```
{
    $sentsql = "COMMIT";

    if($numreginser == 1)
    {
        vermensaje("Registro Insertado");
        echo"<script>location.href='http://planifacad.unanleon.edu.ni/
        NuevoDepartamento.php' </script>";
    }
}
$insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .
pg_last_error($conexion));
}
else
{
    vermensaje("Error-El departamento:$nombredpto ya existe. Inténtelo
    Nuevamente");
    atras();
}
}
```

```
function BuscarDepartamento()
{
    global $nombredpto;
    global $cadenaconexcion;
    $conexion = pg_connect($cadenaconexcion);

    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql = "select * from departamento where nombredpto='$nombredpto' order by
    nombredpto";
    $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
    pg_last_error($conexion));
    $numregistros = pg_num_rows($ResultadoBusqueda);

    if($numregistros)//si existen registros la variable $numregistros!=0
    {
        VerRegistro($numregistros,$ResultadoBusqueda);
    }
}
```




```
$tabla2 = "</tbody></table></FORM>";

echo $titulo;
echo $tabla;

for($i = 0;$i<$n;$i++)
{
    $registro = pg_fetch_array($ResultadoBusqueda,$i);

    $html = "<tr><td align='center'>$registro[nombredpto]</td><td align='center'>$registro[anyofundacion]</td><td align='left'>$registro[nombredirector]</td><td>$registro[extension]</td><td>$registro[observacion]</td></tr>";
    echo $html;
}

echo $tabla2;
echo "<br>";
echo "<p align='center'>";
echo "<img src='http://planifacad.unanleon.edu.ni/Imagenes/atras.jpg' onclick='volver()' width='65' height='65'>";
echo "</p>";
}

function VerRegistroEditar()
{
    global $cadenaconexcion,$nombredpto;

    $conexion = pg_connect($cadenaconexcion);

    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql = "select * from departamento where nombredpto='$nombredpto'";
    $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta $ResultadoBusqueda" . pg_last_error($conexion));
    $numregistros = pg_num_rows($ResultadoBusqueda);
}
```



```
if($numregistros)//si existen registros la variable $numregistros!=0
{
    $departamento = pg_fetch_array($ResultadoBusqueda,0);
}
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><html><head>
<title>Editar Registro Departamento</title></HEAD><body>

<script language="javascript" type="text/javascript">
function volver()
{
    window.history.go(-2);
}

function validar(fr)
{
    if(fr.nombredpto.value=="")
    {
        alert("Error-El campo Nombre Departamento no puede quedar vacio");
        fr.nombredpto.focus();
        return false;
    }
else
{
    if (fr.nombredpto.value.search(/([\D]+[\s]?)+/)==-1)
    {
        alert("Error-El Campo Nombre Departamento solo puede contener
letras");
        fr.nombredpto.select();
        fr.nombredpto.focus();
        return false;
    }
}

if (fr.nombredirector.value=="")
{
    alert("Error-El campo Nombre Director no puede quedar vacio");
    fr.nombredirector.focus();
    return false;
}
```



```
else
{
    if (fr.nombredirector.value.search(/([\D]+[\s]?)+/)==-1)
    {
        alert("Error-El Campo Nombre Director sólo puede contener letras");
        fr.anyofundacion.select();
        fr.anyofundacion.focus();
        return false;
    }
}

if (fr.extension.value=="")
{
    alert("Error-Número de Extensión no valido");
    fr.extension.focus();
    return false;
}
else
{
    if (fr.extension.value.search(/[\d]{4}/)==-1)
    {
        alert("Error-El campo Extensión debe contener 4 dígitos");
        fr.extension.select();
        fr.extension.focus();
        return false;
    }
}

if(confirm("¿Desea Guardar los cambios en el Registro Actual?") == true)
{
    return true;
}
else
{
    window.history.back();
    return false;
}
}
</script>
```



```
<meta name="GENERATOR" content="Quanta Plus">
<meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

</head>
<body>
<FORM action="pprincipaldpto.php" method="POST" onsubmit="return validar(this)">

<div id="masthead" Editar Registro: Departamento</div>
<HR>

<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">

<tr>
<td>Nombre Departamento:</td>
<td><INPUT type="text" name="nombredpto" size="36" maxlength="36"
value="<?php echo "$departamento[nombredpto]" ?>"></td>
</tr>

<?
$year= array();
for($n=1980;$n<=((int)date("Y")); $n++)
array_push($year,$n);
$year=array_reverse($year);
cargaranyo("AñoFundación:", "anyofundacion", $year,
$departamento[anyofundacion]);
?>

<tr>
<td>Nombre Director:</td>
<td><INPUT type="text" name="nombredirector" size="50" maxlength="60"
value="<?php echo "$departamento[nombredirector]" ?>"></td>
</tr>

<tr>
<td>Extensión Telefónica:</td>
<td><INPUT type="text" name="extension" size="4" maxlength="4"
value="<?php echo "$departamento[extension]" ?>"></td>
</tr>
```



```
<tr>
    <td>Observaciones:</td>
    <td><INPUT type="text" name ="observacion" size="50" maxlength="60"
value="<?php echo "$departamento[observacion]" ?>"></td>
</tr>

<tr>
    <td ><INPUT type="submit" value="GUARDAR"></td>
    <td><INPUT type="hidden" name="npageD" value="6"></td>
    <td><INPUT type="hidden" name="nombredptoEditar" size="36"
maxlength="36" value="<?php echo "$departamento[nombredpto]" ?>"></td>
</tr>

</tbody>
</table>
</FORM>

<p align="center">

</p>
<?php

echo "</body></html>";

} //fin del if

else
{
    vermensaje("El departamento:\'$nombredpto\' no se encuentra en la base de datos");
    echo "<script>location.href = 'http://planifacad.unanleon.edu.ni/EditarRegistro.php'
</script>";
}
}

function GuardarRegistroEditar()
{
    global $cadenaconecion;
    global $nombredpto,$anyofundacion,$nombredirector,$observacion,$extension;
    global $nombredptoEditar;

    $resul = VerificarExistencia("departamento","nombredpto",$nombredpto);
```



```
if( !$resul || ($nombredptoEditar == $nombredpto))
{
    $conexion = pg_connect($cadenaconexion);

    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql= "BEGIN TRANSACTION";
    $insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .
    pg_last_error($conexion));

    $sentsql = "update departamento set
    nombredpto='$nombredpto',anyofundacion='$anyofundacion',nombredirector='$no
    mbredirector',observacion='$observacion' where nombredpto
    ='$nombredptoEditar'";

    $insertar = pg_query($conexion,$sentsql);
    $numreginser = pg_affected_rows($insertar);

    if(!$numreginser)
    {
        $error = pg_last_error($conexion);
        $sentsql="ROLLBACK";
        vermensaje("Registro no Guardado.Intentelo Nuevamente");
        vermensaje("Hubo algunos errores al intentar añadir los datos a la base de
        datos. Error: $error");
        atras();
    }
    else
    {
        $sentsql = "COMMIT";

        if($numreginser == 1)
        {
            vermensaje("Registro Modificado");
            echo "<script>location.href =
            'http://planifacad.unanleon.edu.ni/EditarRegistro.php' </script>";
        }
    }
}
```



```
$insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .  
pg_last_error($conexion));  
}  
else  
{  
    vermensaje("Error- El Nombre de Departamento especificado ya existe");  
    atras();  
}  
}
```

```
function VerRegistroBorrar()  
{  
    global $nombredpto;  
    global $cadenaconexcion;  
  
    $conexion = pg_connect($cadenaconexcion);  
  
    if($conexion == FALSE)  
    {  
        vermensaje("ERROR AL CONECTAR A LA BASE DE DATOS");  
        exit();  
    }  
    $mensaje = "El Departamento: $nombredpto no se puede borrar, es referido desde  
las tablas: ";  
  
    $sentsqlcarrera= "select dpto.nombredpto from departamento as dpto where  
dpto.nombredpto in(select nombredpto from carrera) and  
dpto.nombredpto='$nombredpto';  
  
    $ResultadoCarrera = pg_query($conexion,$sentsqlcarrera) or die("Error en la  
consulta" . pg_last_error($conexion));  
    $carrera= pg_num_rows($ResultadoCarrera);  
  
    $sentsqlaulasasig= "select dpto.nombredpto from departamento as dpto where  
dpto.nombredpto in(select nombredpto from aulas_asignadas) and  
dpto.nombredpto='$nombredpto';  
  
    $Resultadoaulasasig = pg_query($conexion,$sentsqlaulasasig) or die("Error en la  
consulta" . pg_last_error($conexion));  
    $aulasasig= pg_num_rows($Resultadoaulasasig);  
  
    $sentsqldocente= "select dpto.nombredpto from departamento as dpto where  
dpto.nombredpto in(select nombredpto from docente) and  
dpto.nombredpto='$nombredpto';
```



```
$Resultadodocente = pg_query($conexion,$sentsqldocente) or die("Error en la consulta" . pg_last_error($conexion));  
$ndocente= pg_num_rows($Resultadodocente);
```

```
$sentsqlasignatura= "select dpto.nombredpto from departamento as dpto where  
dpto.nombredpto in(select nombredpto from asignatura) and  
dpto.nombredpto='$nombredpto'";
```

```
$Resultadoasignatura = pg_query($conexion,$sentsqlasignatura) or die("Error en la consulta" . pg_last_error($conexion));  
$nasignatura = pg_num_rows($Resultadoasignatura);
```

```
if($ncarrera || $naulasasig || $ndocente || $nasignatura)  
{  
    if($ncarrera != 0)  
        $mensaje .= "carrera";  
    if($naulasasig != 0)  
        $mensaje .= ", aulas_asignadas";  
    if($ndocente != 0)  
        $mensaje .= ", docente";  
    if($nasignatura != 0)  
        $mensaje .= ", asignatura";  
    vermensaje($mensaje);  
    atras();  
}
```

```
$sentsql = "select * from departamento where nombredpto='$nombredpto'";  
$ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .  
pg_last_error($conexion));  
$numregistros = pg_num_rows($ResultadoBusqueda);
```

```
if($numregistros)//si existen registros la variable $numregistros!=0  
{  
    $departamento = pg_fetch_array($ResultadoBusqueda,0);  
    echo "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN\" \"http://www.w3.org/TR/html4/loose.dtd\">  
<html>  
  
<head>  
    <title>Borrando Registro Departamento</title>  
    <meta name=\"GENERATOR\" content=\"Quanta Plus\">  
    <meta name=\"AUTHOR\" content=\"Francisco Castro - Karina  
Esquivel\">
```



```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

<script language="javascript" type="text/javascript">

    function volver()
    {
        window.history.go(-2);
    }

    function ConfirmarBorrado()
    {
        if(confirm("¿Desea Borrar el Registro Actual?") == true)
        {
            return true;
        }
        else
        {
            window.history.back();
            return false;
        }
    }
</script>
</head>
<body>;
```

```
echo "<FORM action="pprincipaldpto.php" method="POST" onsubmit="return ConfirmarBorrado()">
```

```
<div id="masthead"
src="http://planifacad.unanleon.edu.ni/Imagenes/insignia2.jpg" width="80"
height="80" align="left"
src="http://planifacad.unanleon.edu.ni/Imagenes/bandera1.jpg" width="80"
height="80" align="right">Borrar Registro: Departamento</div>
<HR>
```

```
<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">
```



```
<tr>
  <td>Nombre Departamento:</td>
  <td><INPUT type="text" name="nombredpto" size="36" maxlength="36"
  readonly value="$departamento[nombredpto]"></td>
</tr>

<tr>
  <td>Año Fundación:</td>
  <td><INPUT type="text" name="anyofundacion" size="4" maxlength="4"
  readonly value=$departamento[anyofundacion]></td>
</tr>

<tr>
  <td>Nombre Director:</td>
  <td><INPUT type="text" name="nombredirector" size="50" maxlength="60"
  readonly value="$departamento[nombredirector]"></td>
</tr>

<tr>
  <td>Extensión Telefónica:</td>
  <td><INPUT type="text" name="extension" size="4" maxlength="4"
  readonly value="$departamento[extension]"></td>
</tr>

<tr>
  <td>Observaciones:</td>
  <td><INPUT type="text" name="observacion" size="50" maxlength="60"
  readonly value="$departamento[observacion]"></td>
</tr>

<tr>
  <td><INPUT type="submit" value="BORRAR"></td>
  <td><INPUT type="hidden" name="npageD" value="3"></td>
</tr>

</tbody>
</table>
</FORM>;

echo "<br>";
echo "<p align =\"center\">";
echo      "<img      src=\"http://planifacad.unanleon.edu.ni/Imagenes/atras.jpg\"
onclick=\"volver()\" width=\"65\" height=\"65\">";
echo "</p>";
```



```
echo "</body></html>";
} //fin del if

else
{
    vermensaje("El departamento:\\"$nombredpto\" no se encuentra en la base de datos");
    echo "<script type 'text/javascript'>window.history.back();</script> ";
}
}

function BorrarRegistroDepartamento()
{
    global $nombredpto;
    global $cadenaconexcion;

    $conexion = pg_connect($cadenaconexcion);
    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql= "BEGIN TRANSACTION";
    $insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .
    pg_last_error($conexion));

    $sentsql = "delete from departamento where nombredpto ='$nombredpto'";
    $insertar = pg_query($conexion,$sentsql) or die("Error en la consulta" .
    pg_last_error($conexion));
    $numreginser = pg_affected_rows($insertar);

    if(!$numreginser)
    {
        $error = pg_last_error($conexion);
        $sentsql="ROLLBACK";
        vermensaje("Registro no Guardado...Intentelo Nuevamente");
        vermensaje("Hubo algunos errores al intentar Borrar los datos a la base de
        datos. Error: $error");
        atras();
    }
}
```



```
else
{
    $sentsql = "COMMIT";
    if($numreginser == 1)
    {
        vermensaje("Registro Borrado");
        echo "location.href = 'http://planifacad.unanleon.edu.ni/BorrarRegistroDepartamento.php'";
    }
}

$insertar = pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" . pg_last_error($conexion));
}

?>
//Nuevo Departamento
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Nuevo Registro Departamento</title>
<script language="javascript" type="text/javascript">
function volver()
{
    window.history.back();
}

function validar(fr)
{
    if(fr.nombredpto.value=="")
    {
        alert("Error-El campo Nombre Departamento no puede quedar vacio");
        fr.nombredpto.focus();
        return false;
    }
    else
    {
        if (fr.nombredpto.value.search(/([\D]+[\s]?)+/)==-1)
        {
            alert("Error-El Campo Nombre Departamento solo puede contener
letras no acentuadas");
        }
    }
}
```



```
        fr.nombredpto.select();
        fr.nombredpto.focus();
        return false;
    }
}

if (fr.nombredirector.value=="")
{
    alert("Error-El campo Nombre Director no puede quedar vacio");
    fr.nombredirector.focus();
    return false;
}
else
{
    if (fr.nombredirector.value.search(/([\D]+[\s]?)+/)==-1)
    {
        alert("Error-El Campo Nombre Director sólo puede contener letras");
        fr.anyofundacion.select();
        fr.anyofundacion.focus();
        return false;
    }
}
if (fr.extension.value=="")
{
    alert("Error-Número de Extensión no valido");
    fr.extension.focus();
    return false;
}
else
{
    if (fr.extension.value.search(/[\d]{4}/)==-1)
    {
        alert("Error-El campo Extensión debe contener 4 dígitos");
        fr.extension.select();
        fr.extension.focus();
        return false;
    }
}
return true;
}
</script>
```



```
<meta name="GENERATOR" content="Quanta Plus">
<meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">
</head>
<body>
```

```
<FORM name="fr" action="pprincipaldpto.php" method="POST" onsubmit="return
validar(this)">
<div id="masthead" Nuevo Registro: Departamento</div>
<HR>
```

```
<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">
```

```
<tr>
    <td>Nombre Departamento:</td>
    <td><INPUT type="text" name="nombredpto" size="36" maxlength="36"></td>
</tr>
```

```
<?
include_once("funcionesdepto.php");

$year= array();
for($n=1980;$n<=((int)date("Y"));$n++)
    array_push($year,$n);
$year=array_reverse($year);
cargaranyo("Año Fundación","anyofundacion",$year,$year[0]);
?>
```

```
<tr>
    <td>Nombre Director:</td>
    <td><INPUT type="text" name="nombredirector" size="50"
maxlength="60"></td>
</tr>
```

```
<tr>
    <td>Extensión Telefónica:</td>
    <td><INPUT type="text" name="extension" size="4" maxlength="4"></td>
</tr>
```



```
<tr>
  <td>Observaciones:</td>
  <td><INPUT type="text" name="observacion" size="50" maxlength="60"
value="Ninguna"></td>
</tr>

<tr>
  <td><INPUT type="submit" value="GUARDAR" ></td>
  <td><INPUT type="hidden" name="npageD" value="1"></td>
</tr>

</tbody>
</table>

</FORM>
<p align="center"

</p>
</body>
</html>
```

```
//Buscar Departamento
<?php
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Buscar Registro Departamento</title>
<meta name="GENERATOR" content="Quanta Plus">
<meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

<script language="javascript" type="text/javascript">
function volver()
{
    window.history.back();
}
</script>
```



```
</head>
<body>
<FORM action="http://planifacad.unanleon.edu.ni/pprincipaldpto.php" method="POST">

<div id="masthead" Buscar Registro: Departamento</div>
<HR>

<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">

<?php
include_once("configuracion.php");
include_once("funcionesdepto.php");

global $cadenaconexcion;
$conexion = pg_connect($cadenaconexcion);
if($conexion == FALSE)
{
    vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
    exit();
}

$sentsql = "select nombredpto from departamento order by nombredpto";

$ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
pg_last_error($conexion));
$numregistros = pg_num_rows($ResultadoBusqueda);
?>

<tr>
    <td>Departamento:</td>
    <td>

<?php
if($ResultadoBusqueda)
{
    echo "<select name = 'nombredpto'>";
```



```
while($arr=pg_fetch_array($ResultadoBusqueda))
{
    //value = Valor que va a tomar la variable como resultado de seleccionar
    //la opcion que ella presenta.
    //el if sirve para añadir por defecto la opcion departameno de computacion
    if($arr[nombredpto] == "Biología")
    {
        echo          "<option          value=\"\$arr[nombredpto]\"
selected>$arr[nombredpto]</option>";
    }
    else
    {
        echo "<option value=\"\$arr[nombredpto]\">$arr[nombredpto]</option>";
    }
}
echo "</select>";
}

else
{
    echo "NO existen registros en la tabla Departamento";
    exit();
}
?>
</td>
</tr>

<tr>
    <td><INPUT type="hidden" name="npageD" value="4"></td>
    <td><INPUT type="submit" value="BUSCAR"></td>
</tr>

</tbody>
</table>
</FORM>

<p align="center"

</p>
</body>
</html>
```



```
//Borrar Departamento
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Borrar Registro Departamento</title>
  <meta name="GENERATOR" content="Quanta Plus">
  <meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <link
    rel="stylesheet"
    href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

  <script language="javascript" type="text/javascript">
  function volver()
  {
    window.history.back();
  }
</script>

</head>
<body>
<FORM action="pprincipaldepto.php" method="POST">

<div id="masthead" Borrar Registro: Departamento</div>
<HR>

<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">

<?php
include_once("configuracion.php");
include_once("funcionesdepto.php");

global $cadenaconexcion;
$conexion = pg_connect($cadenaconexcion);
if($conexion == FALSE)
{
  vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
  exit();
}
```



```
$sentsql = "select nombredpto from departamento order by nombredpto";

$ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
pg_last_error($conexion));
$numregistros = pg_num_rows($ResultadoBusqueda);
?>

<tr>
    <td>Departamento:</td>
    <td>
<?php
if($ResultadoBusqueda)
{
    echo "<select name = 'nombredpto'>";
    while($arr=pg_fetch_array($ResultadoBusqueda))
    {
        //value = Valor que va a tomar la variable como resultado de seleccionar
        //la opcion que ella presenta.
        //el if sirve para añadir por defecto la opcion departameno de computacion
        if($arr[nombredpto] == "Programacion I")
        {
            echo          "<option          value=\"\$arr[nombredpto]\"
selected>$arr[nombredpto]</option>";
        }
        else
        {
            echo "<option value=\"\$arr[nombredpto]\">$arr[nombredpto]</option>";
        }
    }
    echo "</select>";
}
else
{
    echo "NO existen registros en la tabla Departamento";
    exit();
}
?>
</td>
</tr>

<tr>
    <td><INPUT type="hidden" name="npageD" value="2"></td>
    <td><INPUT type="submit" value="BORRAR"></td>
</tr>
```



```
</tbody>
</table>
</FORM>

<p align="center">

</p>
</body>
</html>
```

```
//pprincipaldpto.php
<?php
include_once("funcionesdepto.php");
include_once("configuracion.php");

global $npageD;

switch($npageD)
{
    case 1:
        InsertarNuevoDepartamento();
        break;
    case 2:
        VerRegistroBorrar();
        break;
    case 3:
        BorrarRegistroDepartamento();
        break;
    case 4:
        BuscarDepartamento();
        break;
    case 5:
        VerRegistroEditar();
        break;
    case 6:
        GuardarRegistroEditar();
        break;
    default:
        vermensaje("ERROR-Acceso denegado");
        echo "<script>location.href
'http://planifacad.unanleon.edu.ni/index.php'</script>";
        break;
}

?>
```



```
//ReporteDepartamento.php
<?
include_once("configuracion.php");

global $cadenaconexcion;
$conexion = pg_connect($cadenaconexcion);

if($conexion == FALSE)
{
    vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
    exit();
}

$sentsql = "select * from departamento order by nombredpto";
$ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
pg_last_error($conexion));
$numregistros = pg_num_rows($ResultadoBusqueda);

if($numregistros)//si existen registros la variable $numregistros!=0
{
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Departamentos de la Facultad de Ciencias</title>
<meta name="GENERATOR" content="Quanta Plus">
<meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link
            rel="stylesheet"
            type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

<script language="javascript" type="text/javascript">
    function volver()
    {
        window.history.back();
    }
</script>

</head>
<body>
<FORM method="POST">
```




```
//paginapricipal.php
<?php
//session_start();
?>
<html>
<head>
<title>PLANIFICACIÓN ACADÉMICA DE LA FACULTAD DE CIENCIAS</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<div align="center"><b><h1>Planificación Académica de la Facultad de
Ciencias</h1></div></td>
<link rel="stylesheet" type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

</head>
<body background="http://planifacad.unanleon.edu.ni/Imagenes/paraninfo.jpg"
bgcolor="FFCECB">

<!-- ***** This is the section of code you need to paste into your web pages ***** -->
<script type="text/javascript" src="/Menus/milonic_src.js"></script>
<noscript><a href="http://www.milonic.com/">JavaScript Menus and DHTML Menus
Powered by Milonic</a></noscript>
<script type="text/javascript">
if(ns4)_d.write("<scr"+"ipt type=text/javascript src=/Menus/mmenuns4.js><\scr"+"ipt>");

else _d.write("<scr"+"ipt type=text/javascript src=/Menus/mmenudom.js><\scr"+"ipt>");
</script>
<script type="text/javascript" src="/Menus/menu_data.js"></script>
<!-- ***** JavaScript Menu HTML Code -->

<BR><BR>

<table style="border:0px solid #ffffff" color="#ffffff">

<tr>
<td class=form1>
<a class="link" href="http://www.milonic.com/menusample23.php">Menu
Positioning With Offsets</a> </td>
</tr>
</table>
</body>
</html>
```



```
//index.php
<?php
include_once("funcionespprincipal.php");
include_once("configuracion.php");

//variables enlazadas a los controles del los formularios html
global $npagP;

switch($npagP)
{
    case 0:
        pagina_inicial();
        break;
    case 1:
        validarcontraseña();
        break;
    default:
        vermensaje("ERROR DESCONOCIDO");
        exit();
}
?>

//funcionesprincipal.php
<?php

function pagina_inicial()
{
    //IniciarSesionyDeclaracion();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Planificación Académica de la Facultad de Ciencias</title>
<meta name="GENERATOR" content="Quanta Plus">
<meta name="AUTHOR" content="Francisco Castro - Karina Esquivel">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link
            rel="stylesheet"
            type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

</head>
<body
            background="http://planifacad.unanleon.edu.ni/Imagenes/paraninfo.jpg"
bgcolor="FFCECB">
```



```
<table cellSpacing=0 cellPadding=0 border=0 style="border:0px solid #aaaaaa"
align="center">
```

```
<tr>
```

```
    <td class=backG align=center>
    <div align="center"><b><h1>Planificación Académica de la Facultad de
Ciencias</h1></div></td>
```

```
</tr>
```

```
<tr>
```

```
    <td class=form1 align="center">
    <b><i>UNAN León</i></b> Trabajo Final para optar al título de Ingeniero
en Sistemas de Información<br>Realizado por:
    <b><i>Francisco Castro & Karina Esquivel</i></b>
    </td>
```

```
</tr>
```

```
</table>
```

```
<br><br><br>
```

```
<FORM action="index.php" method="POST" id="formautenticacion" title="Formulario de
contraseña">
```

```
<TABLE width="2" cellspacing="2" border="2" cellpadding="2" align="center"
bgcolor="#e5ddff" frame="border" id="tablaautenticacion">
```

```
<tr>
```

```
    <td>Usuario: </td>
    <td><INPUT type="text" name="usuarioplanif" size="25" maxlength="25"
align="left"></td>
```

```
</tr>
```

```
<tr>
```

```
    <td>Contraseña: </td>
    <td><INPUT type="password" name="passwordusuarioplanif" size="25"
maxlength="25" align="left"></td>
```

```
</tr>
```

```
<tr>
```

```
    <td><INPUT type="submit" name="aceptar" align="middle" value="ACEPTAR"></td>
    <td><INPUT type="hidden" name="npageP" value="1"></td>
```

```
</tr>
```

```
</tbody>
```

```
</TABLE>
```

```
</FORM>
```



```
</body>
</html>
<?php
}

function vermensaje($mensaje)
{
    echo "<script type 'text/javascript'>
    alert ('$mensaje');
    </script> ";
}

function pagina_principal()
{
    echo "<script>location.href = 'http://planifacad.unanleon.edu.ni/paginaprincipal.php'
    </script>";
}

function pagina_ppal_usuarioinvitado()
{
    echo "<script>location.href
'http://planifacad.unanleon.edu.ni/paginaprincipalusuariosinv.php' </script>";
}

function pagina_ppal_usuariodepartamento()
{
    echo "<script>location.href
'http://planifacad.unanleon.edu.ni/paginaprincipalusuariosdpto.php' </script>";
}

function validarcontraseña()
{
    global $cadenaconexcion,$passwordusuarioplanif,$usuarioplanif;
    global $usuarioingresa;

    $conexion = pg_connect($cadenaconexcion);
    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql = "select tipousuario from usuarios where nombreusuario = '$usuarioplanif' and
    password = '$passwordusuarioplanif'";
```




```
<link rel="stylesheet" type="text/css"
href="http://planifacad.unanleon.edu.ni/estilo/estilodepagina.css">

<script language="javascript" type="text/javascript">
function volver()
{
    window.history.back();
}
</script>

</head>
<body>
<FORM action="pprincipaldpto.php" method="POST" onsubmit="return validar(this)">

<div id="masthead" Editar Registro: Departamento</div>
<HR>

<table align="center" bgcolor="#eae9e8" frame="border" id="tabladpto">
<tbody bgcolor="#eae9e8">

<?php
include_once("configuracion.php");
include_once("funcionesdepto.php");

    global $cadenaconexcion;
    $conexion = pg_connect($cadenaconexcion);
    if($conexion == FALSE)
    {
        vermensaje("ERROR AL CONECTAR LA BASE DE DATOS");
        exit();
    }

    $sentsql = "select nombredpto from departamento order by nombredpto";

    $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en la consulta" .
    pg_last_error($conexion));
    $numregistros = pg_num_rows($ResultadoBusqueda);
?>
    <tr>
    <td>Departamento:</td>
    <td>

<?php
```



```
if($ResultadoBusqueda)
{
    echo "<select name = 'nombredpto'>";
    while($arr=pg_fetch_array($ResultadoBusqueda))
    {
        //value = Valor que va a tomar la variable como resultado de seleccionar
        //la opcion que ella presenta.
        //el if sirve para añadir por defecto la opcion departameno de computacion
        if($arr[nombredpto] == "Programacion I")
        {
            echo "                <option                value=\"\$arr[nombredpto]\"
                selected>\$arr[nombredpto]</option>";
        }
        else
        {
            echo "<option value=\"\$arr[nombredpto]\">\$arr[nombredpto]</option>";
        }
    }
    echo "</select>";
}
else
{
    echo "NO existen registros en la tabla Departamento";
    exit();
}
?>

</td>
</tr>

<tr>
    <td><INPUT type="hidden" name="npageD" value="5"></td>
    <td><INPUT type="submit" value="BUSCAR"></td>
</tr>

</tbody>
</table>
</FORM>
<p align="center">

</p>
</body>
</html>
```

