PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

TEMA:

"DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR PARA EL MODELO ACADÉMICO "NAYÓN" DE LA PONTIFICA UNIVERSIDAD CATÓLICA DEL ECUADOR"

AUTOR:

JUAN MANUEL CAMPOVERDE ÁVILA

FRANKLIN SEBASTIÁN OVIEDO MONTENEGRO

DIRECTOR/A:

MBA. FRANCISCO RODRÍGUEZ CLAVIJO

QUITO – ECUADOR, 2013

DEDICATORIA

A Dios, por permitirnos llegar a culminar esta etapa de nuestra vida tan especial superando los momentos difíciles los cuales nos enseñaron a valorarlo cada día más.

A nuestros padres por ser las personas que estuvieron presentes con su apoyo incondicional durante todo nuestro trayecto estudiantil.

A nuestros profesores, por su motivación así como su sabiduría transmitida lo cual nos permite desempeñarnos con éxito en nuestras responsabilidades profesionales.

AGRADECIMIENTOS

Agradecemos a Dios por darnos la salud y la fuerza para poder superar todos los obstáculos presentados durante la realización de este proyecto.

A nuestros padres, que con sus consejos y motivación nos inspiraron el deseo de superación personal constante.

Al Ing. Francisco Rodríguez Clavijo, director de nuestra disertación la cual se llegó a realizar cumpliendo nuestras expectativas con su valiosa guía y asesoramiento.

ÍNDICE GENERAL

INTRODUCCIÓN ---- 8
CAPÍTULO I ---- 9
1.1. NUEVO MODELO ACADÉMICO "NAYÓN" ---- 9
1.1.1. PRINCIPIOS ---- 10
1.1.1.1. LA SITUACIÓN ACTUAL ---- 11
1.1.2. ESTRUCTURA ACADÉMICA ---- 13
1.1.2.1. EDUCACIÓN BASADA EN COMPETENCIAS ---- 14
1.1.2.2. ORGANIZACIÓN DEPARTAMENTAL ---- 15
1.1.2.2.1. ORGANIZACIÓN UNIVERSITARIA POR ESCUELAS O FACULTADES ---- 15
1.1.2.2.1.1. CARACTERÍSTICAS ---- 15
1.1.2.2.2. ORGANIZACIÓN UNIVERSITARIA POR DEPARTAMENTOS ---- 16
1.1.2.2.2.1. CARACTERÍSTICAS ---- 16
1.1.2.2.2.2. PROPUESTA DE DIRECCIONES DIVISIONALES ---- 17
1.1.2.2.2.2.1. DIVISIÓN DE CIENCIAS Y HUMANIDADES ---- 17
1.1.2.2.2.2.3. DIVISIÓN DE ESTUDIOS PROFESIONALES ---- 18
1.1.2.2.2.3. VENTAJAS DE LA DEPARTAMENTALIZACIÓN ---- 20
1.1.2.2.2.3.3. EN EL ASPECTO ACADÉMICO ---- 20
1.1.2.2.2.3.4. EN EL ASPECTO ADMINISTRATIVO ---- 21
1.1.2.2.2.3.5. EN EL ASPECTO ECONÓMICO ---- 21
1.1.2.2.2.4. OBJETIVOS Y METAS DE LA DEPARTAMENTALIZACIÓN ---- 21
1.1.3. FORMULACIÓN MATEMÁTICA ---- 22
1.1.3.2. EL MODELO MATEMÁTICO ---- 22
1.1.3.3. VARIABLES DE DECISIÓN (OPTIMIZACIÓN) ---- 23
1.1.3.4. FUNCIÓN OBJETIVO ---- 24
CAPITULO II ---- 28
2.1. SISTEMAS, MODELOS Y SIMULACIÓN ---- 28
2.1.1. SISTEMA ---- 29
2.1.2. MODELO ---- 30
2.1.3. SIMULACIÓN ---- 31
2.1.4. VENTAJAS DE LA SIMULACIÓN ---- 32
2.1.5. DESVENTAJAS DE LA SIMULACIÓN ---- 33
2.2. HERRAMIENTA PARA ALMACENAR LOS DATOS. ---- 33

2.2.1. BASE DE DATOS ---- 34
2.2.1.1. VENTAJAS DE LAS BASES DE DATOS ---- 35
2.2.1.2. ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS ---- 35
2.2.1.3. DISEÑO DE UNA BASE DE DATOS ---- 36
2.2.1.4. SISTEMA DE GESTIÓN DE BASE DE DATOS ---- 38
2.2.1.5. MANEJADOR DE BASES DE DATOS MYSQL ---- 38
2.3. HERRAMIENTAS PARA DESARROLLO DE SISTEMAS INFORMÁTICOS ---- 39
2.3.1. LENGUAJES DE PROGRAMACIÓN ---- 39
2.3.1.1. PHP ---- 42
2.3.1.2. JAVASCRIPT ---- 43
2.3.1.3. FRAMEWORK ---- 44
2.4. LENGUAJE UNIFICADO DE MODELADO ---- 44
2.4.1. RELACIONES ---- 45
2.4.2. DIAGRAMAS ---- 47
2.4.2.1. DIAGRAMAS DE CASOS DE USO ---- 47
2.4.2.2. DIAGRAMAS DE CLASES ---- 48
2.5. METODOLOGÍA DE DESARROLLO DE SOFTWARE RAD ---- 48
2.5.1. ETAPAS DEL CICLO RAD ---- 49
2.5.1.1. ETAPA DE PLANIFICACIÓN DE LOS REQUISITOS ---- 49
2.5.1.2. ETAPA DE ANÁLISIS Y DISEÑO ---- 49
2.5.1.3. CONSTRUCCIÓN ---- 49
2.5.1.4. IMPLEMENTACIÓN ---- 50
CAPITULO III ---- 51
3.1. MODELO DE DESARROLLO DEL SITIO WEB ---- 51
3.2. FASE PLANIFICACIÓN DE LOS REQUISITOS ---- 51
3.3. FASE DE ANÁLISIS Y DISEÑO ---- 52
3.3.1. IDENTIFICAR CASOS DE USOS Y ESQUEMATIZAR DIAGRAMAS DE CASOS DE USO. ---- 52
3.3.2. NARRACIÓN DE LOS CASOS DE USOS ENCONTRADOS ---- 54
4.3.3. DISEÑO DE DATOS ---- 80
4.3.3.1. DIAGRAMA ENTIDAD RELACIÓN ---- 80
3.3.3.2. DISEÑO DE LA BASE DE DATOS ---- 81
3.3.3.3. DISEÑO DEL MAPA DE NAVEGACIÓN. ---- 89
3.3.4. DISEÑO DE LA INTERFAZ HOMBRE-MÁQUINA ---- 90

3.4. FASE DE CONSTRUCCIÓN ---- 90
3.5. PRUEBAS Y RESULTADOS ---- 90
3.5.1. PRUEBAS DE EJECUCIÓN ---- 90
3.5.2. HARDWARE UTILIZADO PARA PRUEBAS ---- 91
3.5.2.1. CARACTERÍSTICAS DE COMPUTADORES PARA PRUEBAS ---- 91
3.5.2.2. CARACTERÍSTICAS DE SERVIDOR PARA PRUEBAS ---- 91
CAPÍTULO IV ---- 92
4.1. PROCESO DE LÓGICA UTILIZADA ---- 92
4.2. EJECUCIÓN DEL PROGRAMA ---- 94
4.2.1. ADMINISTRACIÓN DE ALUMNOS ---- 95
4.2.1.1. VER ALUMNO ---- 95
4.2.1.2. AGREGAR ALUMNOS ---- 95
4.2.2. AVANZAR SEMESTRE ---- 96
4.2.3. ADMINISTRACIÓN DE CARRERAS ---- 98
4.2.3.1. NUEVA CARRERA ---- 99
4.2.3.2. ACTUALIZACIÓN DE CARRERAS ---- 99
4.2.3.3. ELIMINACIÓN DE CARRERAS ---- 100
4.2.3.4. ALUMNOS EN CARRERAS ---- 101
4.2.3.5. PROFESORES EN CARRERAS ---- 102
4.2.4. ADMINISTRACIÓN DE DEPARTAMENTOS ---- 102
4.2.4.1. CARRERAS POR DEPARTAMENTOS ---- 103
4.2.4.2. ASIGNATURAS – CARRERA -DEPARTAMENTO ---- 104
4.2.4.3. ALUMNOS FORÁNEOS DE DEPARTAMENTOS ---- 105
4.2.4.4. PROFESORES EN DEPARTAMENTOS ---- 105
4.2.5. ADMINISTRACIÓN DE AULAS ---- 106
4.2.6. ADMINISTRACIÓN DE HORARIOS ---- 107
4.2.6.1. NUEVO HORARIO ---- 108
4.2.7. ADMINISTRACIÓN DE PROFESORES ---- 108
4.2.8. ADMINISTRACIÓN DE NIVELES ---- 109
4.2.9. PROCESO DE SIMULACIÓN ---- 110
4.2.10. PROCESO DE INGRESO DE DATOS ---- 111
4.2.11. HISTORIAL ACADÉMICO ---- 111
4.2.11.1. BÚSQUEDA EN HISTORIAL ACADÉMICO ---- 112

4.2.12. HISTORIAL DEPARTAMENTO---------- 130
5.2.1.3. ADMINISTRACIÓN DE MATERIAS OPTATIVAS ---------- 133
5.2.1.4. HISTORIAL AULA ---------- 134
5.2.15. HISTORIAL PROFESORES ---------- 136
4.3. RESULTADOS ESTADÍSTICOS ---------- 139
4.3.1. VARIABLES INDEPENDIENTES---------- 139
4.3.1.1. DIMENSIÓN UTILIZACIÓN ---------- 139
4.3.1.2. DIMENSIÓN ESCALABILIDAD ---------- 139
4.3.2. VARIABLES DEPENDIENTES---------- 140
4.3.2.1. MEDICIÓN PREVIA ANTES DE LA EJECUCIÓN DE LA APLICACIÓN WEB ---------- 140
5. CONCLUSIONES---------- 140
6. RECOMENDACIONES ---------- 142
7. REFERENCIAS---------- 143
7.1. BIBLIOGRAFÍA:---------- 143
7.2. INTERNET: ---------- 143
8. ANEXOS---------- 144

INTRODUCCIÓN

Hoy en día la presencia de sitios Web tanto institucionales como personales es cada vez más usual e incluso para la gran mayoría de empresas y entidades que suministran productos y servicios es una necesidad de primer nivel, ya sea para evitar ser desplazados por su competencia, o con el ánimo de ofrecer mejores y oportunos servicios a sus clientes.

A nivel educativo las Instituciones Públicas y Privadas, especialmente las Universidades, ya cuentan con sus Sitios Web para dar a conocer sus acontecimientos y procesos académicos, es por ello la propuesta para desarrollar el sitio web para la simulación del modelo académico bajo el cual la Pontificia Universidad Católica del Ecuador proyecta desempeñar sus funciones, en su sede de Nayón. Debido a la falta de interés por renovar un esquema universitario adoptado regido por escuelas o facultades, con el trascurso del tiempo y las exigencias que demanda enfrentar de manera eficiente los retos que presenta el siglo XXI a las Instituciones de Educación Superior en el mundo en general y a aquellas que se encuentran bajo la responsabilidad directiva de la Compañía de Jesús, se ha hecho evidente la poca idoneidad estructural de su modelo para estar al día y enfrentar con éxito los desafíos que el presente y el futuro previsible ponen delante.

La falta de interés antes mencionada resulta una dificultad para manejar información clasificada, actualizada y ordenada que indique los comportamientos y tendencias que acogerían todos los elementos participantes que conforman la universidad durante el tiempo en el cual se lleva a cabo la obtención de un título académico. Es por esto que se necesita sistematizar el proceso que genere una simulación del nuevo modelo educativo que se quiere implementar en la institución educativa.

El sitio web permite, conectarse desde computadoras conectadas a una red dentro de los predios del laboratorio de informática de la facultad de Ingeniería de la Universidad para realizar todo el proceso de simulación antes mencionado. Así mismo facilitaría la toma de decisiones por parte de las autoridades universitarias para determinar una óptima planificación de organización y asignación de recursos en la sede de Nayón.

CAPÍTULO I

El presente capítulo hace referencia a la información de la reforma de la estructura y modelo académico que proyecta ejercer la Pontificia Universidad Católica del Ecuador en su sede de Nayón, es decir, su contexto organizacional: principios, fundamentos, características, ventajas, entre otros.

1.1. NUEVO MODELO ACADÉMICO "NAYÓN"

La Pontificia Universidad Católica del Ecuador en la actualidad ejerce su práctica de enseñanza bajo un esquema universitario regido por facultades y escuelas, lo cual asigna todas las responsabilidades de educación hacia un estudiante por parte únicamente de dicha escuela o facultad, es decir se imparten cursos que inclusive pertenecen a disciplinas ajenas a las mismas lo cual muchas veces conlleva a que profesores no especialistas sean los delegados para enseñar determinadas materias; fomentando así inclusive el aislamiento por parte de los estudiantes de distintas carreras, ya que los mismos entran en la escuela, pasan por ésta y salen de ella sin haberse mezclado en los cursos con estudiantes de otras escuelas, todo esto muestra la poca idoneidad estructural del modelo educativo actual de la universidad lo cual no permite enfrentar con éxito los desafíos que el presente exige académicamente.

La universidad, ha decido llevar a cabo una re-ubicación, hacia los terrenos que han sido ya adquiridos en Nayón, por dicho motivo, ha resuelto a su vez, reformar completamente su estructura y modelo académico.

Dado que la cantidad de materias que actualmente maneja la PUCE son excesivas, se vuelve innecesario ofrecer distintos planes de estudio a los cuales los conforman asignaturas comunes entre un determinada carrera y otra, por lo cual es innovador el hecho de unificarlas, y ofrecerlas a manera ya no de facultad, sino de unidad académica, idea central de la reforma deseada.

Con esto, cualquier tipo de estudiante podrá tomar la asignatura en la unidad académica que en el momento este ofertando la misma, para así de acuerdo a su tiempo y capacidad, disminuir tiempo y recursos en el avance de su carrera específica, lo cual promueve una imagen diferente, es decir el departamento ofrece todos los cursos pertenecientes a una misma disciplina dictados por verdaderos especialistas,

evitando que se repitan innecesariamente cursos de materias de iguales características, en toda la institución.

1.1.1. PRINCIPIOS[1]

La Pontificia Universidad Católica del Ecuador ha decidido llevar a cabo una ampliación y reubicación de su sede actualmente establecida en el centro de la ciudad de Quito, hacia los terrenos adquiridos en Nayón; situación la cual ha motivado a que ésta prestigiosa institución, fundada el año de 1946, a su vez efectúe una reforma completa en su estructura y modelo académico, con el objetivo de enfrentar con éxito los retos que presenta el siglo XXI a las Instituciones de Educación Superior en el mundo en general y a aquellas que se encuentran bajo la responsabilidad directiva de la Compañía de Jesús.

Muchos han sido los cambios sociopolíticos que han surgido a través de la historia que provocaron la conformación de modelos educativos, que sin alterar la finalidad que tienen las instituciones de enseñanza de nivel superior, dieron lugar a finalidades académicas, económicas y políticas diversas.

Pese a todo esto, las universidades latinoamericanas han tratado de actualizarse y constituir un fundamento sólido. No obstante, el presente siglo deja ver la poca idoneidad estructural de su modelo para estar al día y enfrentar con éxito los desafíos que el presente y el futuro previsible ponen delante. Las estructuras heredadas desde sus orígenes, por su pesadez administrativa y financiera y su complejidad burocrática de gobierno, se ha convertido en obstáculos para lo que hoy las sociedades esperan en el cumplimiento de su misma misión.

La PUCE[2] al ser una institución latinoamericana que constantemente busca su prosperidad considera prudente el hecho de adoptar un nuevo modelo educativo partiendo de un proceso inicial que lo simule, de manera tal que conforme avance la simulación se pueda ir visualizando indicadores que muestren cantidades de aulas y profesores necesarios para cubrir la demanda estudiantil, porcentajes de aprobación, reprobación y número de créditos que el estudiante cumple hasta finalizar una carrera académica, así como también el tiempo en el cual lo hace, indicadores los cuales

[1] **Extraído de:** http://cimm.ucr.ac.cr/aruiz/libros/Universidad%20y%20Sociedad/Capitulos/modelo.html
[2] **PUCE:** Pontificia Universidad Católica del Ecuador

permitan realizar una comparación con el modelo actual bajo el cual el establecimiento se encuentra ejerciendo sus funciones, de tal manera que una vez finalizada la simulación se tenga claro el nivel de éxito que la universidad pudiese conseguir con el cambio hacia el cual quiere apuntar para enfrentar los retos que se exigen hoy en día y se prevé en un futuro.

A continuación se detalla de manera muy esquemática los rasgos más importantes que al parecer caracterizan las necesidades a las que se enfrentan en la actualidad nuestra educación superior.

1.1.1.1. LA SITUACIÓN ACTUAL

Necesidad creciente de homologación de las IES[3] para responder a:

- Las demandas de la sociedad que requiere de científicos y profesionistas mejor preparados para soportar y hacer avanzar el conocimiento y el desarrollo tecnológico hacía el futuro.
- El aumento en la complejidad de un mundo cada vez más interdependiente con interacciones cada vez más rápidas.
- El incremento en los intercambios de académicos y estudiantes de las IES por todo el planeta.
- La posibilidad de realizar estudios y obtener grados a distancia.
- La acreditación de procesos formativos y diseño o selección de indicadores estandarizados de Evaluación Institucional.
- La realización de proyectos de investigaciones interdisciplinares, interinstitucionales e internacionales.

Necesidad de flexibilidad, rapidez en la respuesta a las siguientes tensiones dialécticas:

Tabla I-1: Tensiones Dialécticas.

o Individuo	— Sociedad
o Teoría	— Práctica
o Formación	— Información
o Educación	— Instrucción

[3] **IES:** Instituciones de Educación Superior

o Universalidad	– Especificidad
o Sustantividad	– Adjetividad
o Cualitativo	– Cuantitativo
o Necesidades sociales	– Práctica profesional

Autor: Campoverde, J. Oviedo, S. 2011

Si bien estas tensiones han estado presentes desde el origen de la institución universitaria, el grado de polarización ahora se ha hecho más intenso sobre todo en los polos de la columna de la derecha, el impacto de la actividad universitaria se concentraba más en la formación de los individuos que en la sociedad misma. Eran pocos los que podían acceder a estos estudios y su relevancia en la transformación de la comunidad en la que se ubicaban, prácticamente marginal. La ciencia clásica se preocupaba más por describir y comprender la realidad que por transformarla intencionadamente. El énfasis formativo predominaba sobre el acervo de información disponible. El desarrollo tecnológico era escaso y poco accesible para las mayorías. Predominaban los estudios teórico-racionales sobre las ciencias positivas empíricas y las cuestiones de carácter universal sobre lo específico. Había poco desarrollo de la división tanto del trabajo como del conocimiento. En un principio se buscó atender más inmediatamente a las necesidades sociales básicas que a los modelos racionalizados de profesión como no fuera la profesión religiosa, médica o jurídica.

A partir de los grandes movimientos sociales, económicos y políticos del siglo XVIII y de manera más significativa en la segunda mitad del siglo siguiente las universidades, varias de ellas re-estructuradas, tuvieron finalidades más típicamente determinadas por el estado, que en el caso de las europeas y algunas coloniales las sostenía financieramente. Durante el siglo XX el número y diversidad de instituciones de educación superior se incrementó de manera explosiva debido al desarrollo económico de los países y sobre todo por la expansión de la demanda educativa generada por el incremento demográfico de la población demandante, la división progresiva del trabajo y la complejidad institucional y profesional que el progreso requería.

Actualmente, son los modelos de organización universitaria por Facultades y Escuelas los que prevalecen y comienza un viraje en Europa para buscar las ventajas que el modelo americano por Departamentos que ha logrado mostrar su mayor adaptabilidad y capacidad de respuesta a los retos. Las IES buscan implantar en sus respectivas instituciones formas de organización estructural y curricular análogas a los descritos para el sistema departamental.

Se han dado tres versiones de estos modelos: el típico departamental, el que agrupa departamentos por objeto de conocimiento, y el matricial donde ciertos departamentos son los proveedores de servicio departamental y otros departamentos son receptores de estos servicios.

En lo que respecta al presente proyecto lo que se busca simular es el modelo americano, es decir en lugar de que la universidad esté organizada por facultades cuyas carreras tendrán varias materias en común, como por ejemplo una carrera técnica en computadoras recibe las mismas materias de ciencias exactas que recibiría una carrera técnica civil, la universidad adoptará una organización basada en departamentos especializados en una sola rama que imparta clases a toda la universidad.

Se considera que esta propuesta busca mejorar en la medida en que los departamentos se agrupan por su finalidad sobre el avance y difusión de las ramas del conocimiento acerca de los distintos aspectos de la realidad, y por las formas de intervención multidisciplinar o interdisciplinar, para la transformación de dicha realidad racionalizadas como diversas formaciones profesionales. De suerte que, más que un sistema matricial, todos los departamentos, según sea el caso, funcionan como proveedores o receptores de conocimiento teórico o práctico proporcionado por expertos en la materia. Se busca configurar así una organización en Red que responda con la mayor flexibilidad y eficiencia posibles para adaptarse rápidamente a las necesidades sociales. Esta forma da por resultado un organismo dinámico más que una estructura fija.

1.1.2. ESTRUCTURA ACADÉMICA[4]

La PUCE en la actualidad ejerce su práctica de enseñanza bajo un esquema universitario regido por facultades y escuelas mientras que la tendencia se inclina hacia un modelo americano el cual se rige por Departamentos, los mismos que se agrupan por su finalidad sobre el avance y difusión de las ramas del conocimiento acerca de los distintos aspectos de la realidad, y por las formas de intervención

[4] **Extraído de:** Comisión Asesora del Rector, PUCE. 2011

multidisciplinar o interdisciplinar, para la transformación de dicha realidad racionalizadas como diversas formaciones profesionales.

1.1.2.1. EDUCACIÓN BASADA EN COMPETENCIAS

Incorporando los avances pedagógicos más modernos acorde con la visión humanista cristiana de la PUCE, se propone que el énfasis en sugestión educativa recaiga cada vez menos en la enseñanza magistral del profesor para aumentar progresivamente la gestión activa del estudiante. Desde esta perspectiva, el sujeto principal de la formación pasa del profesor al alumno, quien, además de ser el destino directo del conocimiento que está dispuesta a ofrecer la Universidad, en esta nueva concepción pedagógica asume simultáneamente la principal responsabilidad de su formación. Pero este cambio no rebaja la importancia del profesor; más bien la refuerza. En el nuevo esquema, pasan a tener mayor relevancia sus tareas de tutor e investigador.

El educando, que, por naturaleza, se encuentra en camino de formación, bajo el nuevo esquema requiere más que nunca que el nuevo guía esté cada vez más enterado y actualizado en los avances científicos que pretende entregar que día a día se generan en el mundo, tanto en materia de conocimiento puro como de conocimiento aplicado. En otras palabras, el profesor se convierte en un importante guía de las aspiraciones y requerimientos educativos del estudiante, en su afán de incrementar su acervo de sabiduría en forma permanente. Este cambio en el papel del estudiante y del profesor, y, sobretodo, en la conformación de los programas de estudio, provocará definitivamente la modificación en las relaciones y vinculaciones del estudiante con la estructura académica y administrativa de la universidad.

A su vez, esta propuesta implica la redefinición y reorganización de todos los sistemas y estructuras académicos y administrativos de la universidad. De acogerse esta iniciativa, el aumento y mejoramiento de las competencias del estudiante pasa a ser muy significativo y con ello se incrementan sustancialmente las posibilidades de también mejorar radicalmente la calidad de líderes que requiere el país.
Complemento y consecuencia de esta innovación pedagógica es la necesidad de mejorar la eficiencia y las competencias individuales y colectivas de la universidad.

1.1.2.2. ORGANIZACIÓN DEPARTAMENTAL

Por su naturaleza, la universidad es la institución que ha acumulado conocimientos en el transcurso de décadas, y por tanto se constituye en el origen de dicho conocimiento a ser trasladado a los estudiantes. Bajo el nuevo esquema, principalmente los Departamentos son el claustro de tutores investigadores, en los cuales se instala el acervo de conocimientos a ser ofertados a los estudiantes.

En el Modelo Educativo que se propone, quienes tienen a su cargo la trasmisión del conocimiento a los educandos deben incorporar y fortalecer su capacidad de investigación, de modo que el acervo de conocimientos antes aludido sea cada vez mayor.

1.1.2.2.1. ORGANIZACIÓN UNIVERSITARIA POR ESCUELAS O FACULTADES

Hace referencia a la estructura académica bajo la cual actualmente rige sus funciones la Pontificia Universidad Católica del Ecuador.

1.1.2.2.1.1. CARACTERÍSTICAS

- La escuela es la dependencia académica básica responsable de todas las condiciones necesarias para conseguir un título profesional, si se extiende hasta la obtención de maestrías y doctorados, se llama facultad.
- La escuela imparte todos los cursos para la obtención del grado académico, aunque pertenezcan a distintas disciplinas. Por ejemplo, la escuela de derecho ofrece sociología, economía, etc.
- Profesores no especialistas imparten los diferentes cursos.
- Los mismos cursos ofrecidos por distintas escuelas, sociología, estadística, economía, psicología, se multiplican innecesariamente en la universidad.
- Impide la formación de verdaderos colegios de profesores, porque éstos pertenecen a distintas disciplinas. Por ejemplo economistas, físicos, matemáticos, psicólogos, etc.
- Consecuentemente, entorpece la cooperación de los profesores de una misma disciplina al elaborar los programas de sus asignaturas y de exámenes iguales para las mismas materias.
- Dificulta la investigación que suele entonces refugiarse en institutos o centros.

- Desvincula la docencia de la investigación, objetivos fundamentales de la universidad moderna, porque resulta casi imposible realizar investigación en una escuela o facultad, formada por profesores de distintas disciplinas.
- Favorece el aislamiento de los estudiantes de las distintas carreras. Estos entran en la escuela, pasan por ésta, salen de ella sin haberse mezclado en los cursos con estudiantes de otras escuelas.
- En cambio, como es de esperar, la universidad organizada departamentalmente ofrece una imagen diferente.

1.1.2.2.2. ORGANIZACIÓN UNIVERSITARIA POR DEPARTAMENTOS

Hace referencia a la estructura académica que tiene proyectado ejercer la Pontificia Universidad Católica del Ecuador en su sede de Nayón.

1.1.2.2.2.1. CARACTERÍSTICAS

- El departamento es la unidad académica básica, formada por una comunidad de profesores e investigadores, relativamente autónoma y responsable de la docencia e investigación en un campo determinado del conocimiento, es decir, en una disciplina: economía, física, química, etc.
- El departamento ofrece todos los cursos pertenecientes a una misma disciplina.
- Permite la comunicación entre todos los profesores de una misma disciplina.
- Evita que se multipliquen innecesariamente los cursos de la misma materia en toda la institución.
- Facilita la investigación y la relaciona con la enseñanza.
- Fomenta la comunicación horizontal entre los estudiantes de diversas carreras o profesiones, y les permite asomarse a campos de conocimiento distinto del propio.
- Permite aprovechar mejor las instalaciones, tales como laboratorios y talleres que en otro tipo de organización tienden a multiplicarse innecesariamente.
- Encomienda la enseñanza de los diversos cursos a verdaderos especialistas.
- Enriquece la especialidad con nuevos cursos que de otra manera difícilmente se habrían ofrecido. La carrera de filosofía quizá nunca habría experimentado la necesidad de ofrecer un curso de filosofía de la técnica. Al convertirse en

departamento de filosofía y prestar servicio a otras carreras, organizará un curso con este contenido para administración de empresas.

- Permite aprovechar mejor a profesores excelentes que, asignados a un departamento, pueden enseñar a estudiantes de distintas carreras.
- Facilita la implantación del currículo flexible, aunque este no necesariamente va unido a la departamentalización.

1.1.2.2.2.2. PROPUESTA DE DIRECCIONES DIVISIONALES

La Pontificia Universidad Católica del Ecuador actualmente abarca una gran cantidad de ramas educativas las cuales al ser divididas en un primer grupo de acuerdo a sus campos de desarrollo científico o humanístico promovería la conservación y desarrollo del conocimiento que les es propio y en un segundo grupo de aplicación a la transformación de la naturaleza aplicando dichos conocimientos al desarrollo de métodos, procedimientos y tecnologías para la transformación de la realidad.

1.1.2.2.2.2.1. DIVISIÓN DE CIENCIAS Y HUMANIDADES

Esta División estaría compuesta por los departamentos cuya finalidad fundamental consiste en el aborde de los aspectos de la realidad que tradicionalmente obedecen a la segmentación del conocimiento y pretenden configurar y explicar los fenómenos que constituyen el universo material y humano, consecuentemente se desempeñan fundamentalmente en la investigación y la formación de académicos para el cultivo y avance en el desarrollo de las ciencias Físicas, Químicas, Biológicas, Matemáticas, Humanas y Sociales. Así como aportar la los departamentos de Estudios profesionales las materias fundamentales que mediante el servicio departamental requieran para la construcción de sus propios perfiles de egreso.

En el caso de la PUCE se reestructurarían bajo el modelo departamental las actuales:

Tabla I-2: División de Ciencias y Humanidades.

DIVISIÓN DE CIENCIAS Y HUMANIDADES
1. Departamento de Teología y Ciencias Humanas
- Sagrada Teología
- Filosofía
- Historia / Ciencias Historia
- Historia con mención en archivonomía y bibliotecología
2. Departamento de Ciencias Sociales
- Antropología con mención en arqueología
- Antropólogo con mención en antropología sociocultural
- Sociología con mención en desarrollo
- Sociología con mención en política
- Sociología con mención en relaciones internacionales
- Gestión Social
- Economía
3. Departamento de Ciencias Exactas y Naturales
- Bioanálisis (QFB)
- Bioquímica Clínica
- Histocitología
- Microbiología
- Microbiología Química
4. Departamento de Lingüística e Idiomas
- Lingüística aplicada con mención en enseñanza de lenguas
- Lingüística aplicada con mención en traducción
- Lingüística

Autor: Comisión Asesora del Rector, PUCE. 2011

1.1.2.2.2.2.3. DIVISIÓN DE ESTUDIOS PROFESIONALES

La función primordial de los departamentos de esta naturaleza es formar a los profesionales destinados a la conservación, transformación de la realidad material de nuestro universo material para, hacer posible la sobrevivencia y construir el desarrollo y progreso de las condiciones de vida de todos sus habitantes. La investigación propia de estos departamentos es la que usa el conocimiento para construir instrumentos, métodos, procesos y procedimientos de intervención sobre la realidad con objeto de modificarla intencionalmente.

El campo de ejercicio profesional de estos egresados así como parte de su formación debe estar lo más íntimamente vinculados a la problemática y ambiente natural de las formas de intervención sobre la dimensión de la realidad correspondiente.

En el caso concreto de la PUCE dicha división quedaría constituida por las siguientes unidades actuales en su versión departamental:

Tabla I-3: División de Estudios Profesionales.

DIVISIÓN DE ESTUDIOS PROFESIONALES	
1.	**Departamento de Administración y Negocios**
	- Ingeniería en Ecoturismo
	- Ingeniería en Gestión Hotelera
	- Licenciatura en Administración: Marketing / Productividad
	- Ingeniería en Contabilidad y Auditoría
	- Ingeniería Comercial
	- Licenciatura Multilingüe en Negocios y Relaciones Internacionales
2.	**Departamento de Educación**
	- Educación con mención en bachillerato
	- Educación con mención en bachillerato semipresencial
	- Educación con mención en educación inicial
	- Educación con mención en educación inicial (modalidad semipresencial)
	- Educación con mención en educación básica
	- Educación con mención en educación básica (modalidad semipresencial
	- Educación en música, lenguaje y movimiento (modalidad semipresencial)
	- Ciencias de la educación musical
3.	**Departamento de Psicología**
	- Psicología
	- Psicología clínica
	Psicología organizacional-
4.	**Departamento de Comunicación**[5]
	- Comunicación
	- Comunicación con mención en comunicación organizacional
	- Comunicación con mención en Periodismo para prensa radio y televisión
	- Comunicación con mención en comunicación y literatura

[5] Pueden unirse en: Departamento de Psicología, Educación y Comunicación.
- Psicología - Psicología Clínica - Educación
- Comunicación

5. Departamento de Ingenierías
- Ingeniería civil
- Ingeniería en sistemas y computación
- Ingeniería geográfica
- Ingeniería geográfica en gestión ambiental
- Ingeniero geógrafo en planificación territorial
- Geografía
6. Departamento de Arquitectura, Diseño y Artes
- Arquitectura
- Artes visuales
- Historia del arte
- Diseño de productos (Diseño Industrial)
- Diseño y comunicación visual (Diseño Gráfico)
7. Departamento de Medicina y Ciencias de la Salud
- Nutrición humana
- Terapia física
- Enfermería
- Medicina (Médico cirujano)
8. Departamento de Jurisprudencia
- Abogacía
- Ciencias Jurídicas
- TOTAL

Autor: Comisión Asesora del Rector, PUCE. 2011

1.1.2.2.2.3. VENTAJAS DE LA DEPARTAMENTALIZACIÓN

La propuesta de reforma de la estructura y modelo académico de la PUCE que se desea llevar a cabo en su sede de Nayón sugiere la idea de cambio y mejora del modelo educativo actual así como ventajas en distintos aspectos.

1.1.2.2.2.3.3. EN EL ASPECTO ACADÉMICO

- Elevar la calidad de los estudios con la reagrupación de las disciplinas y del claustro magisterial de acuerdo a la ciencia que cultivaba. Así se conseguía que los cursos estuviesen a cargo de especialistas.
- Evitar la multiplicidad de cursos con contenidos iguales y la multiplicidad de programas de estudios.
- Favorecer la investigación mediante la agrupación de especialistas en un mismo departamento y así relacionar aquélla con la docencia.

- Intensificar la cooperación interdisciplinaria entre profesores y estudiantes de diferentes profesiones y asegurar de esta guisa la comunicación horizontal inexistente en el sistema tradicional.
- Permitir la adopción del currículo flexible.
- Impedir que profesores de una especialidad impartiesen cursos de otra distinta.

1.1.2.2.2.3.4. EN EL ASPECTO ADMINISTRATIVO

- Evitar la multiplicidad de programas registrados con el mismo título.
- Crear un organismo centralizado que coordinase las actividades académico-administrativas e incrementarse el control de la documentación de la universidad.
- Hacer más efectivo el control adecuado.

1.1.2.2.2.3.5. EN EL ASPECTO ECONÓMICO

- Eliminaba la duplicación de las funciones económico-administrativas con el mejor aprovechamiento de los recursos tanto humanos como materiales.
- Evitaba que algunos profesores impartiesen cursos a grupos reducidos de estudiantes y así lograba también en este punto aprovechar al máximo los recursos humanos y materiales.

1.1.2.2.2.4. OBJETIVOS Y METAS DE LA DEPARTAMENTALIZACIÓN

- Ofrecer a los estudiantes la oportunidad de fomentar su currículo personal.
- Exigir al estudiante un esfuerzo benéfico para su propia maduración al obligarlo a adaptar a sus necesidades los cursos tomados fuera de su departamento.
- Favorecer la comunicación horizontal entre estudiantes de diversas carreras al darles a éstos la oportunidad de reunirse en diversos cursos con alumnos de otras disciplinas.
- Permitir la frecuente revisión de los planes de estudio y de los programas de cada curso, condición necesaria para plegarse al progreso acelerado de las ciencias que obliga a introducir los últimos adelantos y a descartar los elementos obsoletos.
- Fomentar la comunicación entre todos los profesores de las distintas asignaturas o cursos de una misma disciplina.

- Evitar la multiplicación innecesaria de los mismos cursos.
- Promover la investigación.

1.1.3. FORMULACIÓN MATEMÁTICA[6]

Un modelo matemático es el resultado de la abstracción de un sistema real, eliminando las complejidades y haciendo suposiciones pertinentes; se aplica una técnica matemática y se obtiene una representación simbólica del mismo.

1.1.3.2. EL MODELO MATEMÁTICO

Para un entendimiento mejor la modelización matemática para la optimización del servicio académico, se la ha planteado como una adaptación de un problema de transporte.

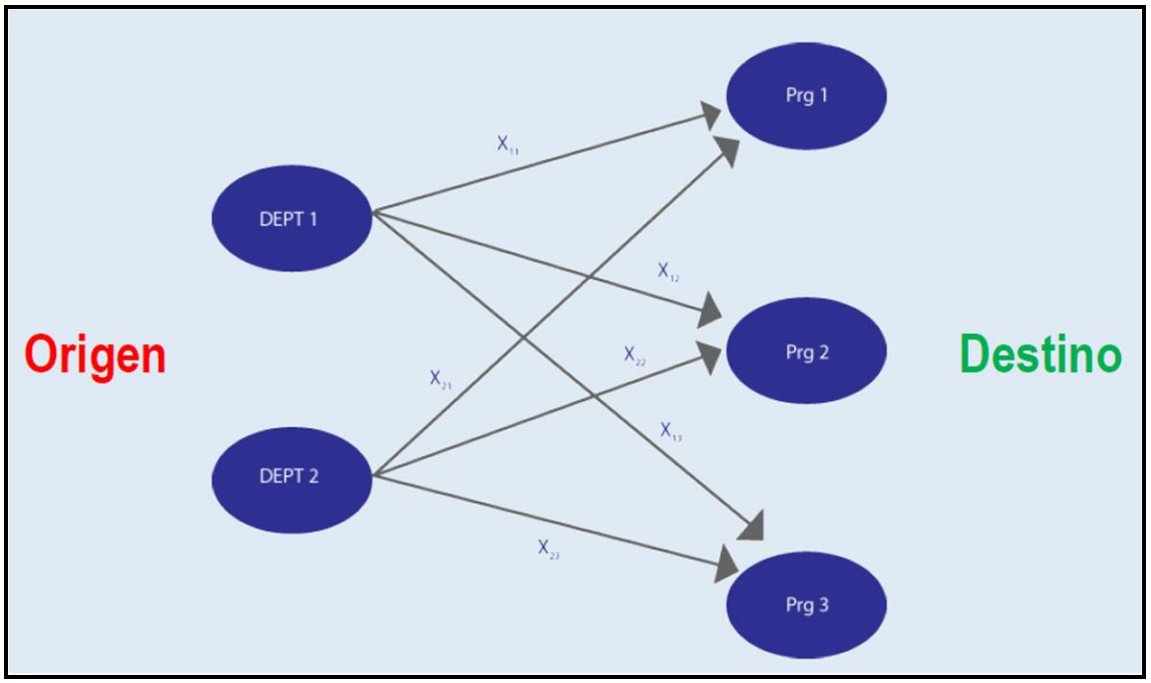
El problema consiste en decidir (optimizar) cuantos conocimientos (créditos) a transferir desde ciertos puntos de origen (departamentos académicos) a ciertos puntos de destino (programas para obtener una titulación) de modo de minimizar los costos de transferencia, dada la oferta y demanda de créditos en dichos puntos.

Se suponen conocidos los costos unitarios de transferencia, los requerimientos de demanda y la oferta disponible.

Este modelo de Optimización matemática consiste en una función objetivo y un conjunto de restricciones en la forma de un sistema de ecuaciones o inecuaciones, es decir la distribución de materias hacia los diferentes Departamentos, cabe mencionar que este modelo planteado es más eficiente ya que se usan todos los recursos disponibles evitando que se queden sin uso lo que pasa en el modelo actual, como por ejemplo, si no se completa el mínimo de estudiantes que son 10, no se abriría un curso, este modelo antiguo perjudicó y perjudica a muchos estudiantes, pero en el modelo que se plantea ya no pasaría lo mismo ya que al tener un tronco común de materias, siempre habría el mínimo requerido, cabe recalcar que los modelos de optimización son usados en casi todas las áreas de toma de decisiones, como en ingeniería de diseño y selección de carteras financieras de inversión , este modelo planteado se lo ha creado para estructurar una nueva estructura académica en la Universidad.

[6] **Extraído de:** Comisión Asesora del Rector, PUCE. 2011

Figura I-1: Modelo Matemático.

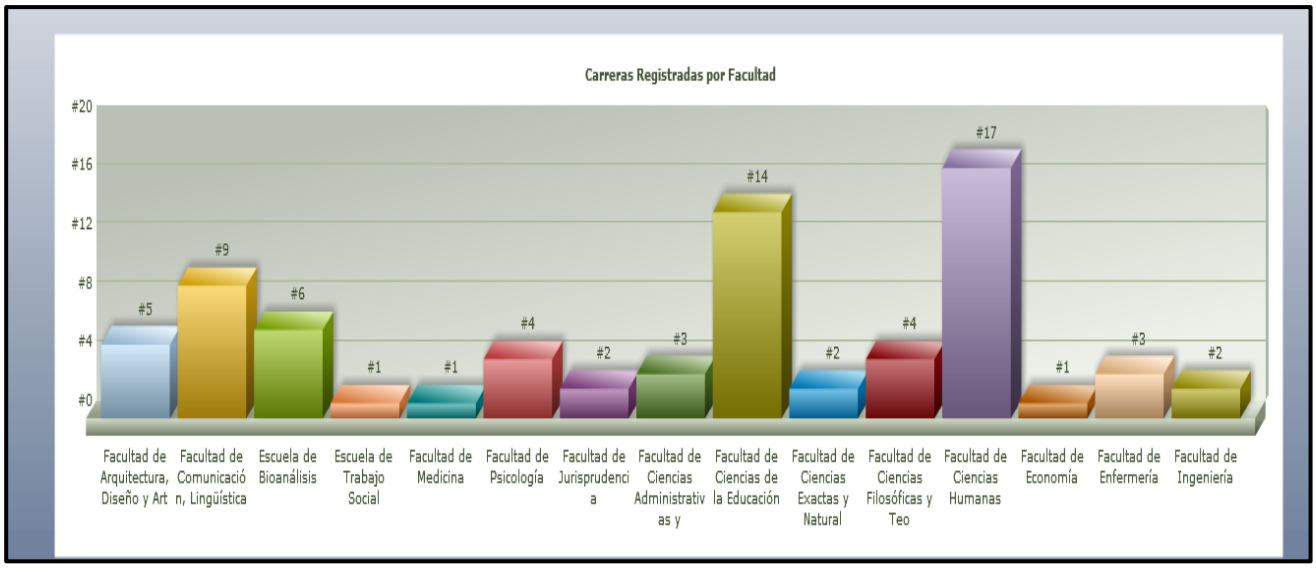


Autor: Comisión Asesora del Rector. 2011

1.1.3.3. VARIABLES DE DECISIÓN (OPTIMIZACIÓN)

En la realización de la optimización hay que considerar la estructura tradicional que tenemos, la cual se muestra en la siguiente figura, la cual no es tan óptima por el modelo que se plantea.

Figura I-2: Carreras Registradas por Facultad



Fuente: Sistema de Tutorías

La construcción del modelo tiene dos fines básicos:

En primer lugar reproducir el comportamiento histórico del sistema y, en segundo, poder simular el comportamiento que el sistema tendría frente a diversas hipótesis que pueden establecerse (cambios de los parámetros que lo definen).

Los elementos estructurales básicos de un modelo son las variables o magnitudes cuyo valor depende de una variable independiente, como por ejemplo el tiempo, los parámetros o constantes y las funciones o relaciones funcionales de las variables entre sí y de éstas con los parámetros.

Por lo cual, se ha planteado las siguientes variables:

- x_{ij}=Número de créditos a transferir desde el $Dpto_i$ (i=1,2), hasta los Prg_j (j=1,2,3)

La variable x_{ij} representa las materias que se van a dictar, la variable $Dpto_i$ representa los departamentos los cuales ofertan una determinada materia, y la variable Prg_j representa los distintos planes que se van a ir armando conforme se van tomando las distintas materias.

- c_{ij}= Costos unitarios que se producen al transferir desde el $Dpto_i$ (i=1,2), hasta los Prg_j (j=1,2,3)

La variable c_{ij}, representa en si las distintas carreras que se van armando en función de los planes generados, cabe mencionar que en este modelo hay un mínimo de créditos a cumplir para ser profesional de alguna carrera, pero no impide esto que se siga cogiendo más materias para especializarse en alguna área específica.

1.1.3.4. FUNCIÓN OBJETIVO

Minimizar el costo total de transferencia del conocimiento dado por la función:

$$C_{1\ 1}X_{1\ 1} + C_{1\ 2}X_{1\ 2} + C_{1\ 3}X_{1\ 3} + C_{2\ 1}X_{2\ 1} + C_{2\ 2}X_{2\ 2} + C_{2\ 3}X_{2\ 3}$$

Al minimizar el costo, estamos haciendo que se tomen todas las materias, propuestas en el modelo.

Sujeto a las siguientes restricciones:

1.-No Negatividad:

$x_{i\ j} \geq 0$

La parte de no negatividad, implica que no tendremos valores negativos, ya que los créditos académicos son positivos.

2.-Demanda:

$Prg_1: x_{1\ 1} + x_{2\ 1} = d_1$
$Prg_2: x_{1\ 2} + x_{2\ 2} = d_2$
$Prg_3: x_{1\ 3} + x_{2\ 3} = d_3$

La demanda está sujeta a los estudiantes que vayan a coger o tomar dichas materias, esto variará con cada semestre.

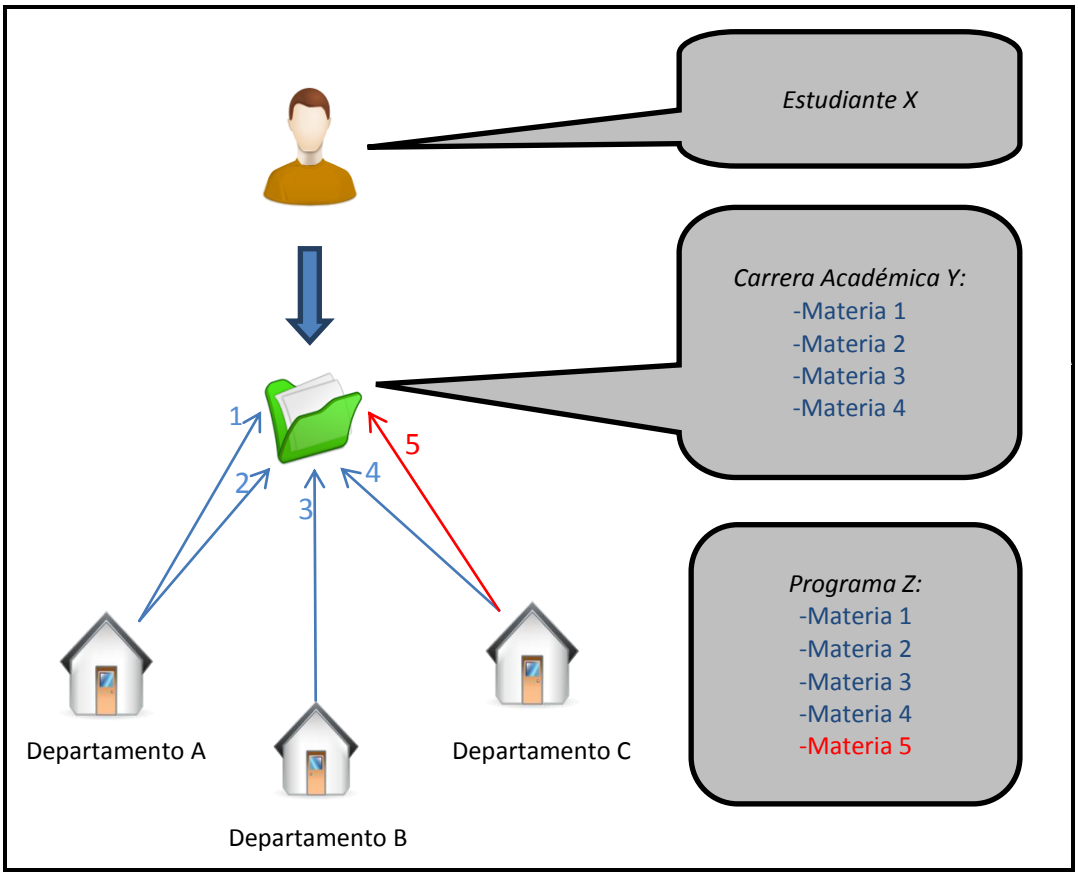
3.-Oferta:

$Dpto_1: x_{1\ 1} + x_{1\ 2} + x_{1\ 3} \leq s_1$
$Dpto_2: x_{2\ 1} + x_{2\ 2} + x_{2\ 3} \leq s_2$

En la parte de la oferta hay que considerar que no todas las carreras se ofrecen o se abren hacia el público cada semestre, lo cual hace que la oferta académica de cada departamento varié. En tal virtud, las variables de decisión deben aceptar soluciones como números reales para tener un modelo de Programación Lineal.

La siguiente imagen muestra la estructura bajo la cual se pretende trabajar en el nuevo modelo educativo.

Figura I-3: Estructura Universitaria basada en Departamentos



Autor: Campoverde, J. Oviedo, S. 2013

Un determinado alumno (Estudiante X) ha decidido seguir una carrera académica (Carrera Académica Y) la cual consta de 4 materias, la primera y segunda asignatura son ofertadas por el Departamento A, la tercera es ofertada por el Departamento B y la cuarta ofertada por el Departamento C, el estudiante además de poder tomar estas 4 materias ha optado por estudiar una quinta materia la cual no es parte del pensum de estudio de la carrera académica que se quiere culminar, para lo cual no existe ningún impedimento conformando así un plan de estudio a lo cual llamamos Programa Z, es decir aparte de poder completar los créditos obligatorios para finalizar su carrera académica dispone de créditos extras con los cuales a la par puede ir culminando otra carrera académica más.

Actualmente si un alumno de la PUCE va a cursar el primer nivel de la carrera de Ingeniería de Sistemas necesita tomar obligatoriamente los créditos referentes a 6 materias las cuales las estudiaría en su totalidad en la Facultad de Ingeniería. Bajo el nuevo modelo educativo cada asignatura podrá ser tomada en el departamento que la

oferta e indicará todas las titulaciones de la cual hace parte, adicional a esto, el estudiante estará en la capacidad de incluir materias que no sean parte del pensum obligatorio de la carrera académica inicialmente escogida, por lo cual si la persona se encuentra en la capacidad y disponibilidad de tiempo pueda ir completando a la vez una carrera adicional. Este ejemplo se estructuraría de la siguiente manera:

Tabla I-4: Ejemplo de Estructura de Créditos bajo el Nuevo Modelo Educativo.

Asignatura	Nivel	Créditos	Departamento	Titulación
Programación Básica	1	6	Ingeniería	Ingeniería de Sistemas
Algebra Lineal	1	6	Ciencias Exactas y Naturales	Ingeniería de Sistemas / Licenciatura en Matemática
Física II	1	5	Ciencias Exactas y Naturales	Ingeniería de Sistemas / Ingeniería Civil
Investigación	1	4	Formación y Orientación Social	Ingeniería de Sistemas / Administración de Empresas
Cálculo Integral	1	4	Ciencias Exactas y Naturales	Ingeniería de Sistemas / Ingeniería Civil
Cálculo Proposicional	1	4	Ingeniería	Ingeniería de Sistemas
Microeconomía	2	3	Ciencias Administrativas	Economía / Administración de Empresas

Autor: Campoverde, J. Oviedo, S. 2011

Créditos obligatorios del primer nivel de la carrera de Ingeniería de Sistemas.

Créditos opcionales.

CAPITULO II

El presente capítulo se compone del marco referencial, es decir, es en cual se exponen una serie de tópicos que sustentan teóricamente el estudio en cuestión.

2.1. SISTEMAS, MODELOS Y SIMULACIÓN[7]

En el mundo actual, tanto en el área de la educación, de los negocios, como en la industria y el gobierno, los proyectos en gran escala y de gran complejidad son la regla y no la excepción. Estos proyectos complejos requieren estudios previos a su construcción o modificación, denominados estudios pilotos. Tales estudios pilotos se realizan utilizando la técnica llamada modelización, es decir, construcción de modelos donde se realiza el estudio con el fin de obtener conclusiones aplicables al sistema real.

Construido el modelo, el proceso de ensayar en él una alternativa se llama simular. El conjunto de alternativas que se definen para su ensayo constituye la estrategia de la simulación.

Los objetivos del proyecto definen cuál es el sistema y cuál el medio ambiente que lo rodea. El sistema procura satisfacer las necesidades cambiantes de ese medio ambiente en el que está insertado. Cada nuevo sistema lo modifica y crea en él nuevas necesidades.

El sistema para poder subsistir debe adaptarse a los cambios.
Uno de los objetivos de la simulación es realizar ensayos de cambios en el sistema probándolos en el modelo, con el fin de elegir la mejor alternativa, y así enfrentar mejor a una realidad que varía día a día.

El presente proyecto tiene como objetivo principal diseñar y construir un sistema informático que simule el modelo académico "Nayón" de la Pontificia Universidad Católica del Ecuador, por lo cual está claramente identificado los siguientes parámetros:

[7] **Extraído de:** http://webdelprofesor.ula.ve/ingenieria/hhoeger/simulacion/PARTE1.pdf

a) Tipo de sistema: Sistema informático, el cual pueda ser dinámico debido a las exigencias académicas de la universidad que están en constante cambio.
b) Modelo a simular: modelo académico "Nayón"
c) Medio ambiente que lo rodea: Pontificia Universidad Católica del Ecuador.

En esta breve introducción se destacan tres conceptos fundamentales, que constituyen el eje del presente proyecto, y sobre los cuales se da a continuación una primera aproximación.

Son los conceptos de: sistema, modelo y simulación.

2.1.1. SISTEMA

Pueden darse varias definiciones de sistema:

- Conjunto de elementos cuya interacción interesa estudiar.
- Conjunto de elementos que interactúan entre sí, con un fin común, que se aísla del universo para su estudio
- Conjunto de partes organizado funcionalmente de manera tal de constituir una unidad interconectada.

Existen otras definiciones sobre sistemas más complejas que estas, tales como:

- El estudio de las interacciones conduce lógicamente al concepto de organización sistemática. (Khailov)
- Totalidad arbitraria de variables que el investigador escoge de un gran número de variables que pertenecen al sistema real. (Ashby)
- Complejo de elementos que actúan recíprocamente o la totalidad de elementos en interacción recíproca. (Bertalanffy)
- Unidad consistente en partes mutuamente interactuantes. (Ackoff)
- La totalidad de objetos junto con su mutua interacción. (Hall y Fagen)
- La esencia de estas definiciones, es el acoplamiento entre los componentes, y la organización como sistema resultante de este acoplamiento.
- Entidad que consiste en dos o más elementos y un conjunto no vacío de relaciones entre los elementos. (Francisco Sagasti)

- La palabra sistema se refiere a una colección de procesos o eventos interrelacionados, abarcados por una frontera reconocible. (F. K. Berrien)
- Un sistema puede ser definido como una reunión de componentes dotados de propiedades identificables y entre los cuales se perciben relaciones. El exterior de un sistema es su contorno. (Charles A. McClelland)

En todas estas definiciones se observa que para que constituyan un sistema los elementos deben ser varios y deben estar relacionados.

2.1.2. MODELO

La simulación de sistemas implica la construcción de modelos. El objetivo es investigar qué pasaría en el sistema si acontecieran determinadas hipótesis.

Desde mucho tiempo atrás la humanidad ha intentado adivinar el futuro. Ha querido conocer qué va a pasar cuando suceda un determinado hecho histórico. La simulación ofrece, sobre bases ciertas, esa predicción del futuro, condicionada a supuestos previos.

Para ello se construyen los modelos, normalmente una simplificación de la realidad. Surgen de un análisis de todas las variables intervinientes en el sistema y de las relaciones que se descubren existen entre ellas.

A medida que avanza el estudio del sistema se incrementa el entendimiento que el analista tiene del modelo y ayuda a crear modelos más cercanos a la realidad.

En el modelo se estudian los hechos salientes del sistema o proyecto. Se hace una abstracción de la realidad, representándose el sistema/proyecto, en un modelo.

El modelo que se construye debe tener en cuenta todos los detalles que interesan en el estudio para que realmente represente al sistema real. Por razones de simplicidad deben eliminarse aquellos detalles que no interesan y que lo complicarían innecesariamente.

Se requiere pues, que el modelo sea una fiel representación del sistema real. No obstante, el modelo no tiene porqué ser una réplica de aquél. Consiste en una descripción del sistema, junto con un conjunto de reglas que lo gobiernan.

La descripción del sistema puede ser abstracta, física o simplemente verbal. Las reglas definen el aspecto dinámico del modelo. Se utilizan para estudiar el comportamiento del sistema real.

Como ejemplo de modelo físico se pueden citar los túneles de viento donde se ensayan los aviones, los simuladores de vuelo, los canales de experiencia donde se ensayan los barcos, etc.

Como ejemplo de modelo abstracto, se pueden citar los modelos econométricos donde, entre otras cosas, se pueden ensayar las consecuencias de medidas económicas antes de aplicarlas.

Dado un sistema, son muchas las representaciones que se pueden hacer de él.
Depende de las facetas del sistema que interesan en el estudio, de la herramienta que se utiliza en el mismo e incluso de la modalidad personal del que lo construye.

2.1.3. SIMULACIÓN

Construido el modelo, se ensaya una alternativa en él con el fin de aplicar las conclusiones al sistema. Los resultados obtenidos no tienen valor si no son aplicables al sistema.

Al presente no se cuenta con información o un determinado software que permita recopilar indicadores respecto a tendencias de alumnos, personal docente y administrativo con relación al comportamiento que adaptarían bajo el modelo al que se desea entrar, todo esto origina un problema ya que no se tiene un comportamiento de la universidad, al menos supuesto o simulado, que permita llegar a conocer todas las consecuencias que traería dicho proceso de adaptación.

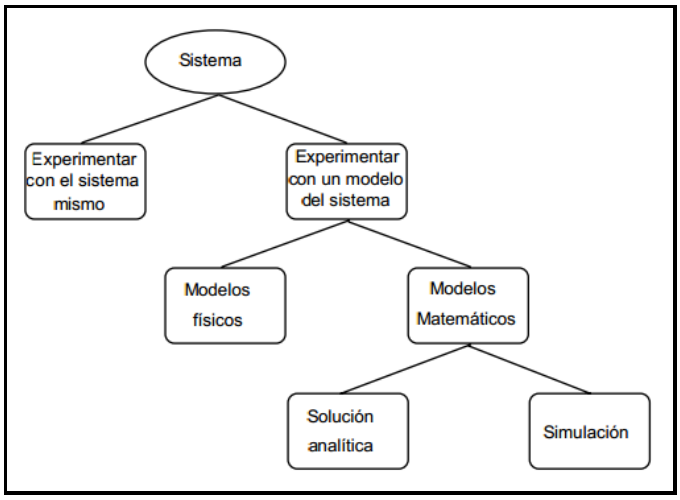
La simulación tiene como principal objetivo la predicción, es decir, puede mostrar lo que sucederá en un sistema real cuando se realicen determinados cambios bajo determinadas condiciones.

Mediante la simulación se han podido estudiar problemas y alcanzar soluciones que de otra manera hubieran resultado inaccesibles.

La simulación involucra dos facetas:

a) Construir el modelo
b) Ensayar diversas alternativas con el fin de elegir y adoptar la mejor en el sistema real, procurando que sea la óptima o que por lo menos sea lo suficientemente aproximada.

Figura II-1: La simulación en el estudio de sistemas.



Fuente: http://webdelprofesor.ula.ve/ingenieria/hhoeger/simulacion/PARTE1.pdf

2.1.4. VENTAJAS DE LA SIMULACIÓN

Permite:

- Adquirir una rápida experiencia a muy bajo costo y sin riesgos. No se compromete la confiabilidad del sistema en los ensayos.
- Identificar en un sistema complejo aquellas áreas con problema o dichas de otra manera cuellos de botella.
- Un estudio sistemático de alternativa.
- No tiene límite en cuanto a complejidad. Todo sistema, por complejo que sea, puede ser modelado, y sobre ese modelo es posible ensayar alternativas.
- Puede ser aplicada para diseño de sistemas nuevos en los cuales se quieren comparar alternativas muy diversas surgidas de utilización de diferentes tecnologías.

En el caso del presente proyecto la simulación se puede utilizar para diseñar un nuevo sistema para probar modificaciones en el modelo educativo bajo el cual la universidad se encuentra rigiendo sus funciones antes que estos cambios se implementen.

2.1.5. DESVENTAJAS DE LA SIMULACIÓN

- No se lo debe utilizar cuando existan técnicas analíticas que permitan plantear, resolver y optimizar todo el sistema o alguna parte del mismo.
  Existe un gran deseo de incluir todo en la simulación donde los resultados son visibles y comprendidos por todas las personas (especialistas o no), las cuales gustan hacer ensayos.
- No es posible asegurar que el modelo sea válido:
  Se corre el riesgo de tomar medidas erróneas basadas en aplicar conclusiones falsas obtenidas mediante un modelo que no representa la realidad.
- No existe criterio científico de selección de alternativas a simular.
  Es posible omitir una buena sugerencia de innovación simplemente porque a nadie se le ocurrió ensayarla.
- Existe el riesgo de utilizar un modelo fuera de los límites para el cual fue construido, queriendo realizar ensayos para el cual el modelo no es válido.
  Es posible elaborar todo un gran andamiaje de pruebas y resultados falsos, basados en un modelo confiable y válido bajo otras condiciones.

2.2. HERRAMIENTA PARA ALMACENAR LOS DATOS.

Los indicadores y tendencias actuales que fueron recopilados (número de Estudiantes matriculados por carrera, número de estudiantes por aula, porcentaje de aprobación, porcentaje de pérdidas, porcentaje de abandono, número de aulas, entre otros, información suministrada por el Sub decanato de la Facultad), además de los resultados de la investigación que permita tomar decisiones en cuanto a número de profesores por área académica y dedicación del alumno, número de aulas, número de créditos que un estudiante tomará de acuerdo al nuevo modelo académico de la PUCE Nayón e historial académico de los alumnos, datos individuales y grupales, deberán ser almacenados en una base de datos de manera tal que se conecte y visualice dicha información en el sistema informático.

2.2.1. BASE DE DATOS[8]

Una base de datos es un sistema informático a modo de almacén. En este almacén se guardan grandes volúmenes de información. Por ejemplo, imaginemos que somos una universidad y deseamos tener almacenados los datos personales y los reportes de calificaciones de todos nuestros estudiantes, que posiblemente sean miles de personas.

Esta información es de gran volumen de tamaño: estamos hablando de veinte o treinta datos multiplicados por miles de personas.

Llegamos a la conclusión de que necesitaríamos una base de datos para automatizar el acceso a la información y poder acceder a ella de manera rápida y fácil además de poder realizar cambios de una manera más eficiente.

Toda base de datos debe tener una serie de características tales como:

a) Seguridad: solo personas autorizadas podrán acceder a la información.
b) Integridad: la información se mantendrá sin pérdidas de datos.
c) Independencia: esta característica es fundamental ya que una buena base de datos debería ser independiente del sistema operativo o programas que interactúen con ella.

Hay más características que debe reunir una base de datos como ser consistente, es decir, que la información se guarde sin duplicidades y de manera correcta. Y finalmente, las bases de datos actuales permiten el manejo correcto de transacciones. Esto significa que se ha de permitir efectuar varias operaciones sobre la base de datos pero tratadas a modo de una sola. Es decir, si en el conjunto de las operaciones de una transacción se produce un error, entonces se deshacen todas las operaciones realizadas anteriormente y se cancela la transacción. Por ejemplo en una transferencia bancaria que comprende varias operaciones como sacar dinero de una cuenta, anotarlo en el extracto de esta cuenta, añadirlo a otra cuenta y anotarlo en el extracto de esta otra cuenta. Supóngase ahora que después de sacar el dinero de la cuenta de origen se produce un error: la transacción completa no se ha cumplido, y en este caso

[8] **Extraído de:** http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=500

la base de datos revierte lo que se haya hecho y deja la situación tal y como estaba antes de comenzar la transacción. Otro factor importante en las bases de datos es el tiempo de respuesta, que evidentemente debe ser lo más rápido posible en devolver o anotar las informaciones. Se debe tener en cuenta que una base de datos puede tener que estar enviando y anotando información correspondiente a decenas de personas conectadas a internet. Si la velocidad de respuesta no es buena, la página se quedaría bloqueada.

2.2.1.1. VENTAJAS DE LAS BASES DE DATOS[9]

La utilización de bases de datos como plataforma para el desarrollo de sistemas de aplicación en las organizaciones se ha incrementado notablemente en los últimos años y esto se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentarán a continuación:

a) Compactación: no hay necesidad de archivos en papel voluminosos.
b) Velocidad: La máquina puede recuperar y actualizar datos más rápidamente que un humano.
c) Menos trabajo laborioso: se puede eliminar gran parte del trabajo de llevar los archivos a mano. Las tareas mecánicas las realizan siempre mejor las máquinas.
d) Actualidad: en el momento que la necesitemos, tendremos a nuestra disposición información precisa y actualizada.

Los beneficios anteriores se aplican aún con más fuerza en un entorno multiusuario, donde es probable que la base de datos sea mucho más grande y compleja que en el caso de un solo usuario. No obstante, en el entorno multiusuario hay una ventaja adicional, que expresaremos así: el sistema de base de datos ofrece a la empresa un control centralizado de sus datos.

2.2.1.2. ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS[10]

Posee tres niveles con el objetivo de separar el software de aplicación con la base de datos física:

[9] **Extraído de:** Date, C. 2011
[10] **Extraído de:** http://www.um.es/docencia/barzana/IAGP/Iagp06.pdf

a) Nivel Físico: es el nivel real de los datos almacenados. Es decir cómo se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar cualificadas para ello. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.
b) Nivel Conceptual: es el correspondiente a una visión de la base de datos desde el punto de visto del mundo real. Es decir tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado. Este nivel lleva asociado el Esquema Conceptual.
c) Nivel Visión: son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeños fragmentos de ésta. El nivel visión es el encargado de dividir estos fragmentos. Un ejemplo sería el caso del empleado que no tiene por qué tener acceso al sueldo de sus compañeros o de sus superiores. El esquema asociado a éste nivel es el Esquema de Visión.

Los 3 niveles vistos, componen lo que conocemos como arquitectura de base de datos a 3 niveles.

2.2.1.3. DISEÑO DE UNA BASE DE DATOS[11]

Hoy en día el proceso de diseño de una base de datos se guía por algunos principios. El primero de ellos es que se debe evitar la información duplicada o, lo que es lo mismo, los datos redundantes, porque malgastan el espacio y aumentan la probabilidad de que se produzcan errores e incoherencias. El segundo principio es que es importante que la información sea correcta y completa. Si la base de datos contiene información incorrecta, los informes que recogen información de la base de datos contendrán también información incorrecta y, por tanto, las decisiones que tome a partir de esos informes estarán mal fundamentadas.

Un buen diseño de base de datos es, por tanto, aquél que:

- Divide la información en tablas basadas en temas para reducir los datos redundantes.

[11] **Extraído de:** http://office.microsoft.com/es-es/access-help/conceptos-basicos-del-diseno-de-una-base-de-datos-HA001224247.aspx

- Proporciona al gestor de base de datos, como MySql, Oracle o PostgreSQL, motores más utilizados en la actualidad, la información necesaria para reunir la información de las tablas cuando así se precise.
- Ayuda a garantizar la exactitud e integridad de la información.
- Satisface las necesidades de procesamiento de los datos y de generación de informes.

El proceso de diseño consta de los pasos siguientes:

d) Determinar la finalidad de la base de datos: esto le ayudará a estar preparado para los demás pasos. Es conveniente plasmar en papel el propósito de la base de datos: cómo piensa utilizarla y quién va a utilizarla.
e) Buscar y organizar la información necesaria: se reúne todos los tipos de información que desee registrar en la base de datos, como los nombres de productos o los números de pedidos.
f) Dividir la información en tablas: se divide los elementos de información en entidades o temas principales, como Productos o Pedidos. Cada tema pasará a ser una tabla.
g) Convertir los elementos de información en columnas: se decide qué información desea almacenar en cada tabla. Cada elemento se convertirá en un campo y se mostrará como una columna en la tabla. Por ejemplo, una tabla Empleados podría incluir campos como Apellido y Fecha de contratación.
h) Especificar claves principales: se elige la clave principal de cada tabla. La clave principal es una columna que se utiliza para identificar inequívocamente cada fila, como Id. de producto o Id. de pedido.
i) Definir relaciones entre las tablas: se examina cada tabla y decida cómo se relacionan los datos de una tabla con las demás tablas, es necesario agregar campos a las tablas o crear nuevas tablas para clarificar las relaciones según sea necesario.
j) Ajustar el diseño: se analiza el diseño para detectar errores, se debe crear las tablas y agregar algunos registros con datos de ejemplo. Se comprueba si puede obtener los resultados previstos de las tablas y finalmente se realiza los ajustes necesarios en el diseño.
k) Aplicar las reglas de normalización: Se debe aplicar reglas de normalización de los datos para comprobar si las tablas están estructuradas correctamente y realizar los ajustes necesarios en las tablas.

2.2.1.4. SISTEMA DE GESTIÓN DE BASE DE DATOS[12]

Se puede definir un SGBD[13] como un conjunto coordinado de programas, procedimientos, lenguajes, etc. Que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, manteniendo su integridad, confidencialidad y seguridad.
Un SGBD es una herramienta que permite interactuar los datos con los usuarios de los datos, de forma que se garanticen todas las propiedades definidas para una base de datos. En algunos casos el SGBD trabajará directamente con los datos, y en otras ocasiones, lo hará a través del sistema operativo de la maquina donde resida el SGBD.

2.2.1.5. MANEJADOR DE BASES DE DATOS MYSQL[14]

El sistema de base de datos operacional MySQL[15] es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo, característica que es difícilmente hallable en otros gestores de bases de datos.

La historia del MySQL se remite a principios de la década de 1980, siendo desarrollado por programadores de IBM. Desde esta época numerosas versiones han surgido y muchas de ellas fueron de gran importancia. Hoy en día Oracle es propietaria de Sun Mycrosystems, empresa que venía desarrollando MySQL, por ende este gestor denominado como el más popular a nivel mundial de base de datos de código abierto pasó también a ser parte de Oracle.

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación

[12] **Extraído de:** http://www.definicionabc.com/tecnologia/mysql.php
[13] **SGBD:** Sistema de Gestión de Base de Datos
[14] **Extraído de:** Mateu, C. 2004
[15] **MySQL:** My Structured Query Language

que se adaptan a diferentes necesidades y requerimientos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores.

2.3. HERRAMIENTAS PARA DESARROLLO DE SISTEMAS INFORMÁTICOS

El objetivo principal del presente proyecto es diseñar y construir un sistema informático que simule el modelo académico "Nayón" de la Pontificia Universidad Católica del Ecuador, por lo cual es necesario identificar el lenguaje de programación en el cual desarrollaremos el sistema mencionado.

2.3.1. LENGUAJES DE PROGRAMACIÓN[16]

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras.

Cada lenguaje de programación es una creación y como tal ha sido cuidadosamente diseñado.

Un lenguaje de programación, digno de su nombre, debe reunir ciertos requisitos. El lenguaje de programación debe ser universal. Es decir, cualquier problema debe tener una solución que puede ser programada en el lenguaje y dicha solución ser implementada en cualquier computador. Este requisito es uno de los más fuertes y pocos lenguajes lo poseen. Se dice que cualquier lenguaje en el cual pueden definirse funciones recursivas se considera universal. De otro lado, un lenguaje sin recursión ni iteración no puede ser universal. Existen ciertos lenguajes de aplicación que no son universales, pero sí podrían ser razonablemente descritos así mismos, como lenguajes de programación.

El lenguaje de programación debe ser implementable en una computadora, es decir; debe ser posible ejecutar un programa en términos del lenguaje en cualquier máquina. La notación matemática generalmente no es implementable porque en su notación es posible formular problemas que no pueden ser resueltos por cualquier computador.

[16] **Extraído de:** Louden, K. 2002

Los lenguajes naturales tampoco son implementables por razones totalmente diferentes: ellos son tan imprecisos y tienden a ser muy ambiguos.

Cada lenguaje tiene sintaxis y semántica:

- La sintaxis de un lenguaje de programación está relacionada con la forma de los programas, por ejemplo, como es que las expresiones, comandos, declaraciones, etc. son puestos juntos en un programa.
- La semántica de un lenguaje de programación está relacionada con el significado de los programas; por ejemplo, cómo ellos se comportarán cuando se ejecutan en una computadora.

La sintaxis de un lenguaje influye en cómo los programas son escritos por el programador, leídos por otro programador y traducidos por el computador. La semántica de un lenguaje determina como los programas son compuestos por el programador, entendidos por otros programadores e interpretados por el computador. La sintaxis es importante; pero la semántica es más importante aún.

Listar lenguajes de programación es fácil, pero clasificarlos por popularidad es complicado. Para conseguirlo, necesitamos recopilar datos sobre las ofertas de trabajo, libros publicados y vendidos o el número de respuestas de los motores de búsqueda como Google, Bing, Yahoo!, Wikipedia, Amazon ó YouTube.
El índice TIOBE[17] intenta reflejar a través de estos patrones el nivel de popularidad de los lenguajes de programación cada mes. Sin embargo, no se tiene en cuenta que lenguaje es el mejor o con el que se han escrito más líneas de código.
En la siguiente tabla se muestra los diez ganadores de febrero extraídos del ranking total producido por la comunidad de programación TIOBE:

[17] **TIOBE:** Es término con en que se conoce al índice de la comunidad de programación que permite medir la popularidad de los lenguajes de programación

Tabla II-1: Ranking 2012 de popularidad de lenguajes de programación según TIOBE

Position Mar 2012	Position Mar 2011	Delta in Position	Programming Language	Ratings Mar 2012	Delta Mar 2011	Status
1	1	=	Java	17.110%	-2.60%	A
2	2	=	C	17.087%	+1.82%	A
3	4	↑	C#	8.244%	+1.03%	A
4	3	↓	C++	8.047%	-0.71%	A
5	8	↑↑↑	Objective-C	7.737%	+4.22%	A
6	5	↓	PHP	5.555%	-1.01%	A
7	7	=	(Visual) Basic	4.369%	-0.34%	A
8	10	↑↑	JavaScript	3.386%	+1.52%	A
9	6	↓↓↓	Python	3.291%	-2.45%	A
10	9	↓	Perl	2.703%	+0.73%	A
11	13	↑↑	Delphi/Object Pascal	1.727%	+0.73%	A
12	30	↑↑↑↑↑↑↑↑↑↑	PL/SQL	1.418%	+1.01%	A
13	11	↓↓	Ruby	1.413%	-0.09%	A
14	23	↑↑↑↑↑↑↑↑	Transact-SQL	0.925%	+0.38%	A
15	15	=	Lisp	0.922%	-0.01%	A
16	22	↑↑↑↑↑↑	Visual Basic .NET	0.784%	+0.22%	A-
17	18	↑	Pascal	0.771%	+0.07%	A
18	32	↑↑↑↑↑↑↑↑↑↑	Logo	0.717%	+0.31%	A--
19	17	↓↓	Ada	0.633%	-0.09%	B
20	19	↓	NXT-G	0.604%	-0.04%	B

Fuente: http://www.redeszone.net/2012/03/14/los-lenguajes-de-programacion-actuales-volumen-i/

Como se puede comprobar, hay dos claros favoritos: Java y C. Se puede observar una línea común sin conocer en detalle estos lenguajes, C además de los tres inmediatamente posteriores: C#, C++ y Objective-C en su nombre contienen a C. Todos ellos, incluso Java, fueron inspirados en su diseño por C, y como muestra de ello, tienen una sintaxis muy similar. Quizá esa haya sido la clave del éxito, pues la curva de aprendizaje para pasar de un lenguaje a otro es menor.

Sin embargo al hablar de desarrollo de sitios web, el lenguaje de programación PHP lleva una enorme notoriedad y acogida entre los desarrolladores Web de todo el mundo debido a su naturaleza flexible y dinámica para crear páginas interactivas y atractivas capaces de atraer más tráfico hacia el sitio web en comparación a otros lenguajes.

2.3.1.1. PHP[18]

PHP[19], es un lenguaje de programación que nació como Personal Home Page Tools, lenguaje el cual fue desarrollado por el programador de origen danés Rasmus Lerdorf en 1994 con el propósito de facilitar el diseño de páginas web de carácter dinámico.

En la actualidad sin embargo, el acrónimo recursivo está vinculado a PHP Hypertext Pre-Processor. El lenguaje es desarrollado hoy en día por The PHP Group aunque carece de una normativa formal. La Free Software Foundation, por lo tanto, considera la licencia PHP como parte del software libre.

El lenguaje PHP suele procesarse directamente en el servidor aunque también puede usarse a través de software capaz de ejecutar comandos y para el desarrollo de otra clase de programas.

Actualmente el PHP suele incrustarse dentro del código HTML[20] de las páginas web y ejecutarse desde un servidor. Se estima que PHP está presente en más de veinte millones de webs y en cerca de un millón de servidores.
Una de las ventajas de PHP es su parecido con lenguajes de programación del tipo estructurado, lo que ayuda a que los programadores puedan desarrollar aplicaciones complejas en poco tiempo. De hecho, para un programador con poca experiencia en este lenguaje, es muy sencillo aprenderlo y trasladar a sus páginas funciones y estructuras que suela utilizar en la creación de otras clases de software.
Si bien existen alternativas para todos los gustos, muchos consideran PHP como una herramienta indispensable a la hora de desarrollar un sitio Web. En principio, permite establecer una conexión con la base de datos, a través de lo cual es posible, por ejemplo, presentar en pantalla datos personales del cliente cuando éste lo requiera. Pero la característica más importante de PHP es que permite modificar dinámicamente el contenido de una página, lo cual resulta esencial hoy en día.
Por ejemplo, en un diario online, se suben noticias muy frecuentemente, y la "primera plana" cambia varias veces a lo largo de una jornada. Sería muy trabajoso tener que alterar el código HTML cada vez que se deseara actualizar la página, ya que no

[18] **Extraído de:** http://www.tecn.upf.es/~ocelma/cpom/per_imprimir/tema2-lenguaje_php.html
[19] **PHP:** Acrónimo de Hypertext Pre-Processor
[20] **HTML:** HyperText Markup Language

sólo sería necesario agregar el artículo más reciente, sino ubicarlo en el primer puesto y mover el resto hacia abajo, eventualmente eliminando alguno para no sobrecargar la página. Si a esto le sumamos las reglas propias de cada diseño, que suelen indicar un tamaño diferente de imagen para la portada y otro para el texto completo, enlaces que aparecen sólo en la vista previa, entre otras actividades, dicha tarea podría ser impracticable.

Gracias a las posibilidades que ofrece PHP, con sólo actualizar la base de datos con la noticia más reciente, un sitio bien diseñado será capaz de disponer dinámicamente toda la información cada vez que alguien cargue la portada. Claro que sus ventajas no terminan ahí, y si se utiliza en conjunto con otros lenguajes, como ser Javascript, y modelos de desarrollo tales como Ajax, los resultados pueden ser deslumbrantes.
En pocas palabras, Javascript se ejecuta en el ordenador del usuario, por lo cual ofrece posibilidades complementarias a las de PHP, de manera que en conjunto pueden realizar tareas como ajustar automáticamente las proporciones de una página de acuerdo a la resolución local, cambiar el contenido sin necesidad de refrescar y recargar, así como subir archivos al servidor ofreciendo información en tiempo real del progreso de carga.

2.3.1.2. JAVASCRIPT[21]

Netscape creó el lenguaje JavaScript en 1996 y lo incluyó en su Netscape Navigator 2,0 a través de un intérprete que lee y ejecuta el código JavaScript añadido en páginas HTML. El lenguaje ha crecido en popularidad de forma constante desde entonces, y ahora está apoyado por los navegadores más populares, que hasta el mes de Enero del 2013, el ranking lo encabezaba Microsoft Internet Explorer, seguido por Firefox y Chrome en tercer lugar.

JavaScript es un lenguaje de programación interpretado. A pesar de que existen intérpretes no dependientes de ningún navegador, es un lenguaje de script[22] que suele encontrarse vinculado a páginas web. JavaScript y Java son dos lenguajes de programación distintos con filosofías muy diferentes. El único punto en común es la sintaxis, ya que cuando Netscape diseñó JavaScript, se inspiró en la sintaxis de Java.

[21] **Extraído de:** Heilmann, C. 2006
[22] **Lenguaje de script:** Es un pequeño lenguaje de programación cuyo código se inserta dentro del documento HTML

2.3.1.3. FRAMEWORK[23]

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Los frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Por ejemplo, un equipo que usa Apache Struts[24] para desarrollar un sitio web de un banco puede enfocarse en cómo los retiros de ahorros van a funcionar en lugar de preocuparse de cómo se controla la navegación entre las páginas en una forma libre de errores. Sin embargo, hay quejas comunes acerca de que el uso de frameworks añade código innecesario y que la preponderancia de frameworks competitivos y complementarios significa que el tiempo que se pasaba programando y diseñando ahora se gasta en aprender a usar frameworks.
Fuera de las aplicaciones en la informática, un framework puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

2.4. LENGUAJE UNIFICADO DE MODELADO[25]

El UML[26] es un sistema de notación que se ha convertido en estándar en el mundo de desarrollo de sistemas. Es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. El UML está constituido por un conjunto de diagramas, y

[23] **Extraído de:** http://es.wikipedia.org/wiki/Framework
[24] **Apache Struts:** Es una herramienta de soporte para el desarrollo de aplicaciones Web
[25] **Extraído de:** Schmuller, J. (s/a)
[26] **UML:** Unified Modeling Language

proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles a los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo.

Figura II-2: Logo de UML.



Fuente: Schmuller, J.

En UML, un sistema viene representado por cinco vistas diferentes que lo describen desde diferentes perspectivas. Cada vista se representa mediante un conjunto de diagramas. En UML están presentes las siguientes vistas:

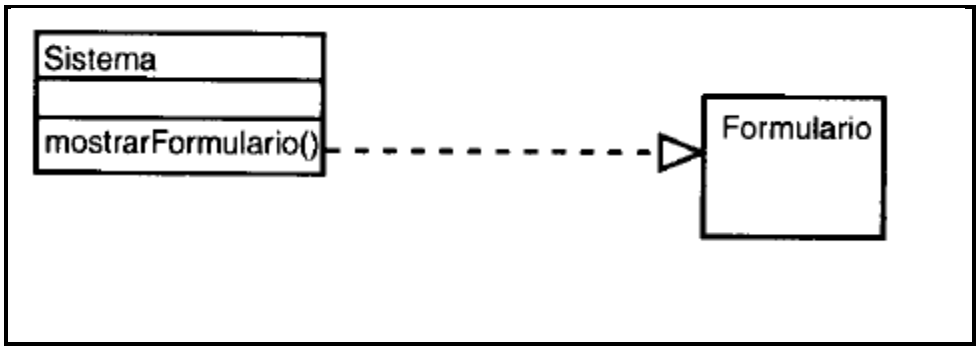
a) Vista del usuario: representa el sistema (producto) desde la perspectiva de los usuarios (llamados actores en UML). El caso de uso es el enfoque elegido para modelar esta vista.
b) Vista estructural: los datos y la funcionalidad se muestran desde dentro del sistema, es decir, modela la estructura estática (clases, objetos y relaciones).
c) Vista del comportamiento: esta parte del modelo del análisis representa los aspectos dinámicos o de comportamiento del sistema. También muestra las interacciones o colaboraciones entre los diversos elementos estructurales descritos en las vistas anteriores.
d) Vista de implementación: los aspectos estructurales y de comportamiento se representan aquí tal y como van a ser implementados.
e) Vista del entorno: aspectos estructurales y de comportamiento en el que el sistema a implementar se representa.

2.4.1. RELACIONES[27]

Dependencia: Establece una relación entre una clase dependiente y otra independiente. No establece un tipo específico de dependencia, simplemente se indica que hay una dependencia entre dos clases.

[27] **Extraído de:** Montilva, J. y Besembel, I. 2007

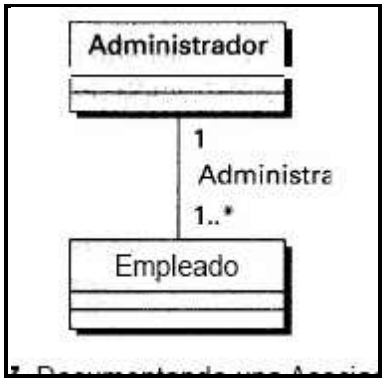
Figura II-3: Relación de dependencia entre dos clases.



Fuente: Schmuller, J.

Asociación: Establece una relación funcional y bidireccional entre dos o más clases, cada instancia de una clase se asocia a cero, uno o más instancias de la otra clase asociada.

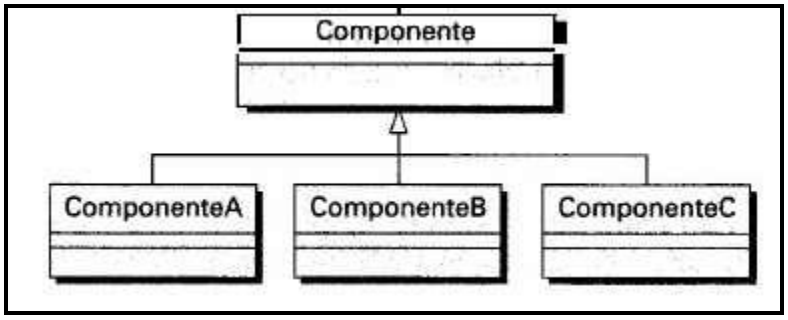
Figura II-4: Representación de una relación de asociación.



Fuente: Pressman, R. 2002

Generalización: Establece una relación del tipo "es un" entre dos o más clases. Una o más clases específicas, denominadas subclases, heredan la estructura y comportamiento de una clase genérica (superclase). Las subclases tienen (heredan) los mismos atributos y operaciones que tiene su superclase.

Figura II-5: Relación de generalización**.**



Fuente: Pressman, R. 2002

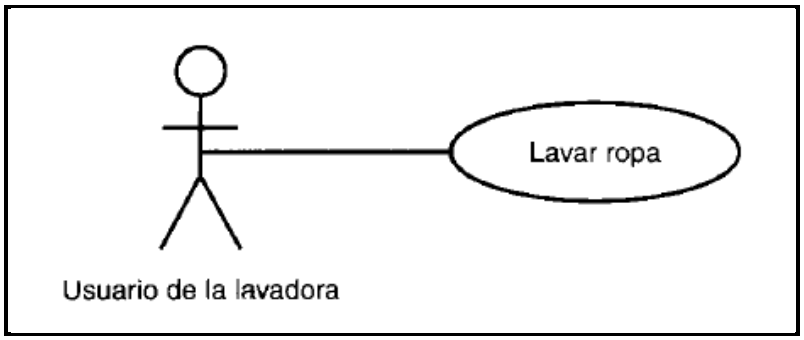
2.4.2. DIAGRAMAS

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas del sistema informático simulador a desarrollar.

2.4.2.1. DIAGRAMAS DE CASOS DE USO[28]

Es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, ésta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general, no sólo por expertos en computación.

Figura II-6: Representación de un diagrama de caso de uso**.**



Fuente: Smuller, J.

[28] **Extraído de:** Smuller, J. (s/a)

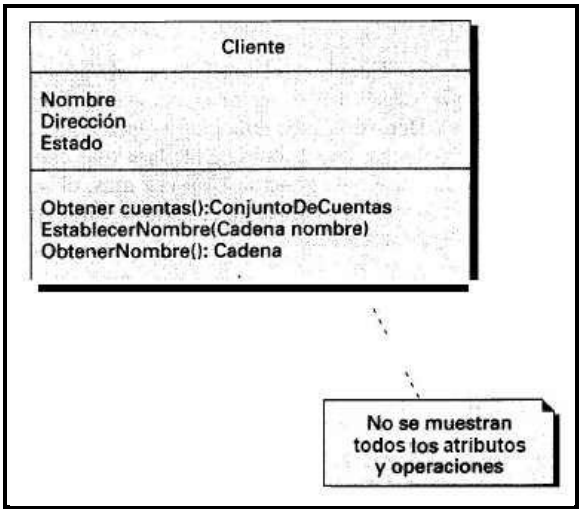
Los elementos implicados en un diagrama de casos de uso son los casos de uso, las relaciones y los actores. Un actor es un rol que interactúa con el sistema. Lo definimos como el rol, porque un actor puede ser tanto un usuario de la aplicación como otro sistema o dispositivos externos.

2.4.2.2. DIAGRAMAS DE CLASES[29]

Es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema, por ejemplo el comportamiento de objetos individuales. El primer elemento de un diagrama de clases es una descripción de clases individuales. La clase describe al cliente de un banco.

Cada cuadro que representa una clase contiene el nombre de la clase, una sección que enumera los atributos de los objetos definidos por la clase, y una sección que describe las operaciones asociadas con tales objetos.

Figura II-7: Representación de un diagrama de clases**.**



Fuente: Pressman, R. 2002

2.5. METODOLOGÍA DE DESARROLLO DE SOFTWARE RAD

La metodología de desarrollo conocida como diseño rápido de aplicaciones RAD[30] propone un proceso de desarrollo de software que permite que se creen sistemas de computadoras utilizables en un periodo de tiempo entre 60 a 90 días. RAD es un ciclo

[29] **Extraído de:** Pressman, R. 2002
[30] **RAD:** Rapid Application Development

de desarrollo diseñado para crear aplicaciones de computadoras de alta calidad de las que acontecen en corporaciones grandes.

2.5.1. ETAPAS DEL CICLO RAD[31]

La metodología RAD consiste de diferentes etapas que suceden de forma paralela y exigen la colaboración de los usuarios en todos los niveles.

2.5.1.1. ETAPA DE PLANIFICACIÓN DE LOS REQUISITOS

Esta etapa requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuáles serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.
Por lo general esta etapa se completa rápidamente cuando se crean equipos que envuelven usuarios y ejecutivos con un conocimiento amplio sobre las necesidades de la institución.

2.5.1.2. ETAPA DE ANÁLISIS Y DISEÑO

Esta consiste de un análisis detallado de las actividades de la compañía en relación al sistema propuesto. Los usuarios participan activamente en talleres bajo la tutela de profesionales de la informática. En ellos descomponen funciones y definen entidades asociadas con el sistema. Una vez se completa el análisis se crean los diagramas que definen las alteraciones entre los procesos y la data. Al finalizar el análisis se traza el diseño del sistema. Se desarrollan los procedimientos y los esquemas de pantallas.
Los prototipos de procedimientos críticos se construyen, se repasan y se prepara el plan para implementar el sistema.

2.5.1.3. CONSTRUCCIÓN

En la etapa de construcción el equipo de desarrolladores trabajando de cerca con los usuarios finaliza el diseño y la construcción del sistema. La construcción de la aplicación consiste de una serie de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados. Las pruebas al sistema se llevan a cabo

[31] **Extraído de:** http://members.fortunecity.com/miadinellie/TrabajoFinal.htm

durante esta etapa. También se crea la documentación y las instrucciones necesarias para manejar la nueva aplicación, rutinas y procedimientos para operar el sistema.

2.5.1.4. IMPLEMENTACIÓN

Esta etapa envuelve la implementación del nuevo producto y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas comprensivas y se adiestran los usuarios. Los cambios organizacionales y la operación del nuevo sistema se hacen en paralelo con el viejo sistema hasta que el nuevo se establezca completamente.

CAPITULO III

El capítulo actual se constituye del marco metodológico, en él se describen aspectos como la planificación y modelo de desarrollo del software o sitio web.

3.1. MODELO DE DESARROLLO DEL SITIO WEB

Para el desarrollo del sitio web se utilizó la metodología de desarrollo conocida como diseño rápido de aplicaciones RAD ya que esta metodología ha tomado gran impacto en el desarrollo de software debido a la necesidad que tienen las instituciones de crear aplicaciones funcionales en un plazo de tiempo corto. Esta modalidad de desarrollo consiste de diferentes etapas que suceden de forma paralela y exigen la colaboración de los usuarios en todos los niveles.

3.2. FASE PLANIFICACIÓN DE LOS REQUISITOS

La planificación tiene como objetivo la definición de cada uno de los requisitos o proceso que se va a llevar a cabo con la creación del sitio web, para llevar el control de los avances académicos en su carrera específica de los estudiantes de la PUCE, que responda de una manera más eficaz y eficiente al proceso manual que se está llevando actualmente.

Es la parte más importante del desarrollo de la aplicación web, entre los principales requerimientos tenemos:

Tabla III-1: Requerimientos del Sistema.

Referencia	Requerimiento
R.001	Ingreso al sistema web
R.002	Visualizar la lista de alumnos distribuidos por estado de matrícula y datos personales
R.003	Ingreso de nuevo alumno
R.004	Distribuir alumnos por carrera académica
R.005	Visualizar la lista de carreras académicas
R.006	Ingreso de nueva carrera académica
R.007	Modificar carrera académica
R.008	Eliminar carrera académica

R.009	Visualizar la lista de alumnos distribuidos por carrera académica
R.010	Visualizar la lista de profesores distribuidos por carrera académica
R.011	Visualizar la lista de aulas distribuidos por carrera académica
R.012	Visualizar la lista de unidades académicas con las carreras académicas pertenecientes a las mismas
R.013	Visualizar la lista de asignaturas pertenecientes a una carrera académica
R.014	Visualizar la lista de alumnos de una determina unidad académica matriculados en una asignatura
R.015	Visualizar la lista de profesores distribuidos por unidad académica
R.016	Visualizar la cantidad de aulas que fueron creadas de acuerdo a la cantidad de alumnos
R.017	Visualizar la lista de alumnos distribuidos por horarios
R.018	Ingreso de nuevo horario
R.019	Visualizar la lista de profesores distribuidos por datos personales
R.020	Modificar profesor
R.021	Eliminar profesor
R.022	Visualizar lista de niveles de estudio distribuidos por alumno
R.023	Ingreso de datos que permitan simular el comportamiento estudiantil a medida que avanza en el desarrollo de su carrera académica
R.024	Visualizar resultados del comportamiento estudiantil una vez que se va generando la simulación

Autor: Campoverde, J. Oviedo, S. 2013

3.3. FASE DE ANÁLISIS Y DISEÑO

Durante esta fase se identificó, diseñó y especificó la forma de cada uno de los componentes del sitio web. Esta fase se realizó casi en forma paralela a la fase de construcción.

3.3.1. IDENTIFICAR CASOS DE USOS Y ESQUEMATIZAR DIAGRAMAS DE CASOS DE USO.

Un actor representa el rol genérico del sistema. El nombre que se le dé a un actor deberá reflejar el papel que tendrá para el sistema.

Los casos de uso son una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Reflejan el uso que harán los actores del sistema; se muestran a través de ellos tanto las funcionalidades que ofrecerá el sistema, como los diferentes inherentes a las situaciones contempladas para cada una de estas.

Tabla III-2: Actores del Sistema para cada Caso de Uso.

Actor	**Caso de Uso**
Administrador del Sistema	Ingreso al sistema web
	Visualizar la lista de alumnos distribuidos por estado de matrícula y datos personales
	Ingreso de nuevo alumno
	Distribuir alumnos por carrera académica
	Visualizar la lista de carreras académicas
	Ingreso de nueva carrera académica
	Modificar carrera académica
	Eliminar carrera académica
	Visualizar la lista de alumnos distribuidos por carrera académica
	Visualizar la lista de profesores distribuidos por carrera académica
	Visualizar la lista de aulas distribuidos por carrera académica
	Visualizar la lista de departamentos con las carreras académicas pertenecientes a las mismas
	Visualizar la lista de asignaturas que pueden ser tomadas en un determinado departamento
	Visualizar la lista de alumnos de una determina unidad académica matriculados en una asignatura
	Visualizar la lista de profesores distribuidos por departamento
	Visualizar la cantidad de aulas que fueron creadas de acuerdo a la cantidad de alumnos
	Visualizar la cantidad de alumnos distribuidos por horarios
	Ingreso de nuevo horario
	Visualizar la lista de profesores distribuidos por datos personales
	Modificar profesor
	Eliminar profesor
	Visualizar lista de niveles de estudio distribuidos por alumno

	Ingreso de datos que permitan simular el comportamiento estudiantil a medida que avanza en el desarrollo de su carrera académica
	Visualizar resultados del comportamiento estudiantil una vez que se va generando la simulación

Autor: Campoverde, J. Oviedo, S. 2013

Se toma en cuenta un único actor, en este caso Administrador de Sistema, ya que el sistema a desarrollar es un prototipo que únicamente servirá de referencia para la obtención de datos por parte de las autoridades de la universidad que simulen la organización y asignación de recursos que se necesitarán en la sede de Nayón, así como los distintos comportamientos adoptados por los estudiantes bajo el nuevo modelo. Es decir no será un sistema de acceso múltiple el cual podrán ingresar y manejarlo distintos perfiles de usuario.

Los diagramas de Casos de Uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción, con los usuarios y/o otros sistemas. O lo que es igual un diagrama que muestre la relación entre los actores y los casos de uso en un sistema.

Estos diagramas son los primeros en generarse ya que permiten capturar de forma sencilla las especificaciones del sistema a desarrollar. Además, ofrecen una visión del sistema, que vamos a desarrollar.

3.3.2. NARRACIÓN DE LOS CASOS DE USOS ENCONTRADOS

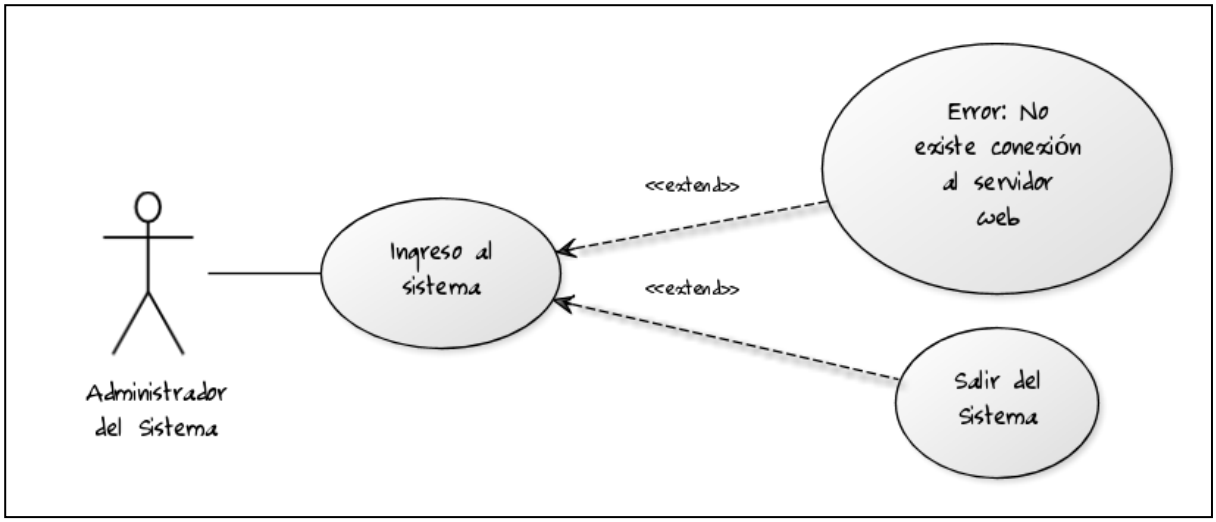
Tabla III-3: UC-001.

Identificación:	UC-001
Caso de uso:	Ingreso al sistema web
Actores:	Administrador del sistema
Propósito:	Permitir al usuario el acceso al sistema.
Descripción:	Este use case empieza cuando un usuario quiere acceder a la aplicación web.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**

1. El usuario ingresa la dirección web del sistema.	
	2. El sistema muestra la pantalla principal
Caso de uso alterno	
Paso 1: No existe conexión al servidor web.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-1: Diagrama UC-001



Fuente: Campoverde, J. Oviedo, S. 2013

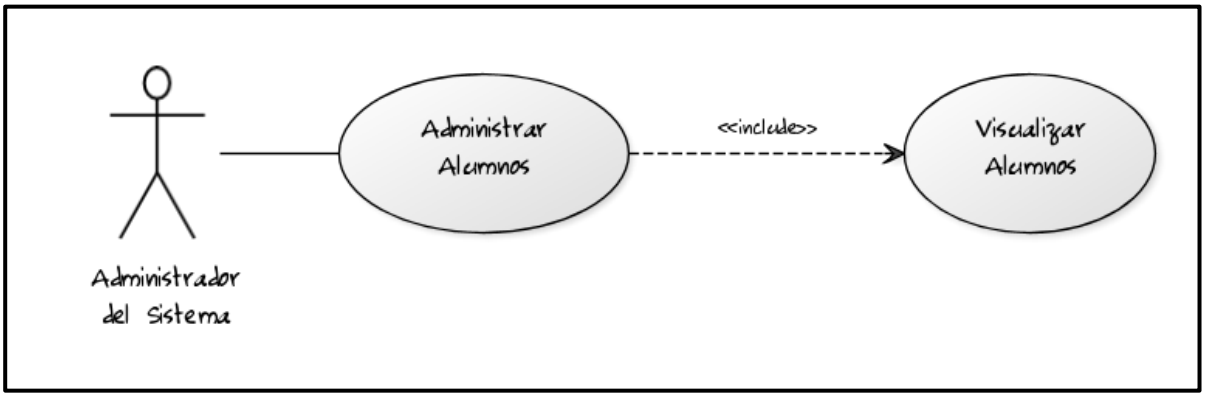
Tabla III-4: UC-002.

Identificación:	UC-002
Caso de uso:	Visualizar la lista de alumnos distribuidos por estado de matrícula y datos personales.
Actores:	Administrador del sistema
Propósito:	Presentar los alumnos con el estado actual de matrícula y datos personales.
Descripción:	Este use case empieza cuando el usuario selecciona el submenú **Ver Alumnos** del menú **Alumnos**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú	

Alumnos.	
	2. El sistema presentará un submenú con distintas opciones de selección.
3.El usuario selecciona el submenú Ver Alumnos	
	4. El sistema presentará un listado de todos los alumnos actualmente ingresados en el sistema, con su actual estado de matrícula y datos personales.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-2: Diagrama UC-002



Fuente: Campoverde, J. Oviedo, S. 2013

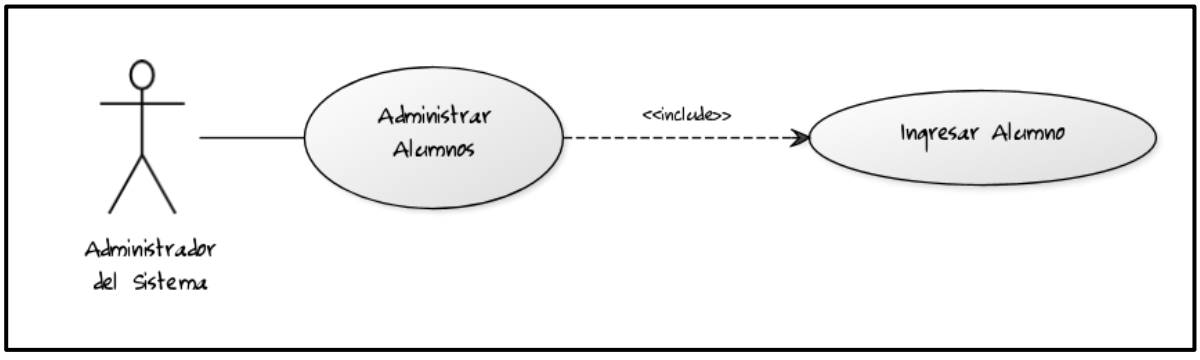
Tabla III-5: UC-003.

Identificación:	UC-003
Caso de uso:	Ingreso de nuevo alumno
Actores:	Administrador del sistema
Propósito:	Permitir ingresar los datos personales de un alumno nuevo.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Agregar Alumno** del menú **Alumnos**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**

1. El usuario selecciona el menú Alumnos.	
	2. El sistema presentará un listado de todos los alumnos actualmente ingresados en el sistema, con su actual estado de matrícula y datos personales.
3. El usuario selecciona el submenú Agregar Alumno.	
	4. El sistema muestra un formulario que le va a solicitar que ingrese los datos personales necesarios para hacer efectivo el registro del nuevo alumno.
5. El usuario ingresa los datos solicitados y presiona el botón Agregar.	
	6. El sistema valido la información ingresada y la guarda.
Caso de uso alterno	
Paso 6: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-3: Diagrama UC-003



Fuente: Campoverde, J. Oviedo, S. 2013

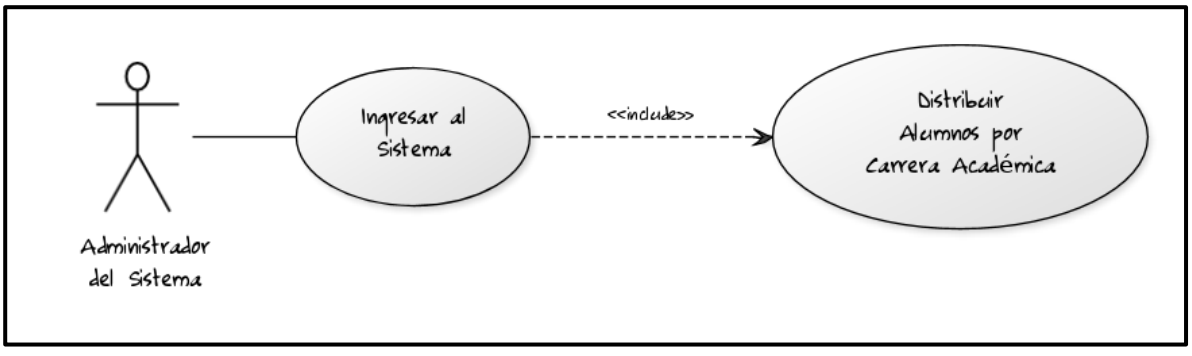
Tabla III-6: UC-004.

Identificación:	UC-004
Caso de uso:	Distribuir alumnos por carrera académica
Actores:	Administrador del sistema

Propósito:	Permitir distribuir en cada carrera académica registrada en el sistema, a todos los alumnos que han sido ingresados.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Avanzar Semestre** en la pantalla de Inicio.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona la opción Avanzar Semestre de la pantalla de inicio.	
	2. El sistema muestra la misma pantalla de inicio.
3.El usuario selecciona el submenú Ver Alumnos del menú Alumnos	
	4. El sistema presentará un listado de todos los alumnos actualmente ingresados en el sistema, además de un gráfico en el cual muestra la distribución de alumnos por carrera académica.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-4: Diagrama UC-004



Fuente: Campoverde, J. Oviedo, S. 2013

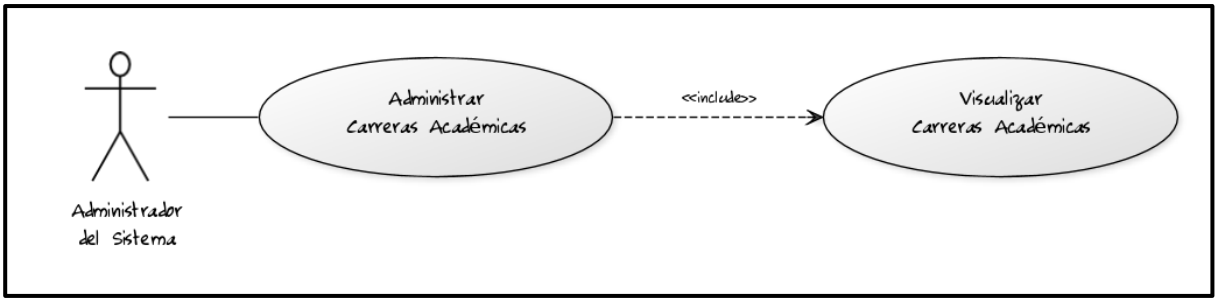
Tabla III-7: UC-005.

Identificación:	UC-005
Caso de uso:	Visualizar la lista de carreras académicas

Actores:	Administrador del sistema
Propósito:	Presentar las carreras académicas que han sido ingresadas previamente al sistema.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-5: Diagrama UC-005



Fuente: Campoverde, J. Oviedo, S. 2013

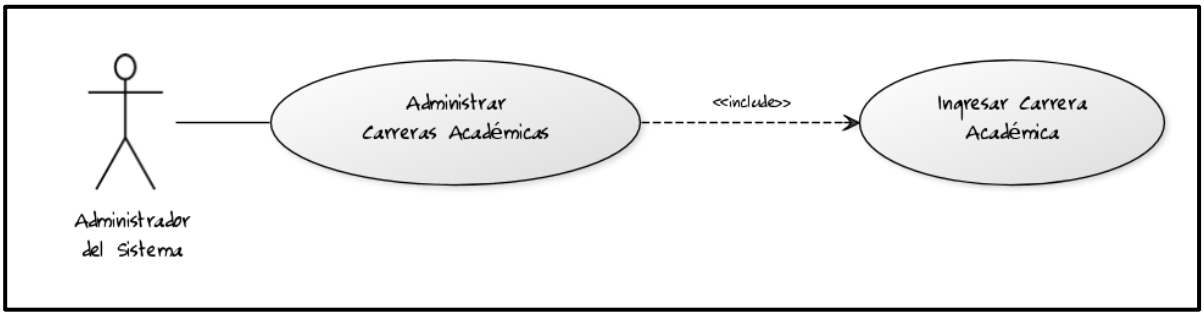
Tabla III-8: UC-006.

Identificación:	UC-006
Caso de uso:	Ingreso de nueva carrera académica
Actores:	Administrador del sistema
Propósito:	Permitir ingresar la información de una carrera académica nueva.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Agregar nueva Carrera** del menú **Carreras**.
Tipo:	Primario

Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Agregar nueva Carrera.	
	4. El sistema muestra un formulario que le va a solicitar que ingrese los datos necesarios para hacer efectivo el registro de la nueva carrera académica.
5. El usuario ingresa los datos solicitados y presiona el botón Agregar.	
	6. El sistema valido la información ingresada y la guarda.
Caso de uso alterno	
Paso 6: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-6: Diagrama UC-006



Fuente: Campoverde, J. Oviedo, S. 2013

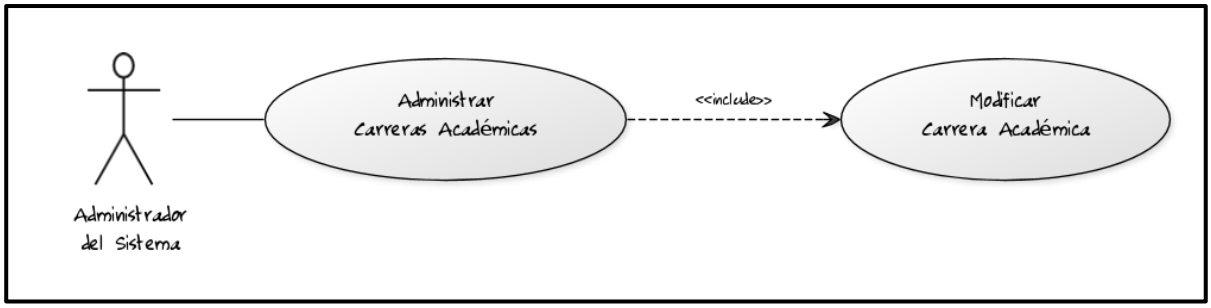
Tabla III-9: UC-007.

Identificación:	UC-007
Caso de uso:	Modificar carrera académica

Actores:	Administrador del sistema
Propósito:	Permitir al usuario cambiar la información ya ingresada de una determinada carrera académica.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Actualizar** del menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Actualizar de la carrera académica que se requiera hacerlo.	
	4. El sistema muestra un formulario con los datos actualmente registrados de la carrera académica seleccionada.
5. El usuario modifica los datos mostrados y presiona el botón Guardar.	
	6. El sistema valido la información ingresada y la guarda.
Caso de uso alterno	
Paso 6: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-7: Diagrama UC-007



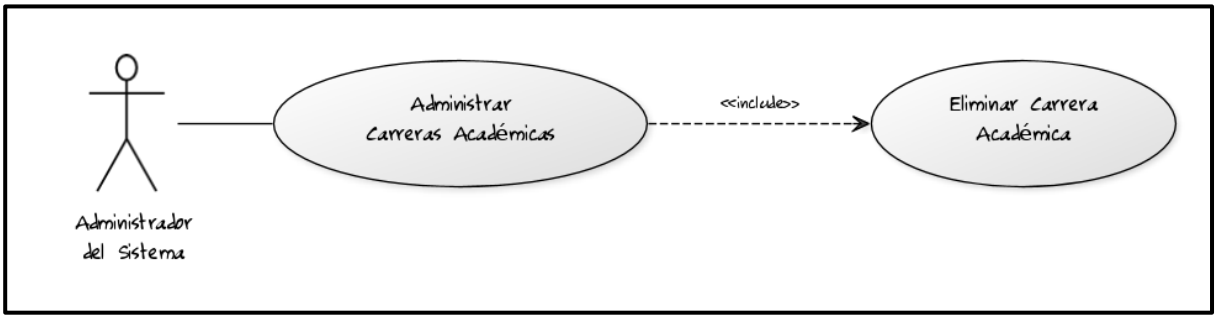
Fuente: Campoverde, J. Oviedo, S. 2013

Tabla III-10: UC-008.

Identificación:	UC-008
Caso de uso:	Eliminar carrera académica
Actores:	Administrador del sistema
Propósito:	Permitir al usuario borrar un determinado registro correspondiente a alguna carrera académica.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Eliminar** del menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Eliminar de la carrera académica que se requiera hacerlo.	
	4. El sistema presentará un listado de todas las carreras académicas sin el registro que ya ha sido previamente eliminado.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-8: Diagrama UC-008



Fuente: Campoverde, J. Oviedo, S. 2013

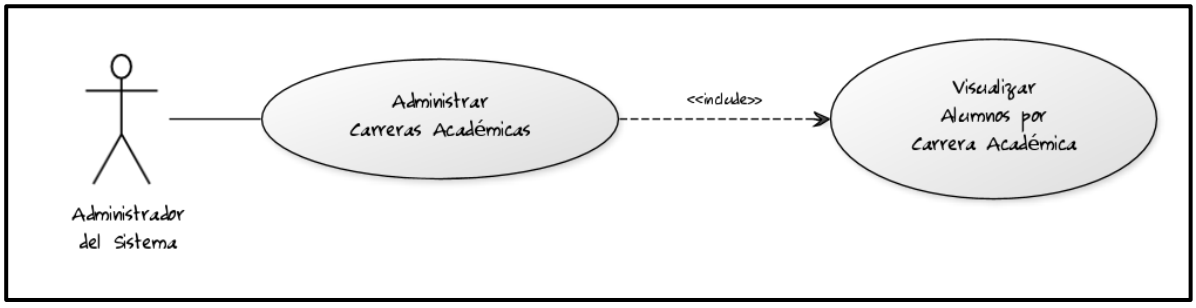
Tabla III-11: UC-009.

Identificación:	UC-009
Caso de uso:	Visualizar la lista de alumnos distribuidos por carrera académica
Actores:	Administrador del sistema
Propósito:	Presentar los alumnos que actualmente están matriculados en una determinada carrera.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Alumnos Matriculados** del menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Alumnos Matriculados de la carrera académica que se requiera visualizar.	
	4. El sistema presentará un listado de todos los alumnos que actualmente se encuentran matriculados en la carrera académica

	seleccionada.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-9: Diagrama UC-009



Fuente: Campoverde, J. Oviedo, S. 2013

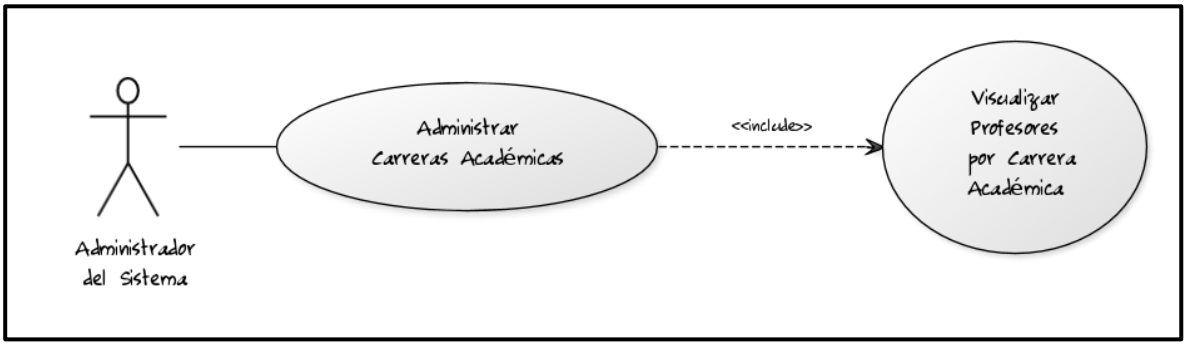
Tabla III-12: UC-010.

Identificación:	UC-010
Caso de uso:	Visualizar la lista de profesores distribuidos por carrera académica
Actores:	Administrador del sistema
Propósito:	Presentar los profesores que actualmente ejercen en una determinada carrera.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Profesores** del menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Profesores de la carrera académica que	

se requiera visualizar.	
	4. El sistema presentará un listado de todos los profesores que actualmente se encuentran ejerciendo en la carrera académica seleccionada.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-10: Diagrama UC-010



Fuente: Campoverde, J. Oviedo, S. 2013

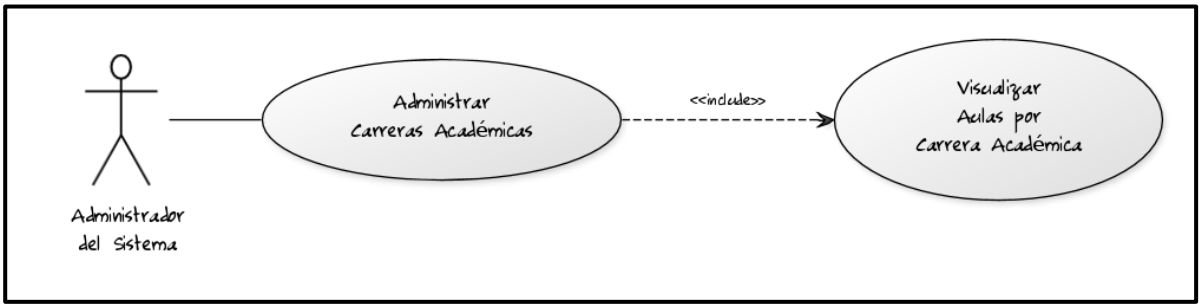
Tabla III-13: UC-011.

Identificación:	UC-011
Caso de uso:	Visualizar la lista de aulas utilizadas en una carrera académica
Actores:	Administrador del sistema
Propósito:	Presentar las aulas que actualmente se utilizan en una determinada carrera.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Aulas** del menú **Carreras**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Carreras.	
	2. El sistema presentará un listado de todas las carreras académicas actualmente ingresados en el sistema y un submenú con distintas opciones de

	selección.
4. El usuario selecciona el submenú Aulas de la carrera académica que se requiera visualizar.	
	4. El sistema presentará un listado de todas las aulas que actualmente se están utilizando en la carrera académica seleccionada.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-11: Diagrama UC-011



Fuente: Campoverde, J. Oviedo, S. 2013

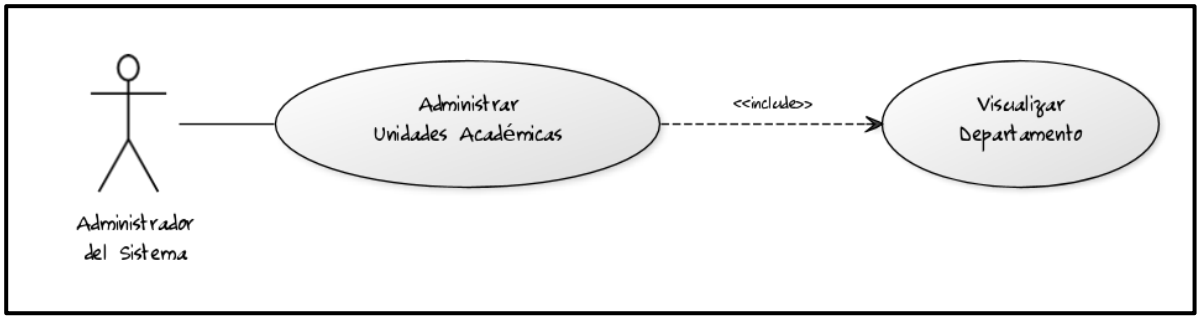
Tabla III-14: UC-012.

Identificación:	UC-012
Caso de uso:	Visualizar la lista de departamentos con las carreras académicas pertenecientes a las mismas
Actores:	Administrador del sistema
Propósito:	Presentar los departamentos que han sido ingresadas previamente al sistema.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Unidades Académicas.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Unidades Académicas.	

	2. El sistema presentará un listado de todos los departamentos actualmente ingresados en el sistema y un submenú con distintas opciones de selección.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-12: Diagrama UC-012



Fuente: Campoverde, J. Oviedo, S. 2013

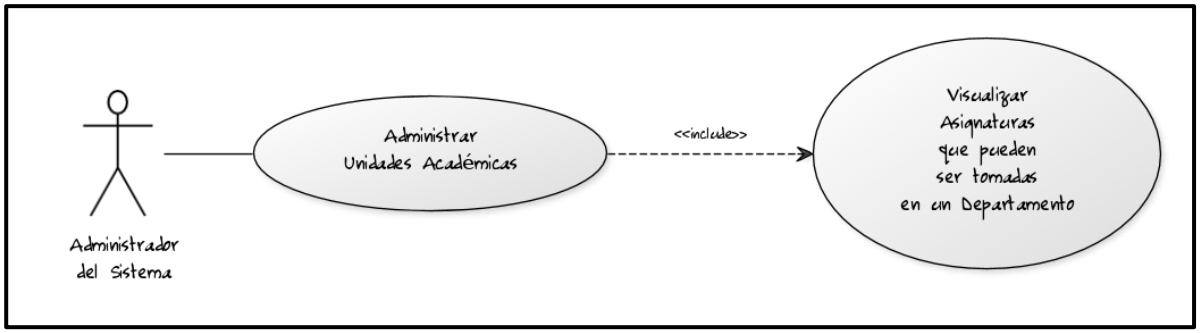
Tabla III-15: UC-013.

Identificación:	UC-013
Caso de uso:	Visualizar la lista de asignaturas que pueden ser tomadas en un determinado departamento.
Actores:	Administrador del sistema
Propósito:	Presentar las asignaturas que se pueden estudiar en una un determinado departamento.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Unidades Académicas.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Unidades Académicas.	
	2. El sistema presentará un listado de todos los departamentos actualmente ingresados en el sistema y un submenú con distintas opciones de selección.

3. El usuario selecciona un departamento.	
	4. El sistema presentará un listado de todas las asignaturas que se pueden estudiar en el departamento seleccionado.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-13: Diagrama UC-013



Fuente: Campoverde, J. Oviedo, S. 2013

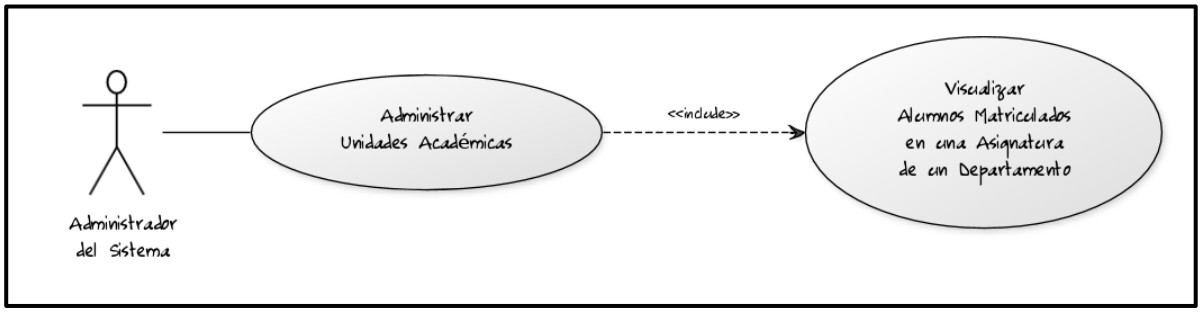
Tabla III-16: UC-014.

Identificación:	UC-014
Caso de uso:	Visualizar la lista de alumnos, de un determinado departamento, matriculados en una asignatura
Actores:	Administrador del sistema
Propósito:	Presentar los alumnos que pertenecen a un departamento y se encuentran estudiando una asignatura en otro departamento.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Unidades Académicas.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Unidades Académicas.	
	2. El sistema presentará un listado de todos los departamentos actualmente ingresados en el

	sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Foráneos del departamento requerido.	
	4. El sistema presentará un listado de todos los alumnos que se encuentran estudiando en el departamento seleccionado perteneciendo a otra diferente.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-14: Diagrama UC-014



Fuente: Campoverde, J. Oviedo, S. 2013

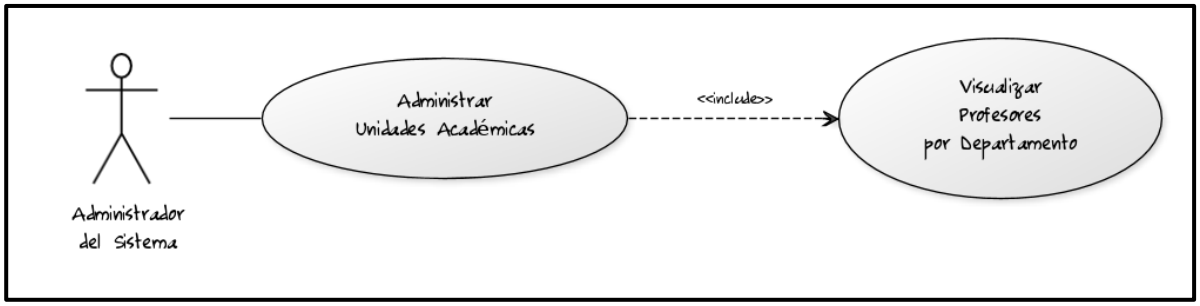
Tabla III-17: UC-015.

Identificación:	UC-015
Caso de uso:	Visualizar la lista de profesores distribuidos por departamento.
Actores:	Administrador del sistema
Propósito:	Presentar los profesores que actualmente ejercen en un determinado departamento.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Unidades Académicas.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Unidades Académicas.	

	2. El sistema presentará un listado de todos los departamentos actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Profesor de la unidad académica requerida.	
	4. El sistema presentará un listado de todos los profesores que actualmente están ejerciendo en el departamento seleccionado.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-15: Diagrama UC-015



Fuente: Campoverde, J. Oviedo, S. 2013

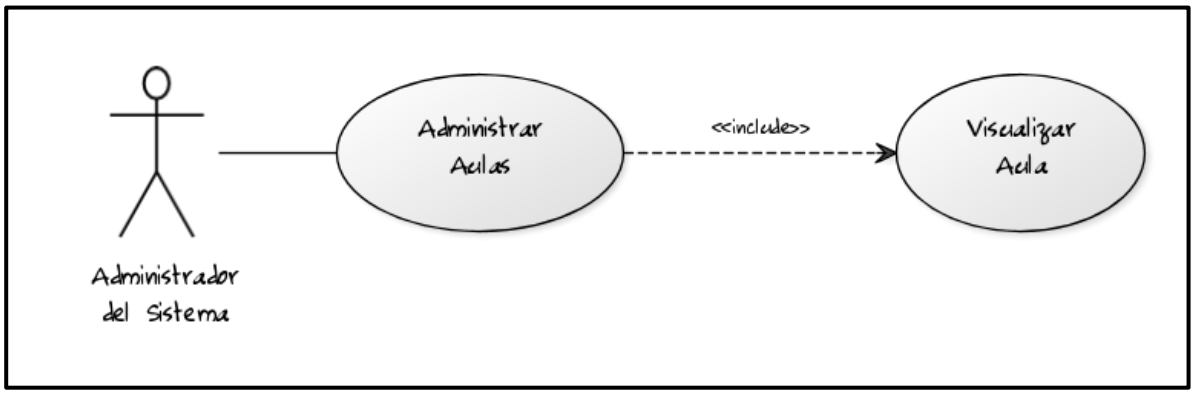
Tabla III-18: UC-016.

Identificación:	UC-016
Caso de uso:	Visualizar la cantidad de aulas que fueron creadas de acuerdo a la cantidad de alumnos
Actores:	Administrador del sistema
Propósito:	Presentar las aulas que han sido creadas para la utilización de los alumnos de acuerdo la cantidad de los mismos.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Aulas.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Aulas	

	2. El sistema presentará un listado de todas las aulas creadas de acuerdo a la cantidad de alumnos y adicionalmente la distribución de los estudiantes por aula.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-16: Diagrama UC-016



Fuente: Campoverde, J. Oviedo, S. 2013

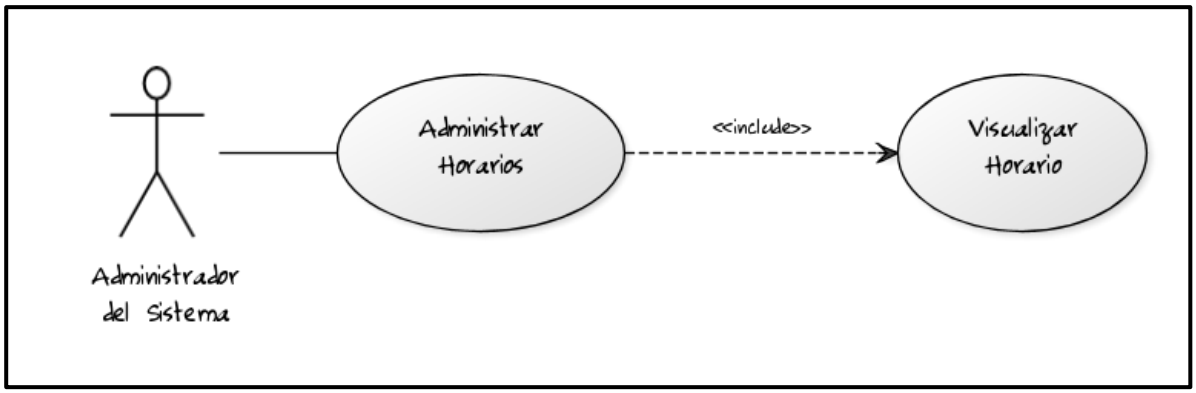
Tabla III-19: UC-017.

Identificación:	UC-017
Caso de uso:	Visualizar la cantidad de alumnos distribuidos por horarios
Actores:	Administrador del sistema
Propósito:	Presentar los horarios disponibles a los alumnos y la cantidad de estudiantes en cada uno de los mismos.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Horarios.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Horarios	
	2. El sistema presentará un listado de todos los horarios disponibles para los alumnos y adicionalmente la cantidad de estudiantes

	existentes en cada uno de dichos horarios.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-17: Diagrama UC-017



Fuente: Campoverde, J. Oviedo, S. 2013

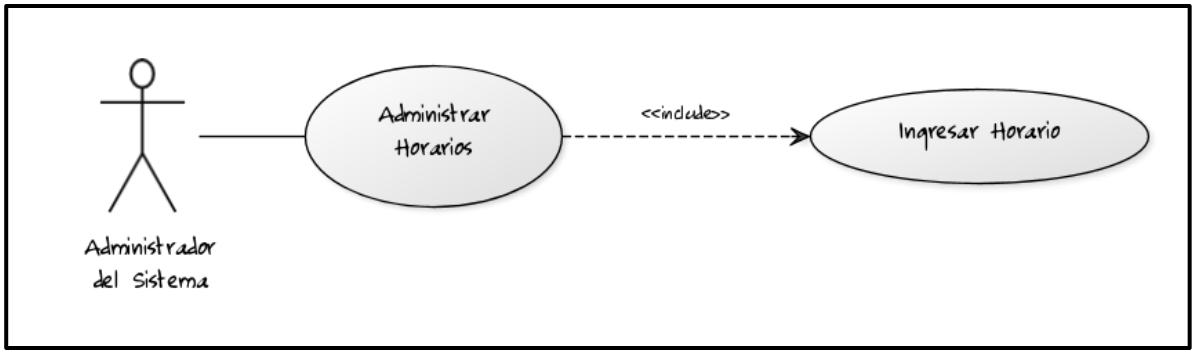
Tabla III-20: UC-018.

Identificación:	UC-018
Caso de uso:	Ingreso de nuevo horario
Actores:	Administrador del sistema
Propósito:	Permitir el ingreso de nuevos horarios para poner a disponibilidad de los alumnos.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Horarios.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Horarios	
	2. El sistema presentará un listado de todos los horarios disponibles para los alumnos y adicionalmente la cantidad de estudiantes existentes en cada uno de dichos horarios.
3. El usuario selecciona el submenú Agregar nuevo Horario.	

	4. El sistema muestra un formulario que le va a solicitar que ingrese los datos necesarios para hacer efectivo el registro del nuevo horario.
5. El usuario ingresa los datos solicitados y presiona el botón Agregar.	
	6. El sistema valida la información ingresada y la guarda.
Caso de uso alterno	
Paso 6: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-18: Diagrama UC-018



Fuente: Campoverde, J. Oviedo, S. 2013

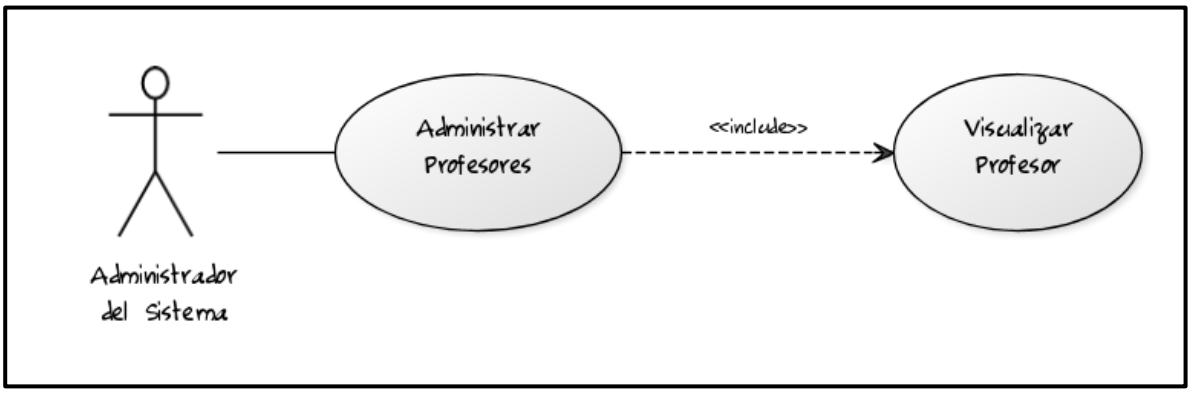
Tabla III-21: UC-019.

Identificación:	UC-019
Caso de uso:	Visualizar la lista de profesores distribuidos por datos personales
Actores:	Administrador del sistema
Propósito:	Presentar todos los profesores que han sido ingresados en el sistema.
Descripción:	Este use case empieza cuando el usuario selecciona el menú Profesores.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Profesores.	

	2. El sistema presentará un listado de todos los profesores que previamente han sido ingresados en el sistema y un submenú con distintas opciones de selección.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-19: Diagrama UC-019



Fuente: Campoverde, J. Oviedo, S. 2013

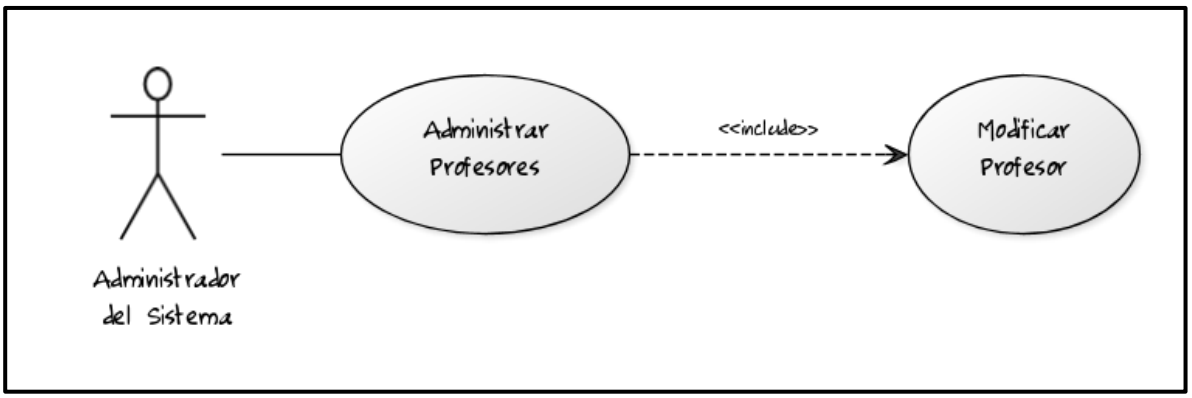
Tabla III-22: UC-020.

Identificación:	UC-020
Caso de uso:	Modificar profesor
Actores:	Administrador del sistema
Propósito:	Permitir al usuario cambiar la información ya ingresada de un determinado profesor.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Actualizar** del menú **Profesores.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Profesores.	
	2. El sistema presentará un listado de todos los profesores que previamente han sido ingresados en el sistema y un submenú con distintas opciones de selección.

3. El usuario selecciona el submenú Actualizar.	
	4. El sistema muestra un formulario con los datos actualmente registrados del profesor seleccionado.
5. El usuario modifica los datos mostrados y presiona el botón Guardar.	
	6. El sistema valida la información ingresada y la guarda.
Caso de uso alterno	
Paso 6: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-20: Diagrama UC-020



Fuente: Campoverde, J. Oviedo, S. 2013

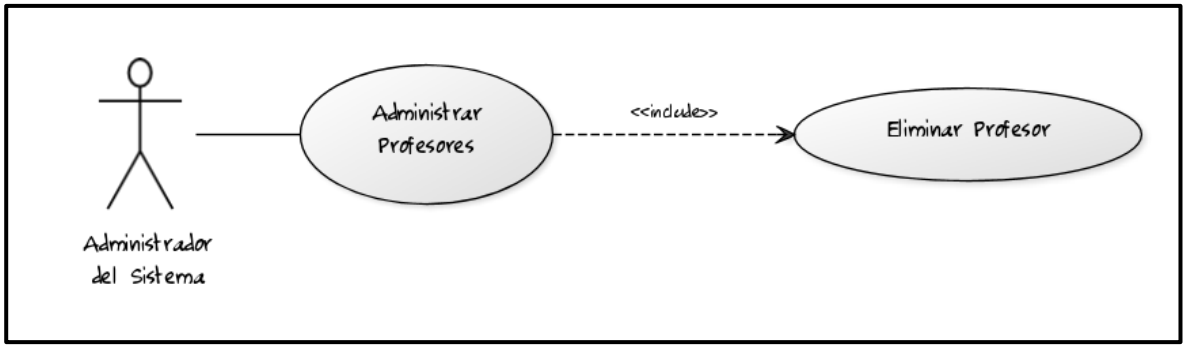
Tabla III-23: UC-021.

Identificación:	UC-021
Caso de uso:	Eliminar profesor
Actores:	Administrador del sistema
Propósito:	Permitir al usuario borrar un determinado registro correspondiente a algún profesor.
Descripción:	Este use case empieza cuando el usuario selecciona la opción **Eliminar** del menú **Profesores**.
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**

1. El usuario selecciona el menú Profesores.	
	2. El sistema presentará un listado de todos los profesores actualmente ingresados en el sistema y un submenú con distintas opciones de selección.
3. El usuario selecciona el submenú Eliminar de la carrera académica que se requiera hacerlo.	
	4. El sistema presentará un listado de todos los profesores sin el registro que ya ha sido previamente eliminado.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-21: Diagrama UC-021



Fuente: Campoverde, J. Oviedo, S. 2013

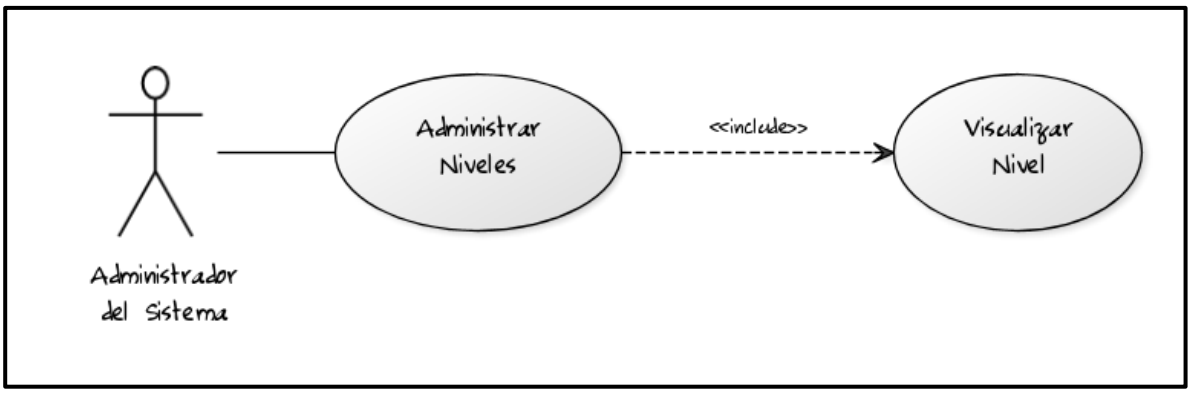
Tabla III-24: UC-022.

Identificación:	UC-022
Caso de uso:	Visualizar lista de niveles de estudio distribuidos por alumno
Actores:	Administrador del sistema
Propósito:	Presentar los niveles de estudio que se manejan para las diferentes carreras académicas y la cantidad de estudiantes en cada uno de los mismos.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Nivel.**
Tipo:	Primario

Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Nivel.	
	2. El sistema presentará un submenú con distintas opciones de selección.
3.El usuario selecciona el submenú Ver Nivel	
	4. El sistema presentará un listado de todos los niveles que actualmente se manejan para las diferentes carreras académicas y la cantidad de estudiantes en cada uno de los mismos.

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-22: Diagrama UC-022



Fuente: Campoverde, J. Oviedo, S. 2013

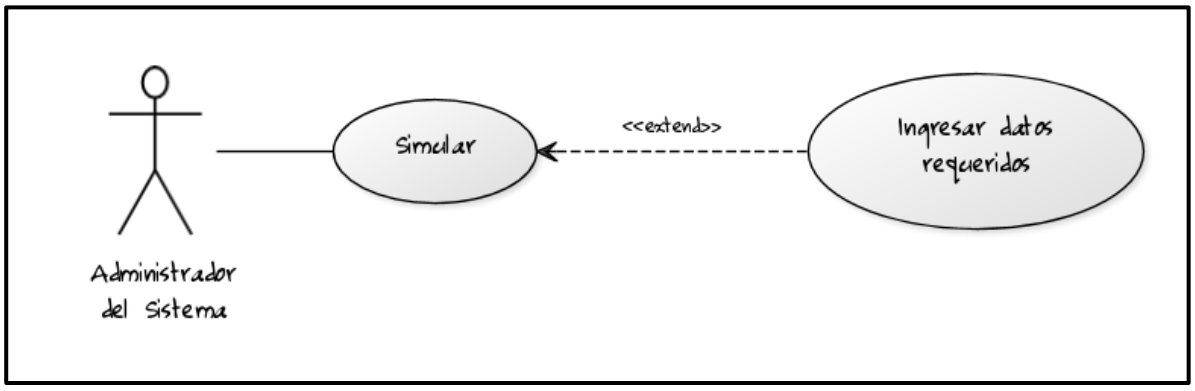
Tabla III-25: UC-023.

Identificación:	UC-023
Caso de uso:	Ingreso de datos que permitan simular el comportamiento estudiantil a medida que avanza en el desarrollo de su carrera académica.
Actores:	Administrador del sistema
Propósito:	Recopilar indicadores actuales del comportamiento promedio que tiene actualmente el estudiante durante su proceso de estudio.
Descripción:	Este use case empieza cuando el usuario selecciona el menú **Simulación.**

Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el menú Simulación.	
	2. El sistema muestra un formulario que le va a solicitar que ingrese los datos necesarios para llevar a cabo el proceso de simulación.
3. El usuario ingresa los datos solicitados y presiona el botón Crear Estudiantes.	
	4. El sistema valida la información ingresada y genera los cálculos respectivos.
Caso de uso alterno	
Paso 4: La información no se ha ingresado de manera completa o correcta.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-23: Diagrama UC-023



Fuente: Campoverde, J. Oviedo, S. 2013

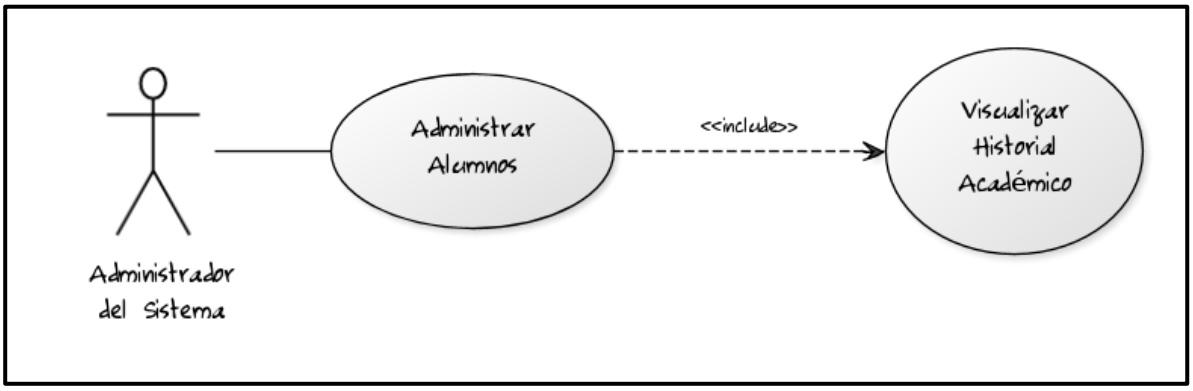
Tabla III-26: UC-024.

Identificación:	UC-024
Caso de uso:	Visualizar resultados del comportamiento estudiantil una vez que se va generando la simulación.
Actores:	Administrador del sistema

Propósito:	Identificar tendencias de alumnos, profesores, así como los factores que influyen durante el desarrollo de una determinada carrera académica y con esto, deducir los comportamientos que adoptan los estudiantes.
Descripción:	Este use case empieza cuando el usuario selecciona el submenú **Historial Académico** del menú **Alumnos.**
Tipo:	Primario
Curso típico de evento	
Acción del Actor	**Respuesta del sistema**
1. El usuario selecciona el submenú Historial Académico del menú Alumnos.	
	2. El sistema muestra un formulario que le va a solicitar seleccionar un estudiante del cual se requiera conocer cómo ha ido desarrollando sus estudios para culminar su carrera académica.
3. El usuario selecciona un estudiante.	
	4. El sistema presentará las asignaturas que han sido tomados por el alumno seleccionado a lo largo de su proceso de estudio, así como los factores (aulas, horarios, profesores, departamentos) que han influido para determinar el tiempo de culminación de la carrera académica.
Caso de uso alterno	
Paso 4: El estudiante seleccionado no muestra resultados debido a que no ha desarrollado avance en su carrera académica.	

Fuente: Campoverde, J. Oviedo, S. 2013

Figura III-24: Diagrama UC-024



Fuente: Campoverde, J. Oviedo, S. 2013

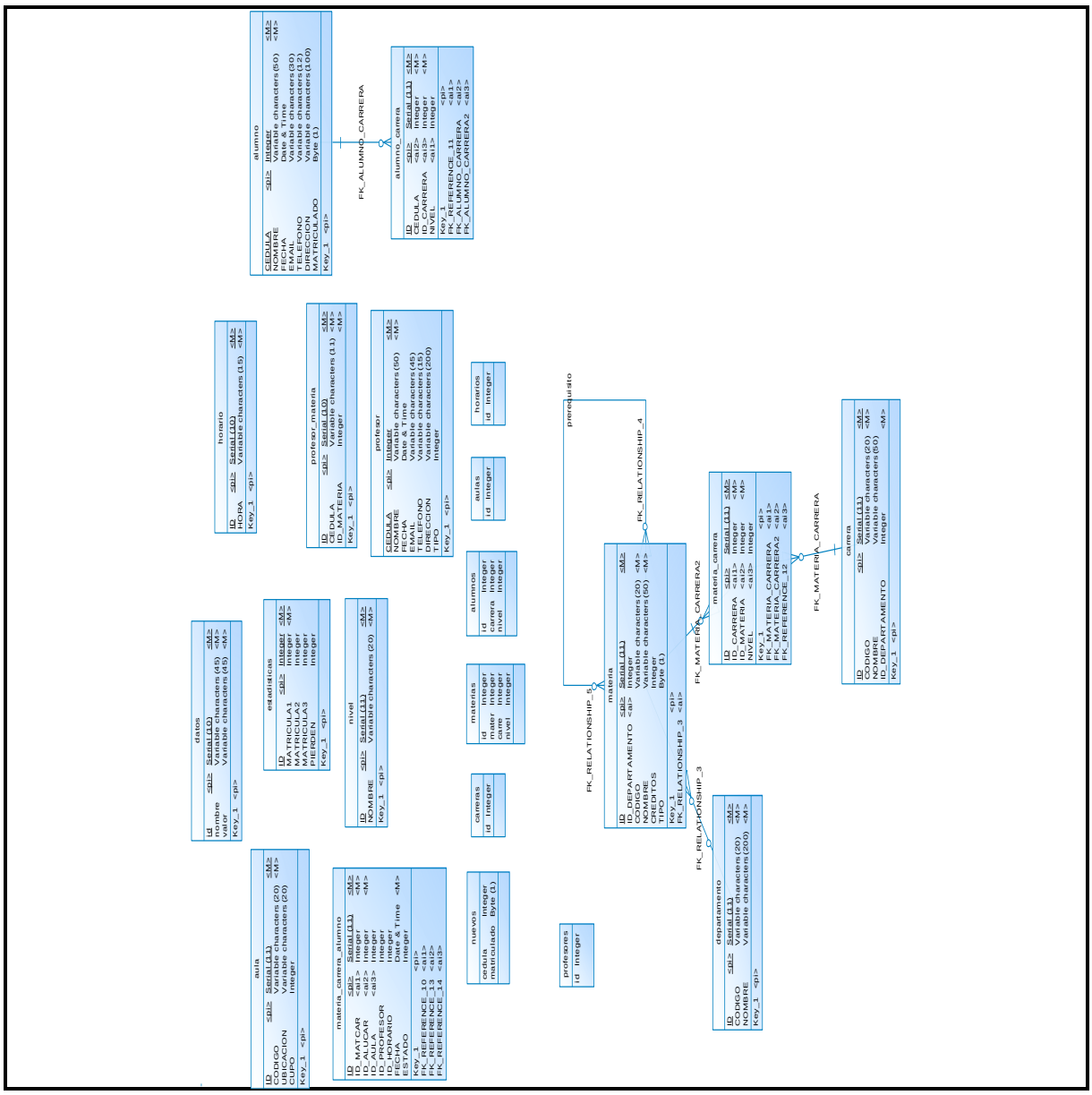
3.3.3. DISEÑO DE DATOS

Se diseñaron las estructuras de los archivos lógicos en los que se almacenaría la información relacionada al control del avance académico.

3.3.3.1. DIAGRAMA ENTIDAD RELACIÓN

Se realizó una representación gráfica conceptual de la cardinalidad de las relaciones existentes entre los archivos que conforman la base de datos del sitio web.

Figura III-25: Diagrama Entidad-Relación Modelo Educativo por Departamentos



Fuente: Campoverde, J. Oviedo, S. 2011

3.3.3.2. DISEÑO DE LA BASE DE DATOS

Las estructuras de los archivos de la base de datos, las tablas y sus campos mediante los que se relacionan son las siguientes:

1. AULA

Esta tabla almacena los datos de las aulas de la Pontificia Universidad Católica del Ecuador en su sede de Nayón.

Campo	Descripción	Tipo de Dato
ID	Clave Primaria. Identificador numérico del aula.	int(11)
CODIGO	Código del aula generado de acuerdo a su identificador y departamento al que pertenece.	varchar(20)
UBICACION	Departamento en el cual se encuentra el aula.	varchar(20)
CUPO	Cantidad de alumnos que pueden matricularse en el aula.	int(11)

2. DATOS

Registra cantidades numéricas necesarias para generar el proceso de simulación (Cantidad de Estudiantes por aula, Porcentaje de Estudiantes que salen de la universidad, Porcentaje de Estudiantes que cambian de carrera).

Campo	Descripción	Tipo de Dato
id	Clave Primaria. Identificador numérico del dato requerido para generar proceso de simulación.	int(10)
nombre	Descripción del dato requerido para generar proceso de simulación.	varchar(45)
valor	Cantidad del dato requerido para generar proceso de simulación.	varchar(45)

3. HORARIO

Almacena los horarios de clase.

Campo	Descripción	Tipo de Dato
ID	Clave Primaria. Identificador del horario de clase.	int(10)
HORA	Hora en la cual inicia hasta que termina la clase a tomar.	varchar(15)

4. MATERIA_CARRERA_ALUMNO

Esta tabla se genera de la relación entre la tabla Materia, Carrera y Alumno.

Campo	Descripción	Tipo de Dato
ID	Clave Primaria. Identificador generado.	int(11)
ID_MATCAR	Clave Foránea de la tabla Materia_Carrera.	int(11)
ID_ALUCAR	Clave Foránea de la tabla Aula_Carrera.	int(11)
ID_AULA	Clave Foránea de la tabla Aula.	int(11)

ID_PROFESOR	Clave Foránea de la tabla Profesor.	int(11)
ID_HORARIO	Clave Foránea de la tabla Horario.	int(11)
FECHA	Fecha en la cual el estudiante toma la materia.	datetime
ESTADO	Estado de matrícula del estudiante en la materia.	int(10)

5. ESTADÍSTICAS

Registra valores estadísticos necesarios para generar el proceso de simulación.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador del cálculo estadístico.	int(11)
MATRICULA1	Describe la cantidad de alumnos aprobados en primera matricula.	int(11)
MATRICULA2	Describe la cantidad de alumnos aprobados en segunda matricula.	int(11)
MATRICULA3	Describe la cantidad de alumnos aprobados en tercera matricula.	int(11)
PIERDEN	Describe la cantidad de alumnos que deben salir de la universidad por reprobación.	int(11)

6. NIVEL

En esta tabla el sistema registra los niveles de estudio que se tiene a lo largo de una carrera académica.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador del nivel.	int(11)
NOMBRE	Nombre del Nivel de Estudio.	varchar(20)

7. PROFESOR_MATERIA

Esta tabla se genera de la relación entre la tabla Profesor y Materia.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador generado.	int(10)
CEDULA	Clave foránea de la tabla Profesor.	varchar(11)
ID_MATERIA	Clave foránea de la tabla Materia.	int(10)

8. PROFESOR

Esta tabla almacena los datos personales de los profesores.

Campo	**Descripción**	**Tipo de Dato**
CEDULA	Clave Primaria. Cédula de Identidad del profesor.	int(11)
NOMBRE	Nombre del profesor.	varchar(50)
FECHA	Fecha desde la cual ejerce en la universidad el profesor.	datetime
EMAIL	Dirección de correo electrónico del profesor.	varchar(45)
TELEFONO	Teléfono del profesor.	varchar(15)
DIRECCION	Dirección de domicilio del profesor.	varchar(200)
TIPO	Tipo de profesor (Tiempo completo, medio tiempo).	int(10)

9. ALUMNO

Esta tabla almacena los datos personales de los alumnos.

Campo	**Descripción**	**Tipo de Dato**
CEDULA	Clave Primaria. Cédula de Identidad del alumno.	int(11)
NOMBRE	Nombre del alumno.	varchar(50)
FECHA	Fecha desde la cual estudia en la universidad el alumno.	datetime
EMAIL	Dirección de correo electrónico del alumno.	varchar(30)
TELEFONO	Teléfono del alumno.	varchar(12)
DIRECCION	Dirección de domicilio del alumno.	varchar(100)
MATRICULADO	Estado de matrícula del alumno.	tinyint(1)

10. ALUMNO_CARRERA

Esta tabla se genera de la relación entre la tabla Alumno y Carrera.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador generado.	int(11)
CEDULA	Clave foránea de la tabla Alumno.	int(11)
ID_CARRERA	Clave foránea de la tabla Carrera.	int(11)
NIVEL	Clave foránea de la tabla Nivel.	int(11)

11. MATERIA

Registra todas las materias a ser estudiadas por los alumnos.

Campo	**Descripción**	**Tipo de Dato**

ID	Clave Primaria. Identificador numérico de la materia.	int(11)
ID_DEPARTAMENTO	Clave foránea de la tabla Departamento.	int(11)
CODIGO	Código generado de la materia de acuerdo a su identificador y departamento al que pertenece.	varchar(20)
NOMBRE	Nombre de la materia	varchar(50)
CREDITOS	Número de créditos de la materia.	int(11)
TIPO	Descripción del tipo de materia.	tinyint(1)

12. DEPARTAMENTO

Esta tabla almacena los departamentos o unidades académicas de la universidad.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador numérico del departamento.	int(11)
CODIGO	Código del departamento generado de acuerdo a su identificador y nombre.	varchar(20)
NOMBRE	Nombre del departamento.	varchar(200)

13. MATERIA_CARRERA

Esta tabla se genera de la relación entre la tabla Materia y Carrera.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador generado.	int(11)
CODIGO	Código necesario generado entre la materia y la carrera.	varchar(20)
NOMBRE	Nombre necesario generado entre la materia y la carrera.	varchar(200)

14. CARRERA

Esta tabla registra las carreras académicas.

Campo	**Descripción**	**Tipo de Dato**
ID	Clave Primaria. Identificador numérico de la carrera.	int(11)
CODIGO	Código de la carrera generado de acuerdo a su identificador y nombre.	varchar(20)

NOMBRE	Nombre de la carrera.	varchar(50)
ID_DEPARTAMENTO	Clave foránea de la tabla Carrera.	int(10)

TABLAS TEMPORALES:

Son tablas que se eliminan cuando todos los usuarios que hacen referencia a la tabla se desconectan de la instancia de SQL Server.

15. NUEVOS

En esta tabla el sistema registra los alumnos matriculados en el sistema.

Campo	**Descripción**	**Tipo de Dato**
cedula	Clave Primaria. Cédula de identidad del alumno nuevo.	int(11)
matriculado	Indicador del estado del alumno.	tinyint(1)

16. CARRERAS

Almacena el registro de la nueva carrera ingresada al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico de la carrera.	int(11)

17. MATERIAS

Almacena el registro de la nueva materia ingresada al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico de la materia.	int(11)
carre	Clave foránea. Identificador numérico de la carrera.	int(11)
nivel	Clave foránea. Identificador numérico del nivel.	int(11)

18. ALUMNOS

Almacena el registro del nuevo alumno ingresado al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico del alumno.	int(11)
carrera	Clave foránea. Identificador numérico de la	int(11)

	carrera.	
nivel	Clave foránea. Identificador numérico del nivel.	int(11)

19. AULAS

Almacena el registro de la nueva aula ingresada al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico del aula.	int(11)

20. HORARIOS

Almacena el registro del nuevo horario ingresado al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico del horario.	int(11)

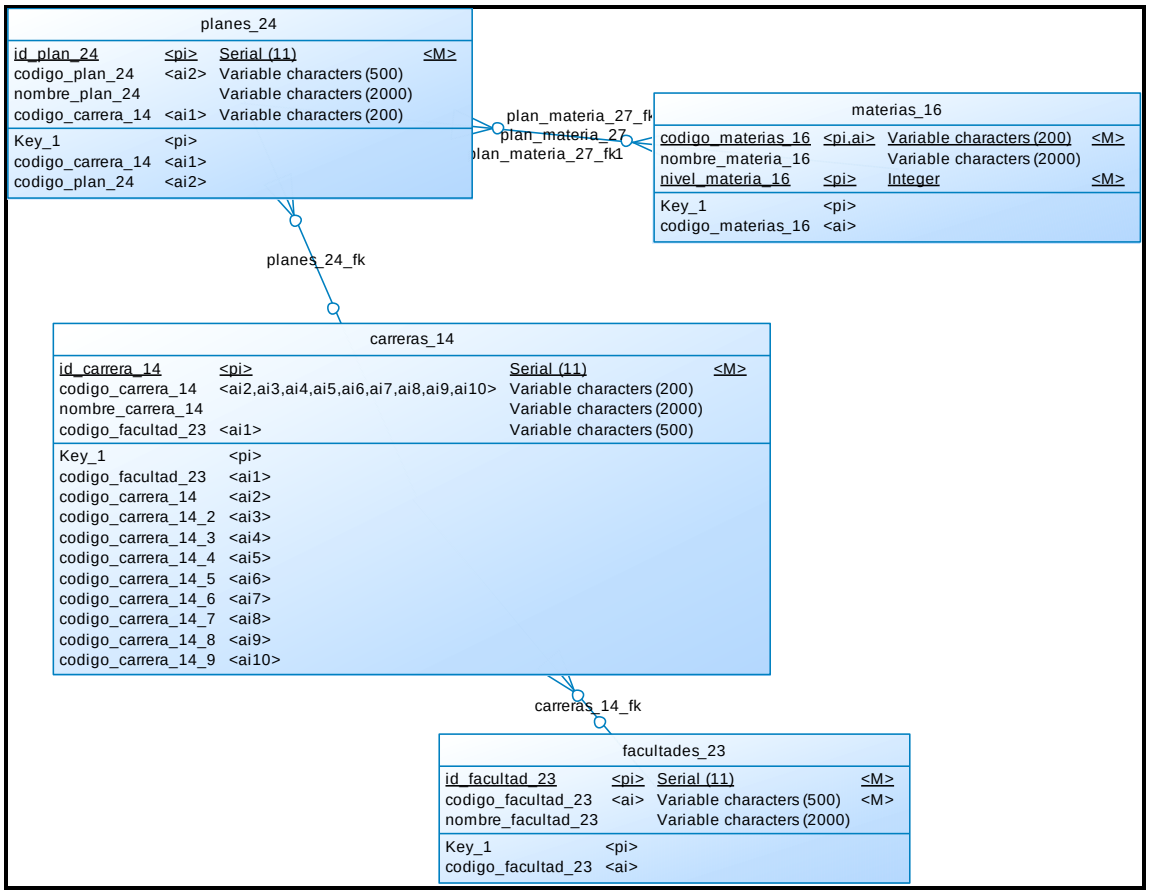
21. PROFESORES

Almacena el registro del nuevo profesor ingresado al sistema.

Campo	**Descripción**	**Tipo de Dato**
id	Clave Primaria. Identificador numérico del profesor.	int(11)

En la anterior estructura regida por Facultades, bajo la cual actualmente ejerce sus funciones la universidad tenemos que considerar lo siguiente:

Figura III-26: Diagrama Entidad-Relación Modelo Educativo por Facultades

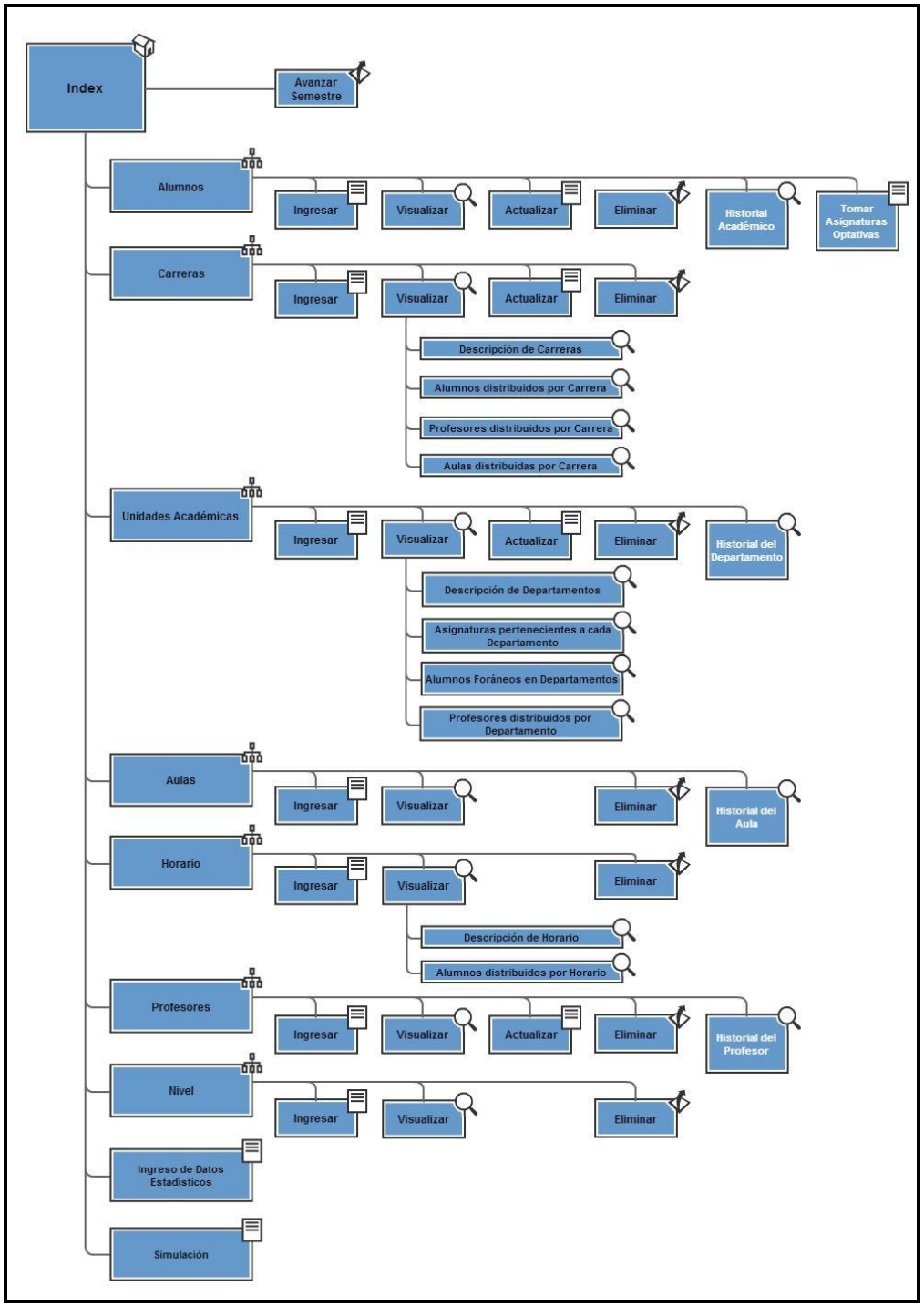


Fuente: Campoverde, J. Oviedo, S. 2011

1. Cada Carrera pertenece a una facultad y no a un departamento
2. Cada Plan de estudio esta generado por cada carrera que oferta la facultad
3. Hay materias cuyo contenido es el mismo en otras Facultades
4. No se optimizan recursos, al existir la misma materia que se da en otras facultades la estructura física como tal no está cubierta un 100%

3.3.3.3. DISEÑO DEL MAPA DE NAVEGACIÓN.

Figura III-27: Mapa de Navegación del Sitio Web**.**



Fuente: Campoverde, J. Oviedo, S. 2011

3.3.4. DISEÑO DE LA INTERFAZ HOMBRE-MÁQUINA

Durante el diseño de la interfaz hombre-máquina se establecieron las propiedades y validaciones comunes que tendrían los formularios web y menús de opciones e informes que permitirían la interacción entre el usuario y el sitio web, también se elaboraron prototipos basándose en las características comunes previamente establecidas. Los prototipos se utilizaron como patrones para continuar el diseño de la interfaz hombre-máquina, lo cual se realizó casi en forma paralela a la fase de construcción del software.

3.4. FASE DE CONSTRUCCIÓN

Durante esta tarea se elaboraron patrones de lógica de programación. En base a estos patrones se desarrollaron prototipos de código para los formularios.
La construcción del software fue desarrollada basándose en el diseño de datos y en los prototipos de interfaces y de código elaborados durante la fase de diseño.
Como se enunció anteriormente la mayoría de los formularios web fueron codificados basándose en los prototipos elaborados durante la fase de diseño. Hay bloques de código que no se basan en los prototipos debido a que tienen una lógica de programación única en todo el sitio web, sin embargo todos los segmentos de código han sido documentados.

3.5. PRUEBAS Y RESULTADOS

Para garantizar el correcto funcionamiento del sitio web se realizaron pruebas de validación tanto en la interfaz como en el funcionamiento interno de la base de datos, utilizando diferentes tipos de temas.

3.5.1. PRUEBAS DE EJECUCIÓN

En las pruebas realizadas en la interfaz se verificó que el funcionamiento del sitio web sea operativo, es decir, que los temas del avance académico sean grabados en forma adecuada y que las respuestas sean las esperadas, así como que la integridad de la información se mantenga.
También se realizaron exámenes minuciosos al momento de imprimir los reportes de los avances registrados por el usuario.

3.5.2. HARDWARE UTILIZADO PARA PRUEBAS

A fin de verificar el correcto funcionamiento del sitio web desarrollado, se han realizado las pruebas necesarias. Para la realización de las pruebas se subió el sitio web a un servidor y se lo utilizó desde distintos computadores.

Los requisitos mínimos para instalar el sistema informático que simule el modelo académico Nayón para de la Pontificia Universidad Católica del Ecuador (PUCE) y el sitio web son los siguientes:

3.5.2.1. CARACTERÍSTICAS DE COMPUTADORES PARA PRUEBAS

Tabla III-27: Características del Computador.

CARACTERÍSTICAS	DESCRIPCIÓN
Procesador	Intel Core2 Duo 2.0 GHZ
Memoria RAM	2.00 GB
Sistema Operativo	Windows XP

Fuente: Campoverde, J. Oviedo, S. 2011

3.5.2.2. CARACTERÍSTICAS DE SERVIDOR PARA PRUEBAS

Tabla III-28: Características del Servidor.

CARACTERÍSTICAS	DESCRIPCIÓN
Procesador	Intel(R) Xeon (TM) 2.80GHZ
Memoria RAM	512 MB
Sistema Operativo	Windows Server 2008

Fuente: Campoverde, J. Oviedo, S. 2011

CAPÍTULO IV

El presente capítulo muestra los resultados y conclusiones obtenidas de la investigación basada en la aplicación desarrollada.

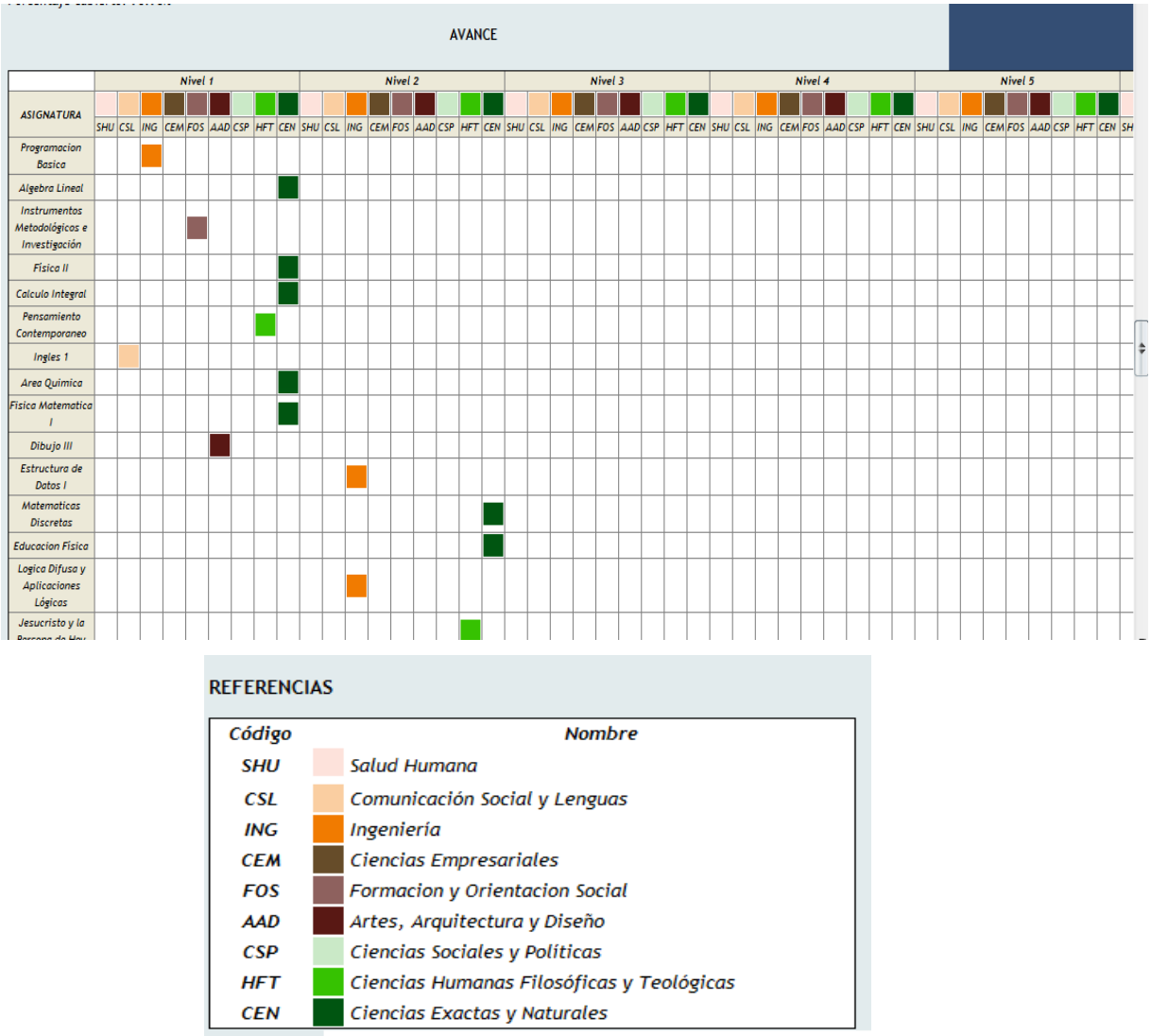
4.1. PROCESO DE LÓGICA UTILIZADA

Para la simulación se utiliza Monte Carlo, el cual realiza el análisis de riesgo con la creación de modelos de posibles resultados mediante la sustitución de un rango de valores, una distribución de probabilidad, para cualquier materia inherente. Luego, se hace que el estudiante avance en el semestre una y otra vez, cada vez usando un grupo diferente de materias aleatorias, las cuales nos indicarán al azar si pasó o perdió el semestre, verificando su avance durante toda la carrera en función de la nueva distribución por departamentos planteada anteriormente.

Dependiendo del número de materias y de los niveles especificados, para completar una simulación Monte Carlo puede ser necesario realizar miles o decenas de miles de re cálculos. La simulación Monte Carlo produce distribuciones de valores de los resultados posibles.

La aplicación de esta distribución se muestra en la siguiente imagen al ver la malla de materias de cada estudiante, conforme avanza:

Figura V-1: Lógica de Simulación



Fuente: Campoverde, J. Oviedo, S. 2013

La importancia actual del método Montecarlo se basa en la existencia de problemas que tienen difícil solución por métodos exclusivamente analíticos o numéricos, pero que dependen de factores aleatorios o se pueden asociar a un modelo probabilística artificial (resolución de integrales de muchas variables, minimización de funciones, etc.). Gracias al avance en diseño de los computadores, cálculos Montecarlo que en otro tiempo hubieran sido inconcebibles, por lo cual se presentan como asequibles para la resolución de casos como el realizado en esta simulación.

Las ventajas de está simulación son:

- Es un proceso relativamente eficiente y flexible.

- Puede ser usada para analizar y sintetizar una compleja y extensa situación académica real, pero no puede ser empleada para solucionar un modelo de análisis cuantitativo convencional.
- En este caso, esta simulación es el único método disponible.
- Los modelos de simulación se estructuran y nos resuelve en general problemas trascendentes.
- La simulación permite estudiar los efectos del avance de la carrera de cualquier estudiante, así como el tiempo que se demora en terminar.

Esta información recaudada en la simulación, será fundamental como respaldo de al decisiones académicas; no pueden quedar dudas que el conocimiento de la probabilidad de que un estudiante o los profesores está en cualquier materia específica, en sí brinda una cierta seguridad de que la información disponible ha sido empleada con la máxima eficacia.

4.2. EJECUCIÓN DEL PROGRAMA

En el navegador escribir la siguiente dirección: http://localhost/NAYONV4/
En la primera parte se visualiza un menú el cual ayuda en la simulación, adicionalmente se tiene un gráfico donde están las carreras distribuidas por el número de alumnos:

Figura V-2: Pantalla Inicio (Index)**.**



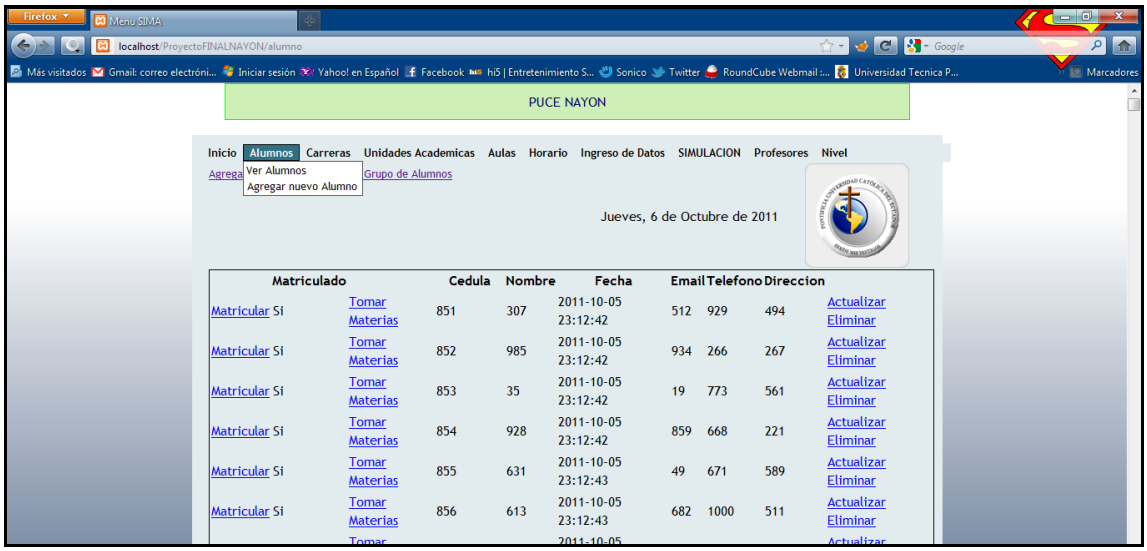
Fuente: http://localhost/NAYONV4/

4.2.1. ADMINISTRACIÓN DE ALUMNOS

4.2.1.1. VER ALUMNO

Al dar clic en alumnos, se tiene la siguiente ventana:

Figura V-3: Pantalla Administración de Alumnos**.**



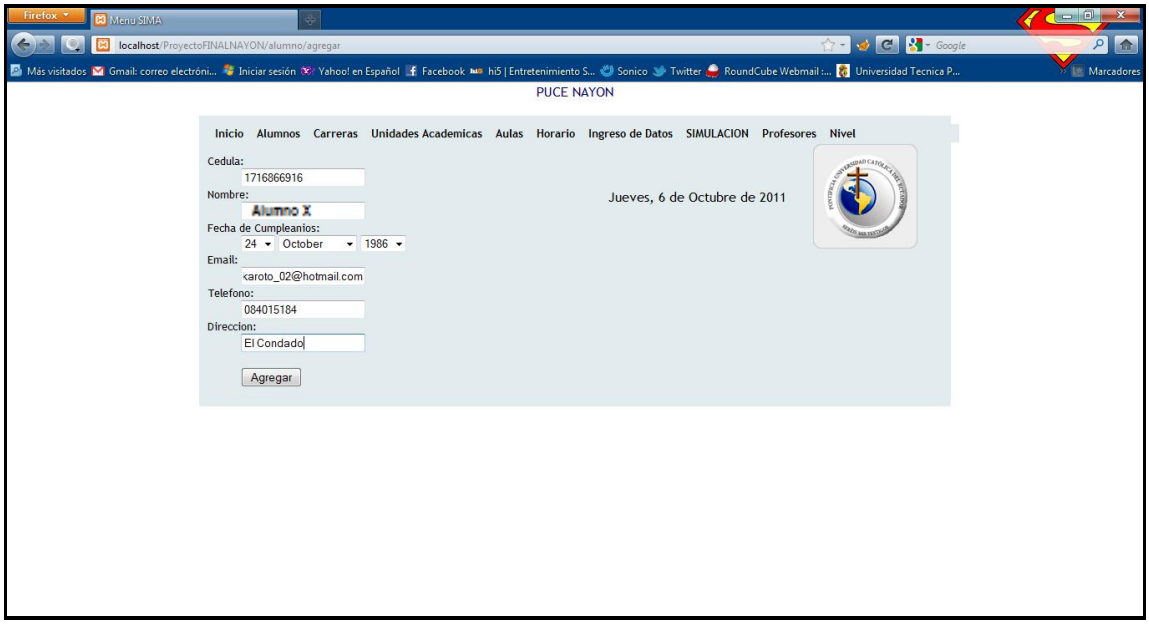
Fuente: http://localhost/NAYONV4/

En esta ventana se tiene dos parámetros de entrada (agregar alumno, y agregar grupo de alumnos)

4.2.1.2. AGREGAR ALUMNOS

Al dar clic en agregar alumno se tiene la siguiente ventana:

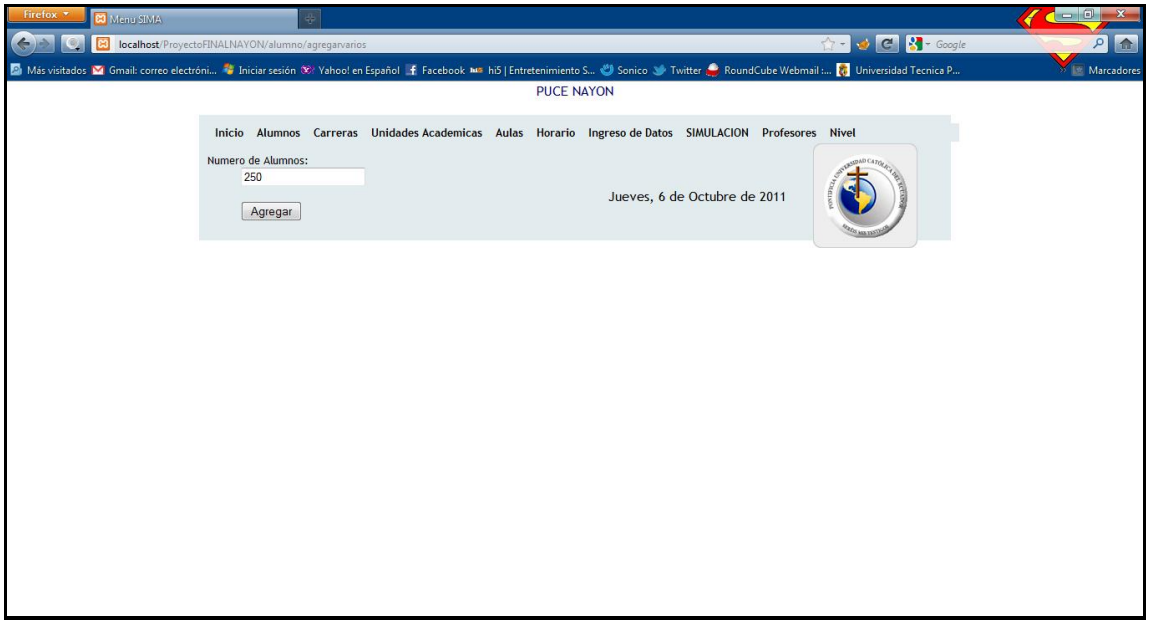
Figura V-4: Pantalla Agregar Alumno**.**



Fuente: http://localhost/NAYONV4/

Al dar clic en agregar grupo de alumnos se muestra lo siguiente:

Figura V-5: Pantalla Agregar Grupo de Alumno**.**



Fuente: http://localhost/NAYONV4/

4.2.2. AVANZAR SEMESTRE

Para distribuir los alumnos en los diferentes departamentos es necesario dar clic en Inicio, posterior a lo cual es necesario hacer clic en avanzar semestre:

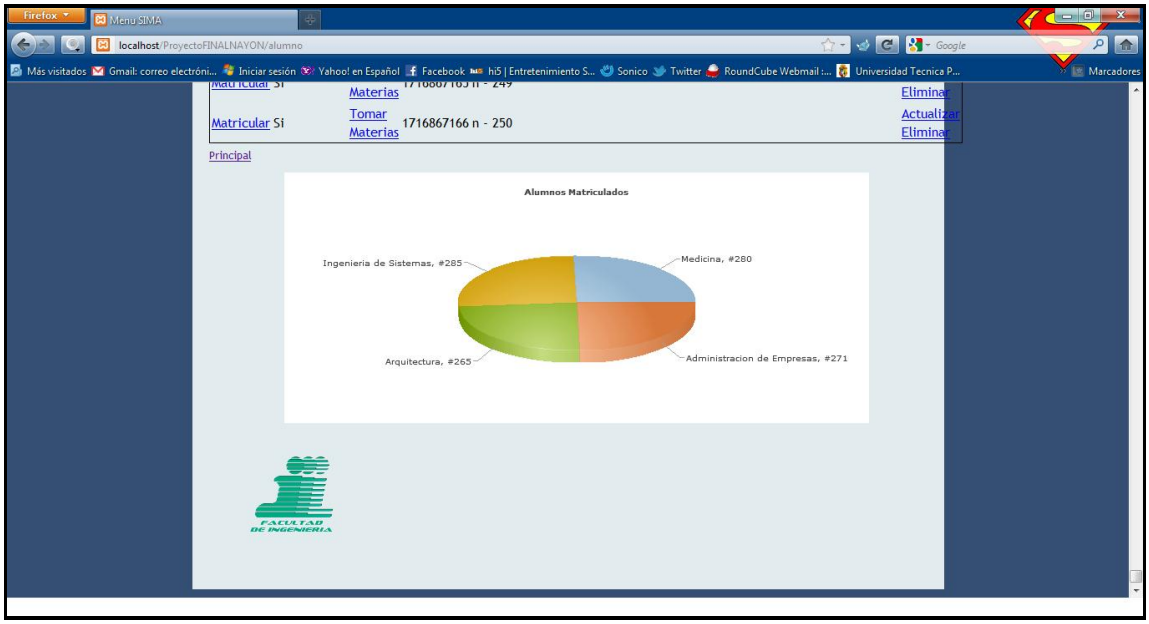
Figura V-6: Pantalla Avanzar Semestre**.**



Fuente: http://localhost/NAYONV4/

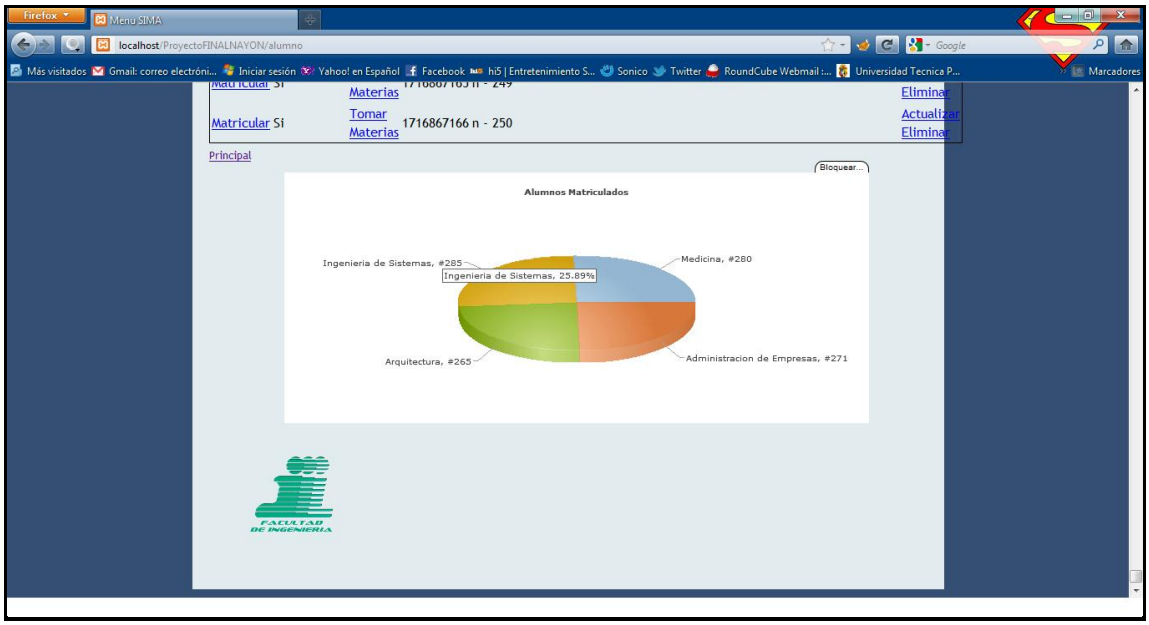
Al dar clic en Alumnos, al final se visualiza un gráfico el cual muestra la distribución por carrera:

Figura V-7: Distribución de Alumnos por Carrera (Cantidad)**.**



Fuente: http://localhost/NAYONV4/

Figura V-8: Distribución de Alumnos por Carrera (Porcentaje)**.**



Fuente: http://localhost/NAYONV4/

4.2.3. ADMINISTRACIÓN DE CARRERAS

Al hacer clic en carreras se tiene lo siguiente:

Figura V-9: Pantalla Administración de Carreras**.**

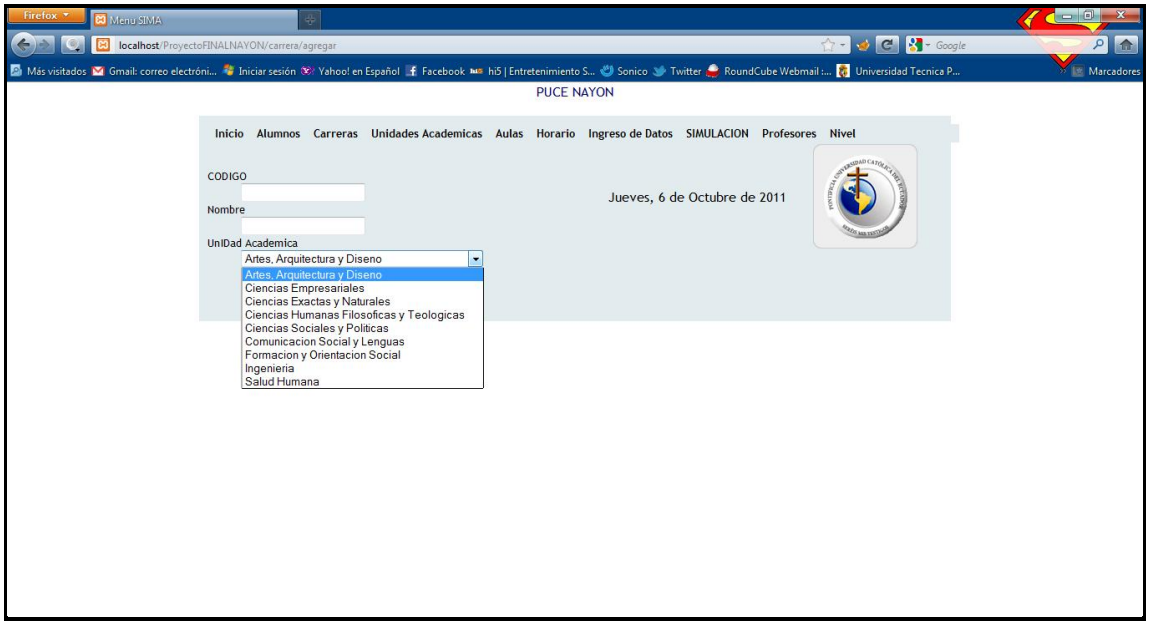


Fuente: http://localhost/NAYONV4/

4.2.3.1. NUEVA CARRERA

Dentro de los parámetros de entrada se tiene: agregar nueva carrera, al dar clic en agregar nueva carrera se muestra lo siguiente:

Figura V-10: Pantalla Agregar Carrera.

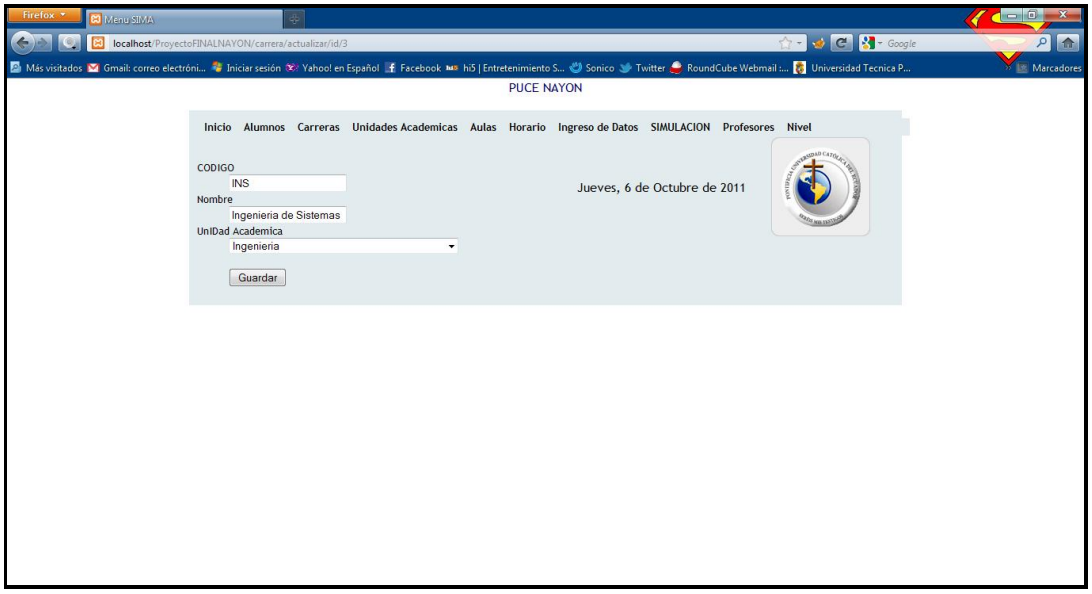


Fuente: http://localhost/NAYONV4/

4.2.3.2. ACTUALIZACIÓN DE CARRERAS

Regresando a carreras se tiene a lado de las carreras un link llamado actualizar.
Al dar clic en actualizar se visualiza lo siguiente:

Figura V-11: Pantalla Actualizar Carrera.

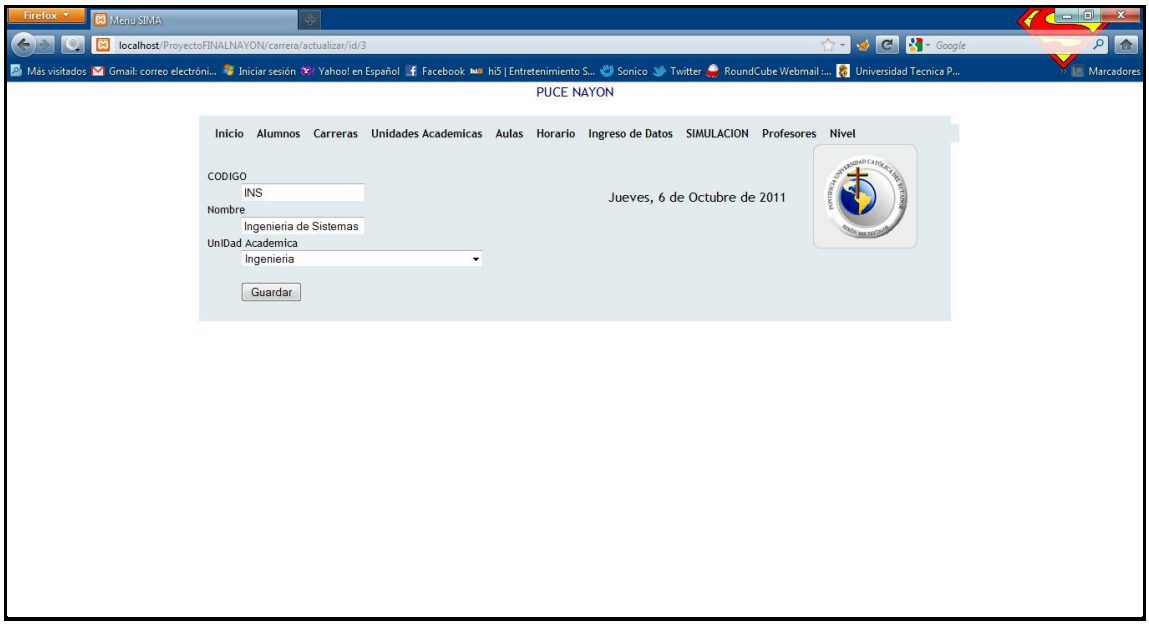


Fuente: http://localhost/NAYONV4/

4.2.3.3. ELIMINACIÓN DE CARRERAS

Al regresar a carreras y dar clic en eliminar se tiene lo siguiente:

Figura V-12: Pantalla Eliminar Carrera**.**

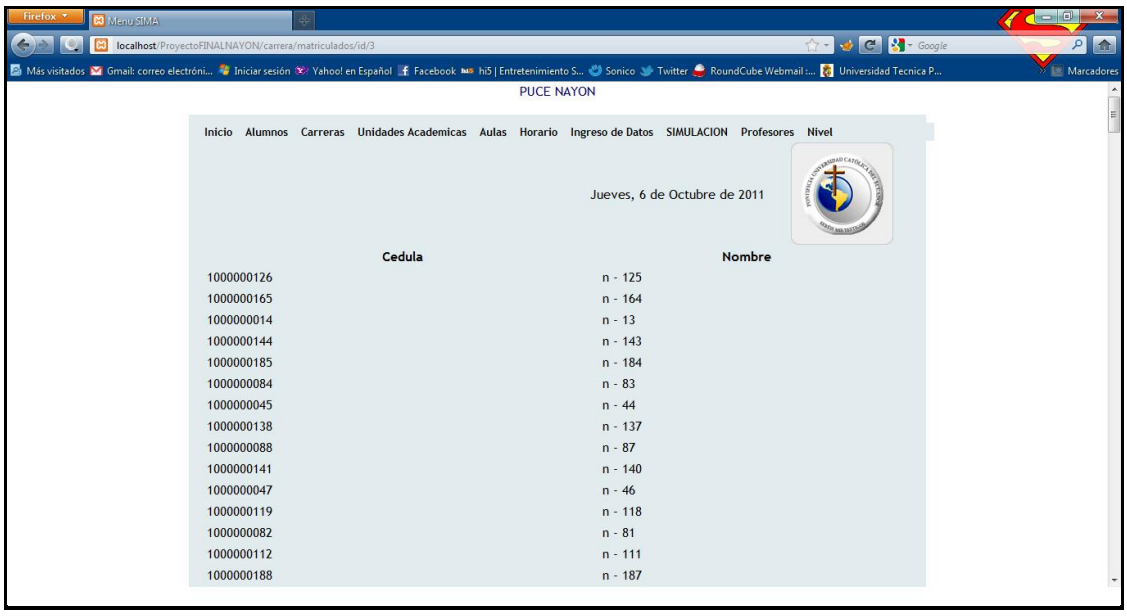


Fuente: http://localhost/NAYONV4/

4.2.3.4. ALUMNOS EN CARRERAS

Al volver a Carreras y dar clic en alumnos, se tiene los alumnos que toman esa carrera:

Figura V-13: Pantalla Alumnos de una determinada Carrera**.**

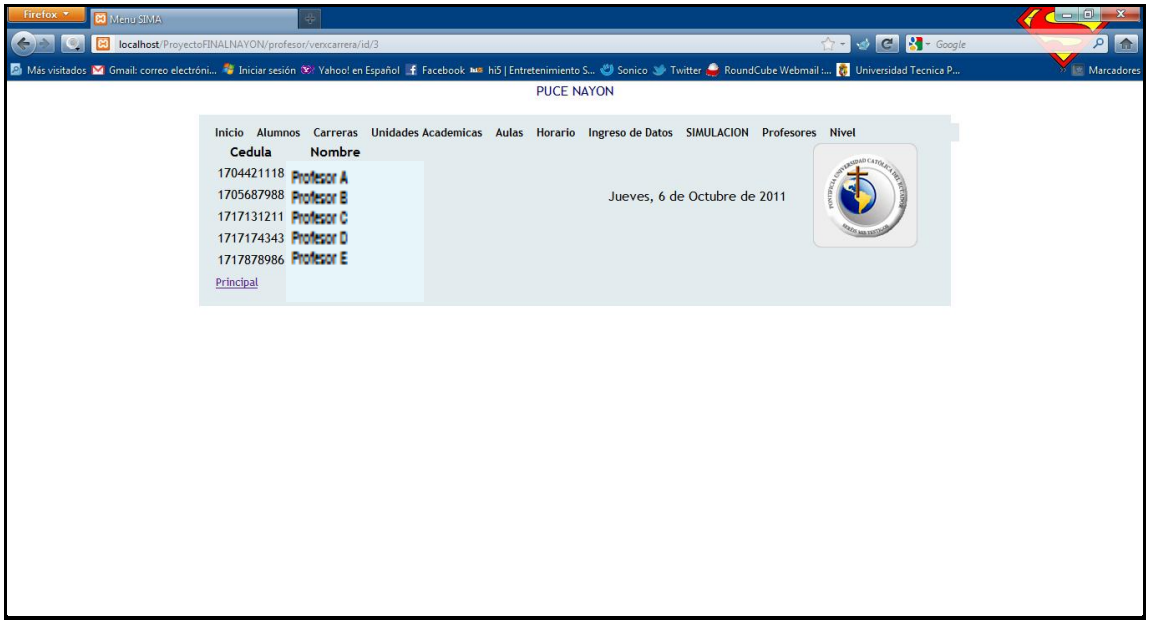


Fuente: http://localhost/NAYONV4/

4.2.3.5. PROFESORES EN CARRERAS

Al dar clic en profesores, se tiene los profesores de la carrera:

Figura V-14: Pantalla Profesores de una determinada Carrera**.**



Fuente: http://localhost/NAYONV4/

4.2.4. ADMINISTRACIÓN DE DEPARTAMENTOS

Al dar clic en departamentos, se tiene lo siguiente:

Figura V-15: Pantalla Administración de Departamentos**.**



Fuente: http://localhost/NAYONV4/

4.2.4.1. CARRERAS POR DEPARTAMENTOS

Al seleccionar un determinado departamento (para el ejemplo Ingeniería), y dar clic se muestra lo siguiente:

Figura V-16: Pantalla Selección de Opciones de un determinado Departamento**.**



Fuente: http://localhost/NAYONV4/

4.2.4.2. ASIGNATURAS – CARRERA -DEPARTAMENTO

Al seleccionar el link materias del departamento deseado (para el ejemplo Ingeniería) y se obtiene sus diferentes asignaturas:

Figura V-17: Pantalla Asignaturas de un determinado Departamento**.**



Fuente: http://localhost/NAYONV4/

4.2.4.3. ALUMNOS FORÁNEOS DE DEPARTAMENTOS

Una vez que se regresa a departamentos se debe dar clic en Foráneos. Dando clic en Foráneos se visualiza los alumnos de otros departamentos que toman materias en el departamento seleccionado previamente (para el ejemplo Ingeniería):

Figura V-18: Pantalla Alumnos Foráneos en un determinado Departamento**.**

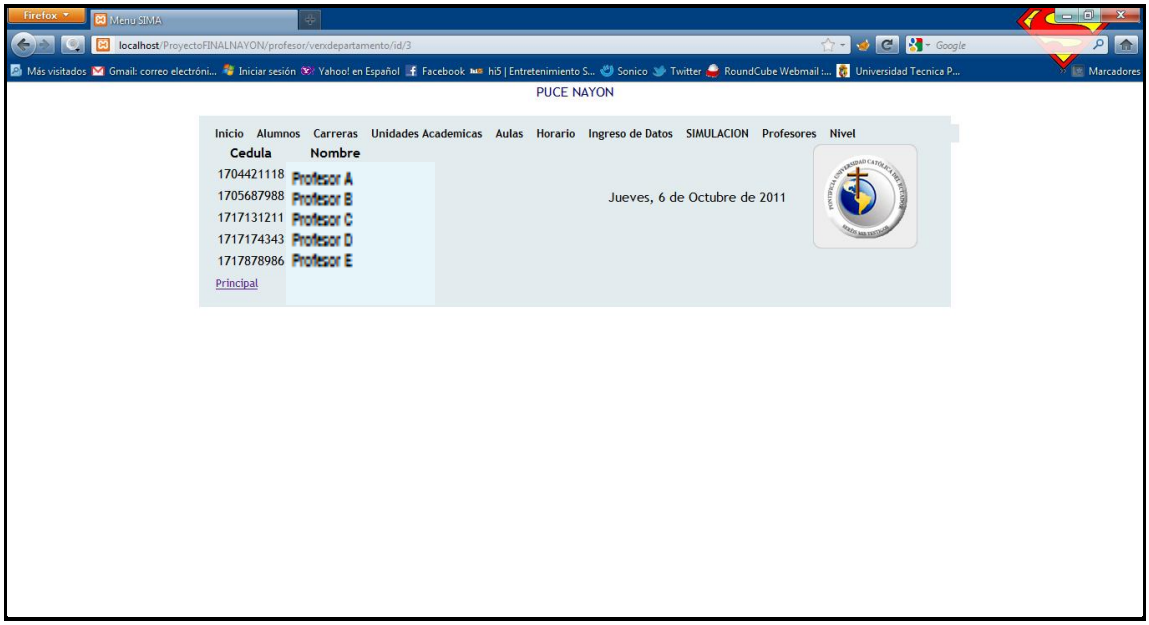


Fuente: http://localhost/NAYONV4/

4.2.4.4. PROFESORES EN DEPARTAMENTOS

Al dar clic en departamentos, se selecciona alguna unidad académica y al dar clic en profesor se muestra los profesores en el departamento:

Figura V-19: Pantalla Profesores de un determinado Departamento**.**

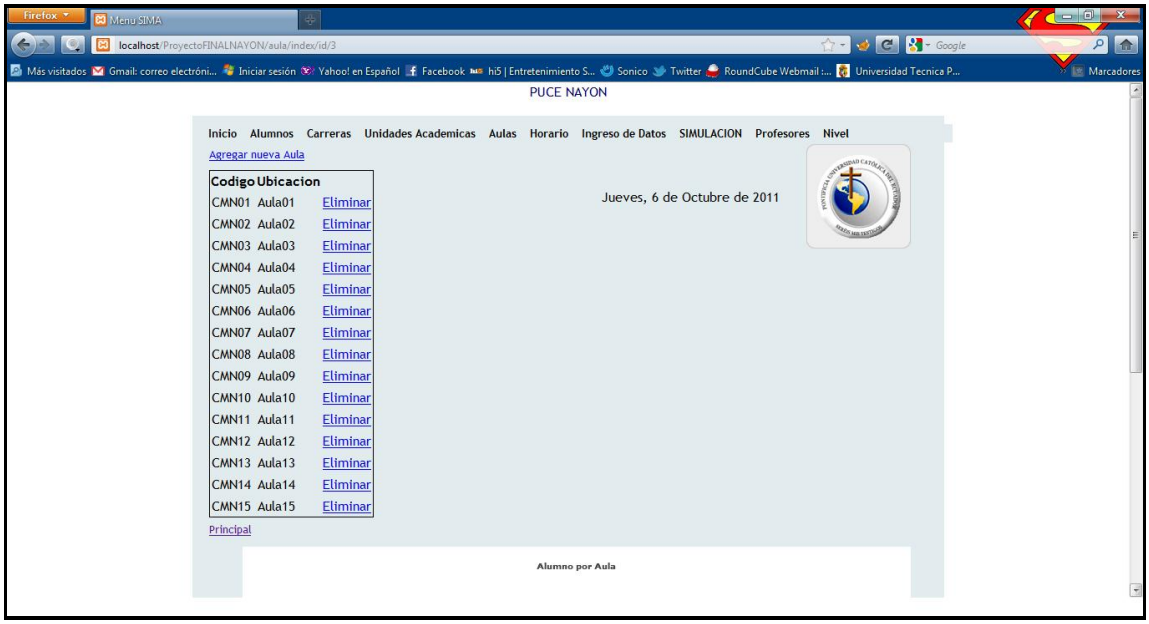


Fuente: http://localhost/NAYONV4/

4.2.5. ADMINISTRACIÓN DE AULAS

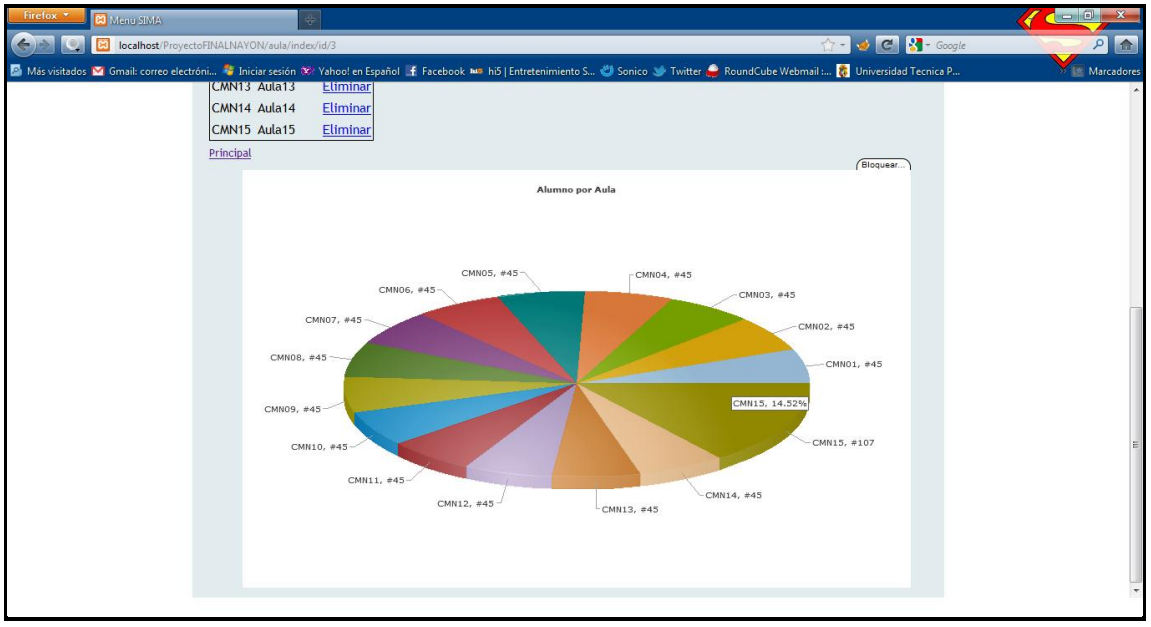
Al dar clic en aulas se tiene las aulas creadas en la simulación y adicionalmente la distribución de los estudiantes por aula:

Figura V-20: Pantalla Aulas Creadas**.**



Fuente: http://localhost/NAYONV4/

Figura V-21: Distribución de Estudiantes por Aula**.**



Fuente: http://localhost/NAYONV4/

4.2.6. ADMINISTRACIÓN DE HORARIOS

Al ir al menú principal se selecciona horario, con lo cual se logra visualizar los horarios, en esta ventana se puede ver un gráfico de los alumnos de dichos horarios:

Figura V-22: Pantalla de Horarios y Distribución de Estudiantes por Horario**.**

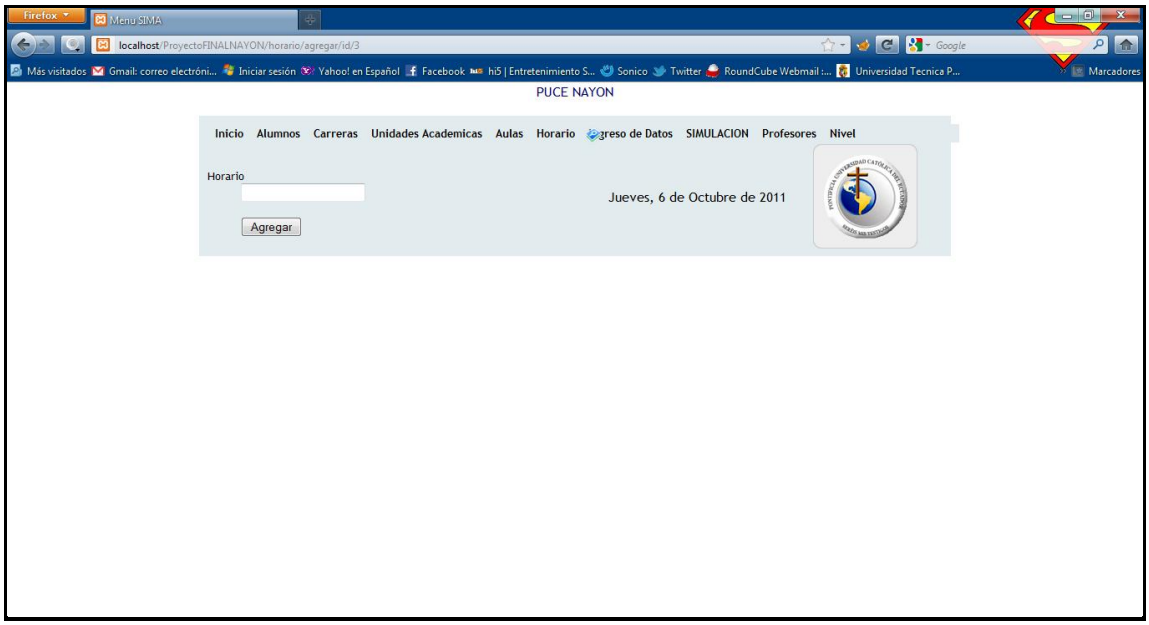


Fuente: http://localhost/NAYONV4/

4.2.6.1. NUEVO HORARIO

Adicionalmente al dar clic en Agregar nuevo horario, es posible agregar más horarios:

Figura V-23: Pantalla Agregar Horario**.**



Fuente: http://localhost/NAYONV4/

4.2.7. ADMINISTRACIÓN DE PROFESORES

Al dar clic en profesores se tiene la siguiente ventana:

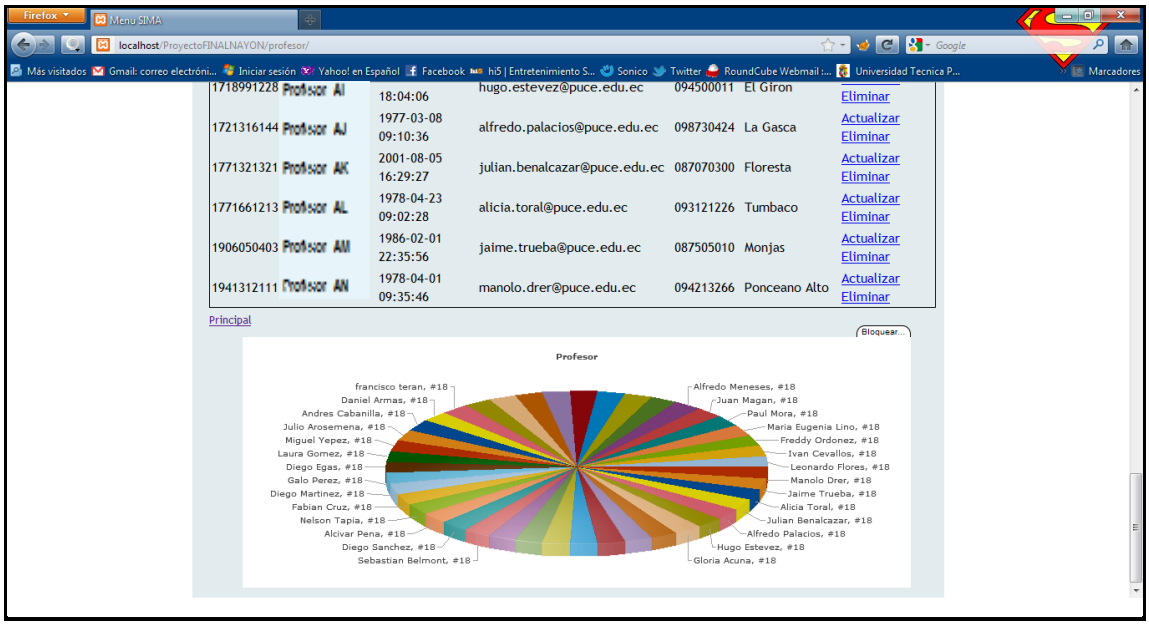
Figura V-24: Pantalla Administración de Profesores**.**



Fuente: http://localhost/NAYONV4/

En la parte de abajo se muestra un gráfico con los profesores asignados:

Figura V-25: Distribución de Profesores Asignados**.**



Fuente: http://localhost/NAYONV4/

4.2.8. ADMINISTRACIÓN DE NIVELES

Al dar clic en menú nivel, se tiene todos los niveles que han sido registrados y al final de la pantalla un gráfico estadístico con los alumnos por nivel:

Figura V-26: Pantalla Administración de Niveles y Distribución de Niveles por Alumno**.**

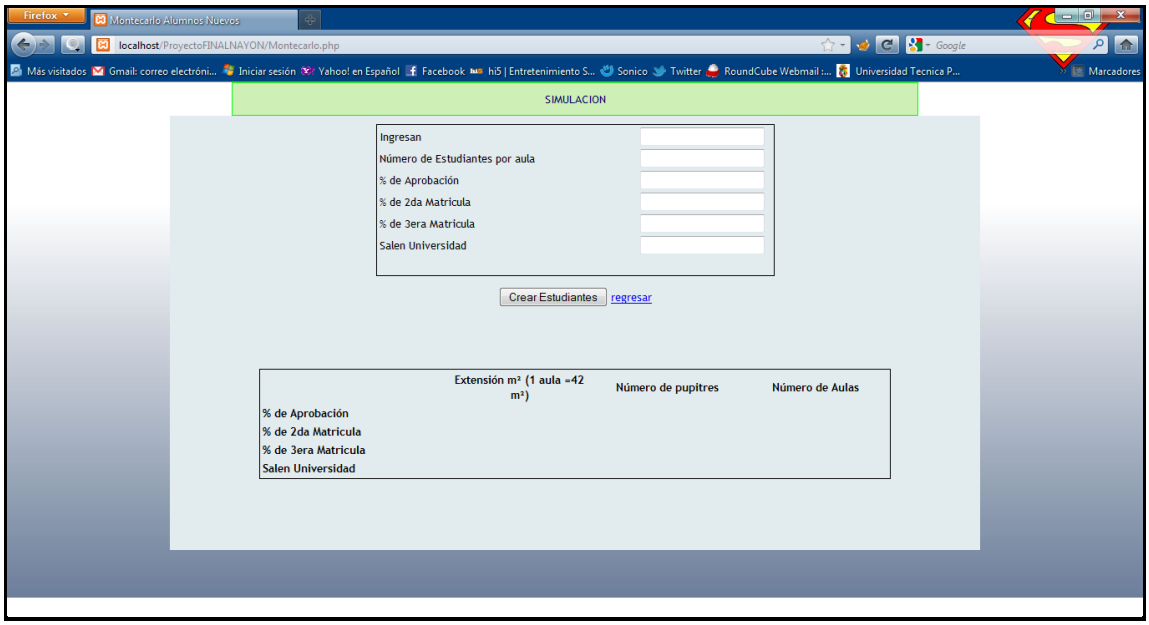


Fuente: http://localhost/NAYONV4/

4.2.9. PROCESO DE SIMULACIÓN

Al dar clic en el menú simulación, se tiene un menú para hacer un cálculo de alumnos por aula para casos generales:

Figura V-27: Pantalla Simulación**.**

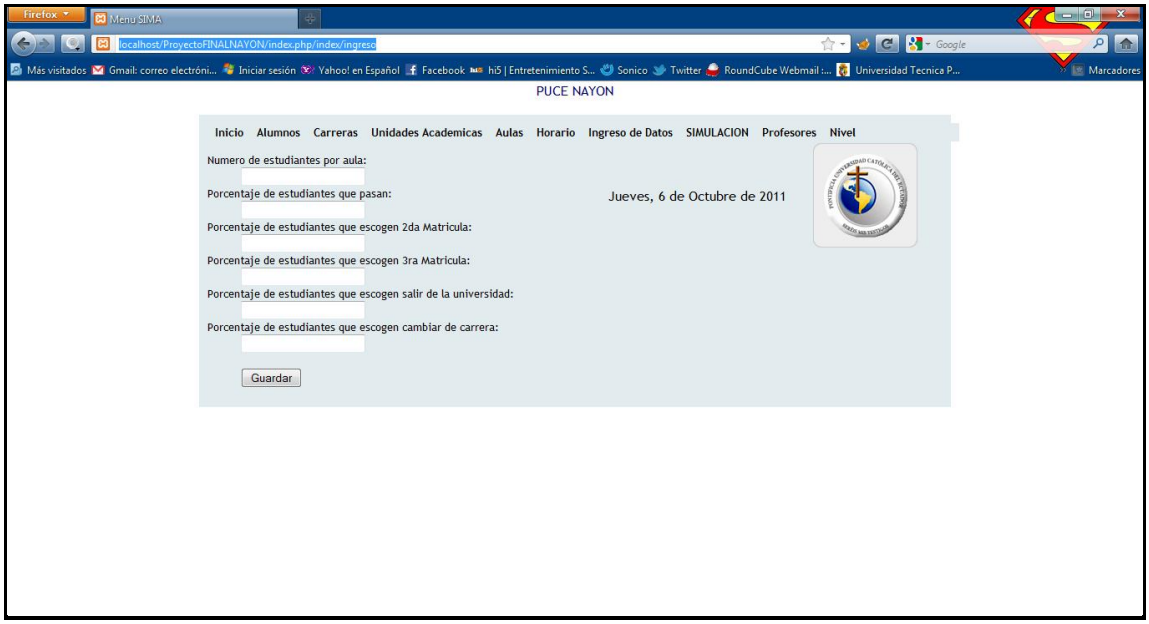


Fuente: http://localhost/NAYONV4/

4.2.10. PROCESO DE INGRESO DE DATOS

Al dar clic en menú ingreso de datos se tiene la siguiente ventana:

Figura V-28: Pantalla Ingreso de Datos**.**



Fuente: http://localhost/NAYONV4/

4.2.11. HISTORIAL ACADÉMICO

Al dirigirse hacia el menú alumno y seleccionar historial académico, se visualiza la siguiente ventana:

Figura V-29: Submenú Historial Académico.

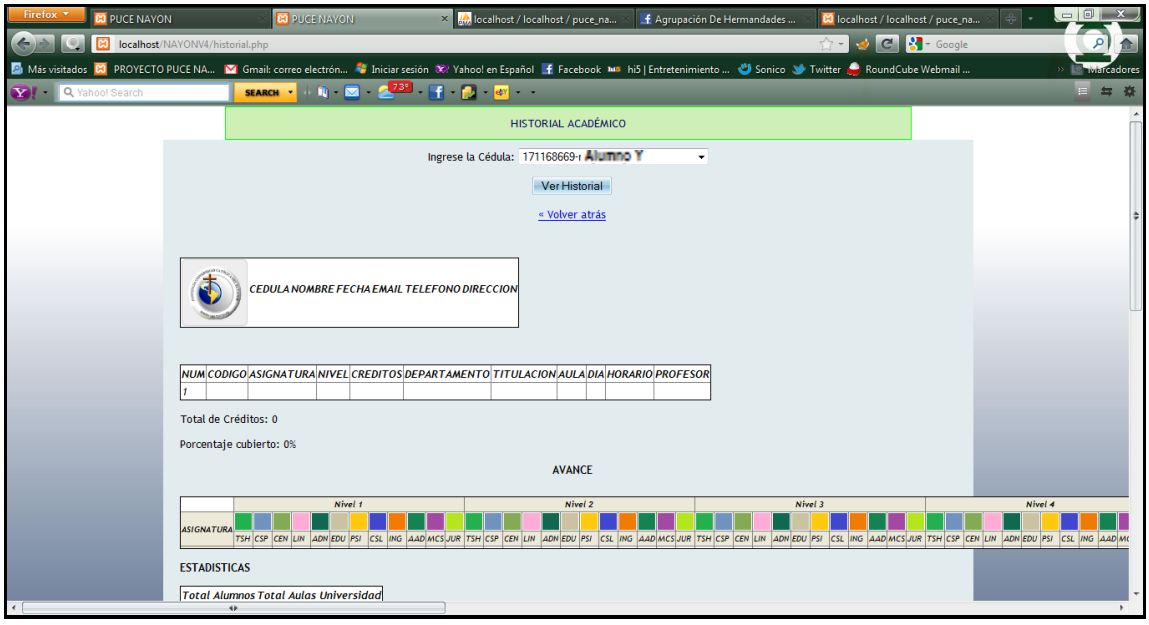


Fuente: http://localhost/NAYONV4/

4.2.11.1. BÚSQUEDA EN HISTORIAL ACADÉMICO

Una vez seleccionado el submenú historial académico aparecerá la siguiente ventana:

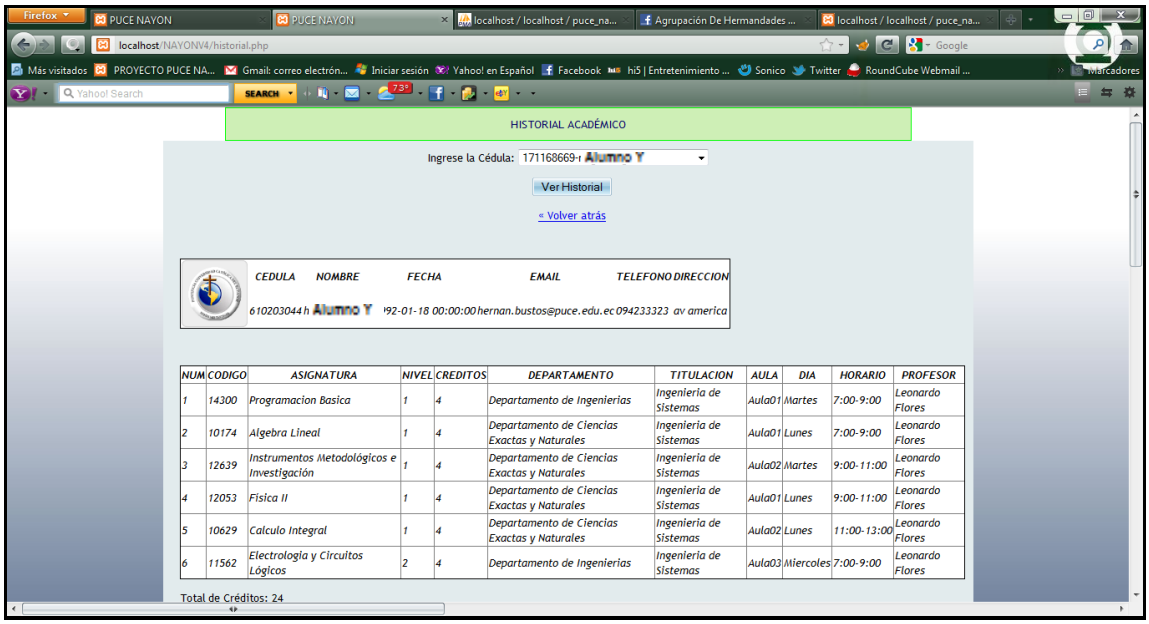
Figura V-30: Pantalla Historial Académico**.**



Fuente: http://localhost/NAYONV4/

Al seleccionar un estudiante (como primer ejemplo Alumno Y, estudiante cursando una determinada carrera), se muestra lo siguiente:

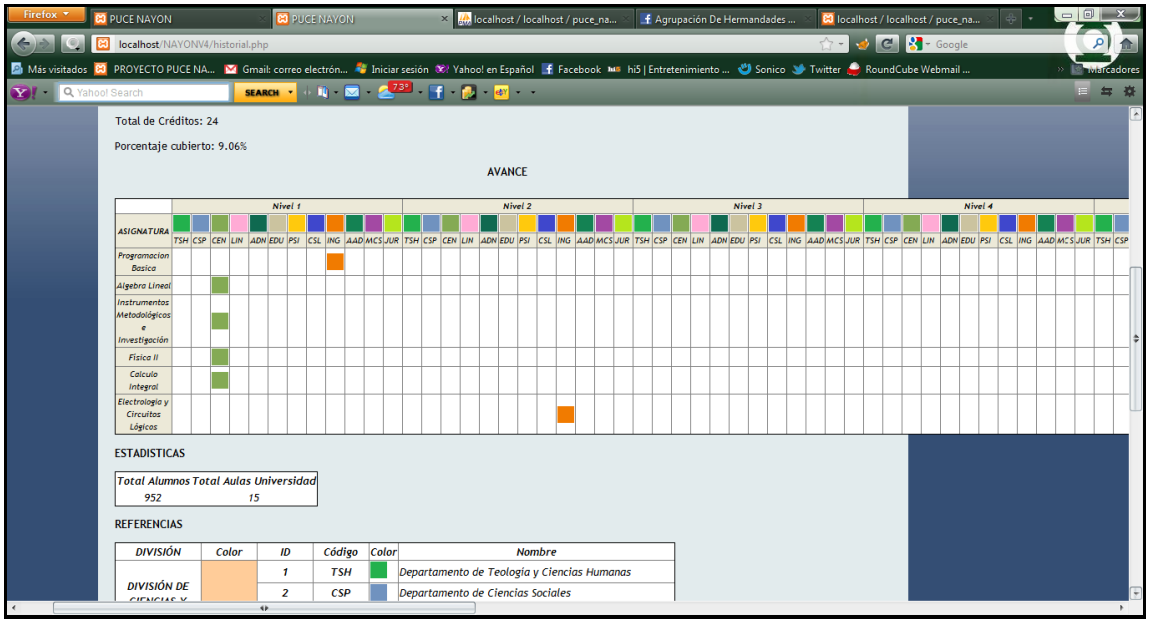
Figura V-31: Pantalla Historial Académico de Estudiante seleccionado (Ejemplo 1).



Fuente: http://localhost/NAYONV4/

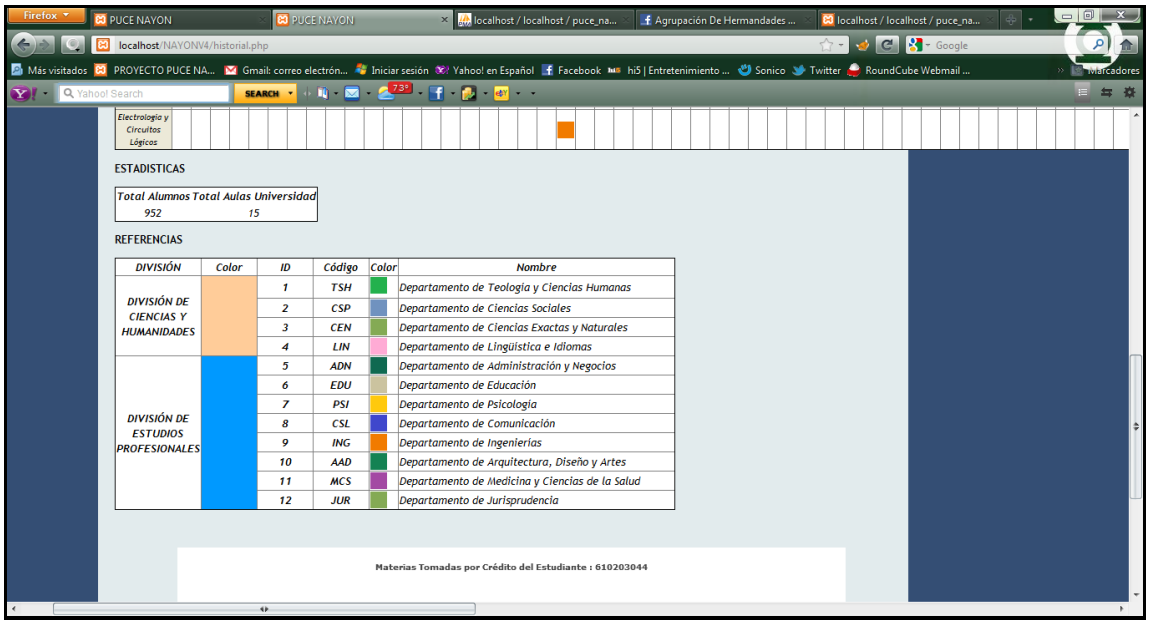
Aparecerá el kardex de avance de las asignaturas y su avance de manera gráfica:

Figura V-32: Kardex de Estudiante seleccionado (Ejemplo 1)**.**



Fuente: http://localhost/NAYONV4/

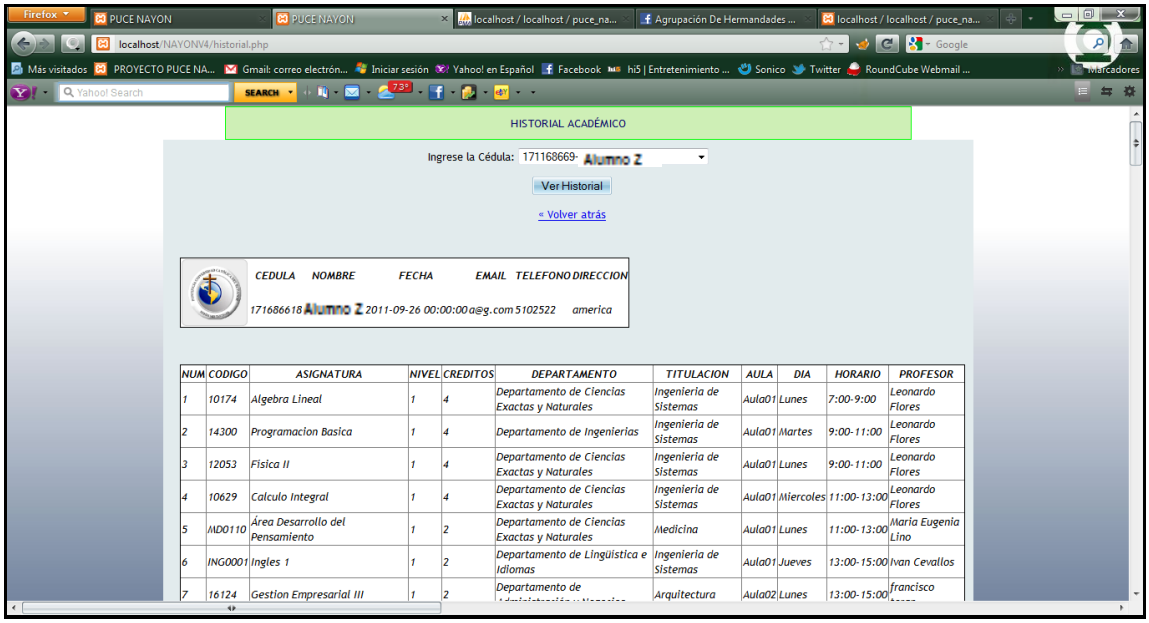
Figura V-33: Referencias de Departamentos de Estudiante seleccionado (Ejemplo 1).



Fuente: http://localhost/NAYONV4/

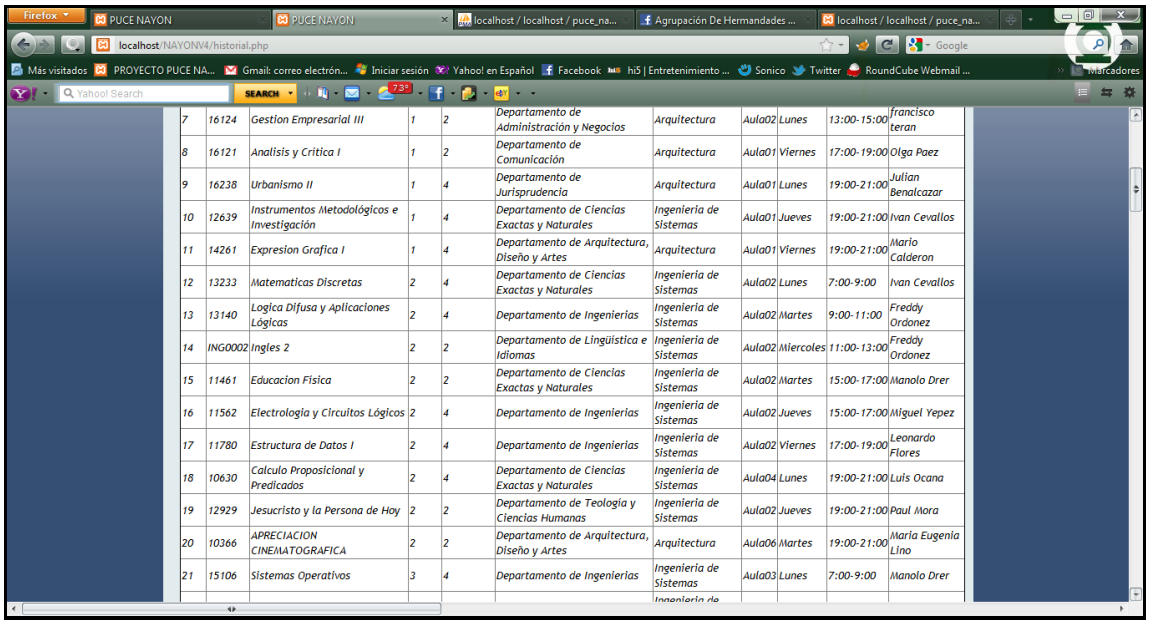
Si se selecciona otro estudiante (como segundo ejemplo Alumno Z, estudiante que terminó la carrera de Ingeniería en Sistemas) se tiene lo siguiente:

Figura V-34: Pantalla 1 Historial Académico de Estudiante seleccionado (Ejemplo 2).



Fuente: http://localhost/NAYONV4/

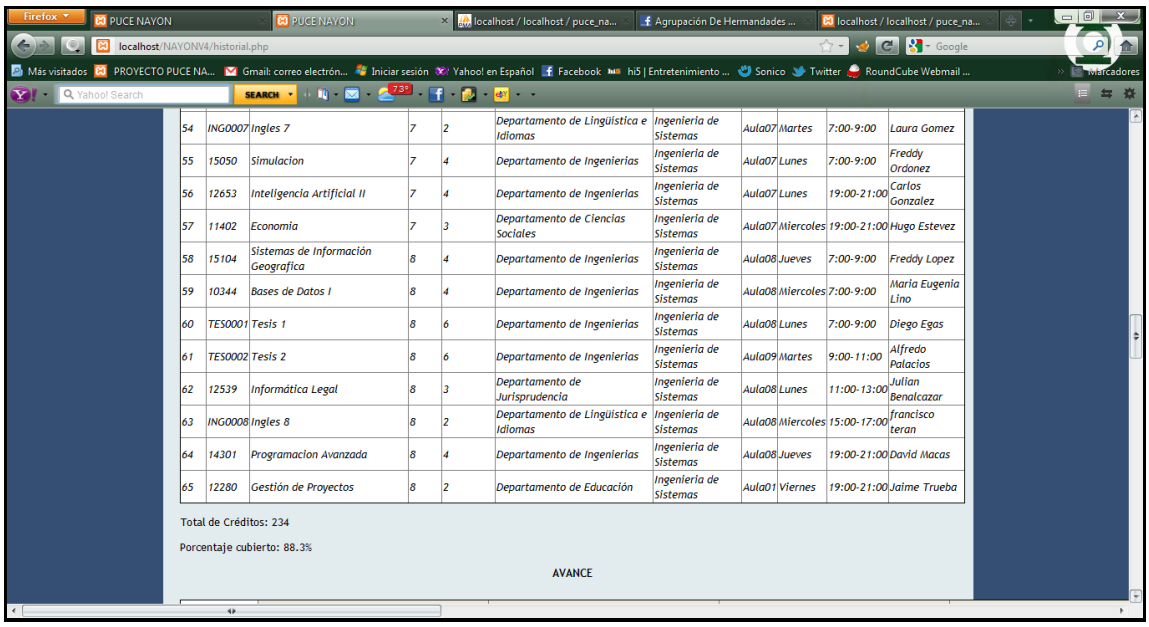
Figura V-35: Pantalla 2 Historial Académico de Estudiante seleccionado (Ejemplo2).



7	16124	Gestion Empresarial III	1	2	Departamento de Administración y Negocios	Arquitectura	Aula02	Lunes	13:00-15:00	francisco teran
8	16121	Analisis y Critica I	1	2	Departamento de Comunicación	Arquitectura	Aula01	Viernes	17:00-19:00	Olga Paez
9	16238	Urbanismo II	1	4	Departamento de Jurisprudencia	Arquitectura	Aula01	Lunes	19:00-21:00	Julian Benalcazar
10	12639	Instrumentos Metodológicos e Investigación	1	4	Departamento de Ciencias Exactas y Naturales	Ingeniería de Sistemas	Aula01	Jueves	19:00-21:00	Ivan Cevallos
11	14261	Expresion Grafica I	1	4	Departamento de Arquitectura, Diseño y Artes	Arquitectura	Aula01	Viernes	19:00-21:00	Mario Calderon
12	13233	Matematicas Discretas	2	4	Departamento de Ciencias Exactas y Naturales	Ingeniería de Sistemas	Aula02	Lunes	7:00-9:00	Ivan Cevallos
13	13140	Logica Difusa y Aplicaciones Lógicas	2	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula02	Martes	9:00-11:00	Freddy Ordonez
14	ING0002	Ingles 2	2	2	Departamento de Lingüística e Idiomas	Ingeniería de Sistemas	Aula02	Miercoles	11:00-13:00	Freddy Ordonez
15	11461	Educacion Fisica	2	2	Departamento de Ciencias Exactas y Naturales	Ingeniería de Sistemas	Aula02	Martes	15:00-17:00	Manolo Drer
16	11562	Electrologia y Circuitos Lógicos	2	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula02	Jueves	15:00-17:00	Miguel Yepez
17	11780	Estructura de Datos I	2	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula02	Viernes	17:00-19:00	Leonardo Flores
18	10630	Calculo Proposicional y Predicados	2	4	Departamento de Ciencias Exactas y Naturales	Ingeniería de Sistemas	Aula04	Lunes	19:00-21:00	Luis Ocana
19	12929	Jesucristo y la Persona de Hoy	2	2	Departamento de Teologia y Ciencias Humanas	Ingeniería de Sistemas	Aula02	Jueves	19:00-21:00	Paul Mora
20	10366	APRECIACION CINEMATOGRAFICA	2	2	Departamento de Arquitectura, Diseño y Artes	Arquitectura	Aula06	Martes	19:00-21:00	Maria Eugenia Lino
21	15106	Sistemas Operativos	3	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula03	Lunes	7:00-9:00	Manolo Drer

Fuente: http://localhost/NAYONV4/

Figura V-36: Pantalla 3 Historial Académico de Estudiante seleccionado (Ejemplo 2).



54	ING0007	Ingles 7	7	2	Departamento de Lingüística e Idiomas	Ingeniería de Sistemas	Aula07	Martes	7:00-9:00	Laura Gomez
55	15050	Simulacion	7	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula07	Lunes	7:00-9:00	Freddy Ordonez
56	12653	Inteligencia Artificial II	7	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula07	Lunes	19:00-21:00	Carlos Gonzalez
57	11402	Economia	7	3	Departamento de Ciencias Sociales	Ingeniería de Sistemas	Aula07	Miercoles	19:00-21:00	Hugo Estevez
58	15104	Sistemas de Información Geografica	8	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula08	Jueves	7:00-9:00	Freddy Lopez
59	10344	Bases de Datos I	8	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula08	Miercoles	7:00-9:00	Maria Eugenia Lino
60	TES0001	Tesis 1	8	6	Departamento de Ingenierias	Ingeniería de Sistemas	Aula08	Lunes	7:00-9:00	Diego Egas
61	TES0002	Tesis 2	8	6	Departamento de Ingenierias	Ingeniería de Sistemas	Aula09	Martes	9:00-11:00	Alfredo Palacios
62	12539	Informática Legal	8	3	Departamento de Jurisprudencia	Ingeniería de Sistemas	Aula08	Lunes	11:00-13:00	Julian Benalcazar
63	ING0008	Ingles 8	8	2	Departamento de Lingüística e Idiomas	Ingeniería de Sistemas	Aula08	Miercoles	15:00-17:00	francisco teran
64	14301	Programacion Avanzada	8	4	Departamento de Ingenierias	Ingeniería de Sistemas	Aula08	Jueves	19:00-21:00	David Macas
65	12280	Gestión de Proyectos	8	2	Departamento de Educación	Ingeniería de Sistemas	Aula01	Viernes	19:00-21:00	Jaime Trueba

Total de Créditos: 234

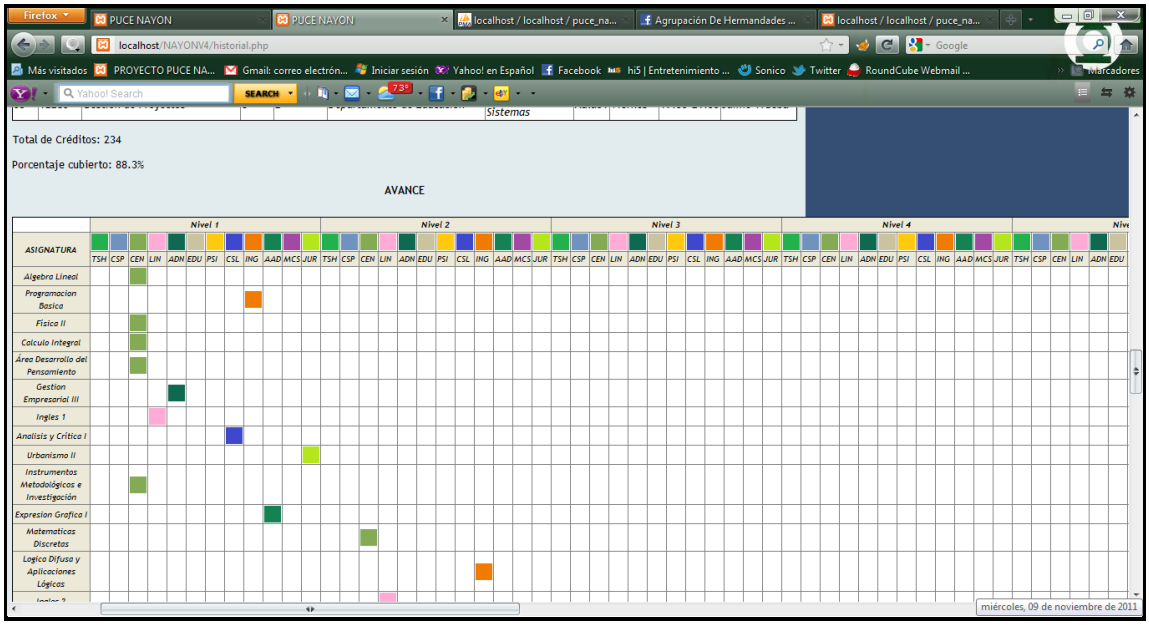
Porcentaje cubierto: 88.3%

AVANCE

Fuente: http://localhost/NAYONV4/

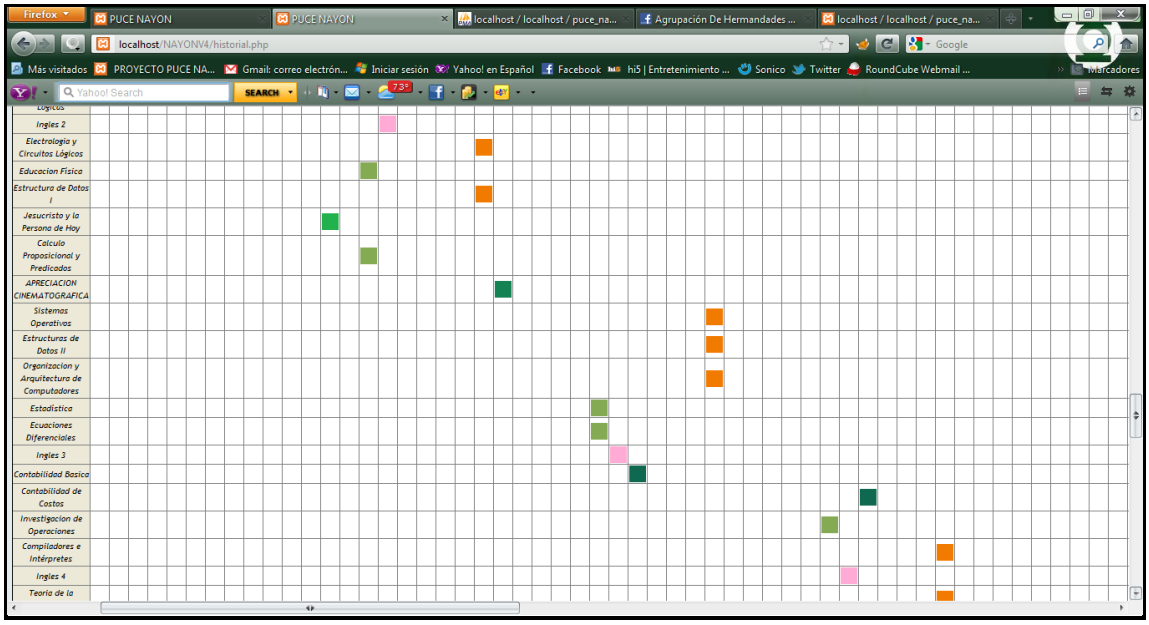
Gráficamente se representa el avance del alumno:

Figura V-37: Pantalla 1 Kardex de Estudiante seleccionado (Ejemplo 2).



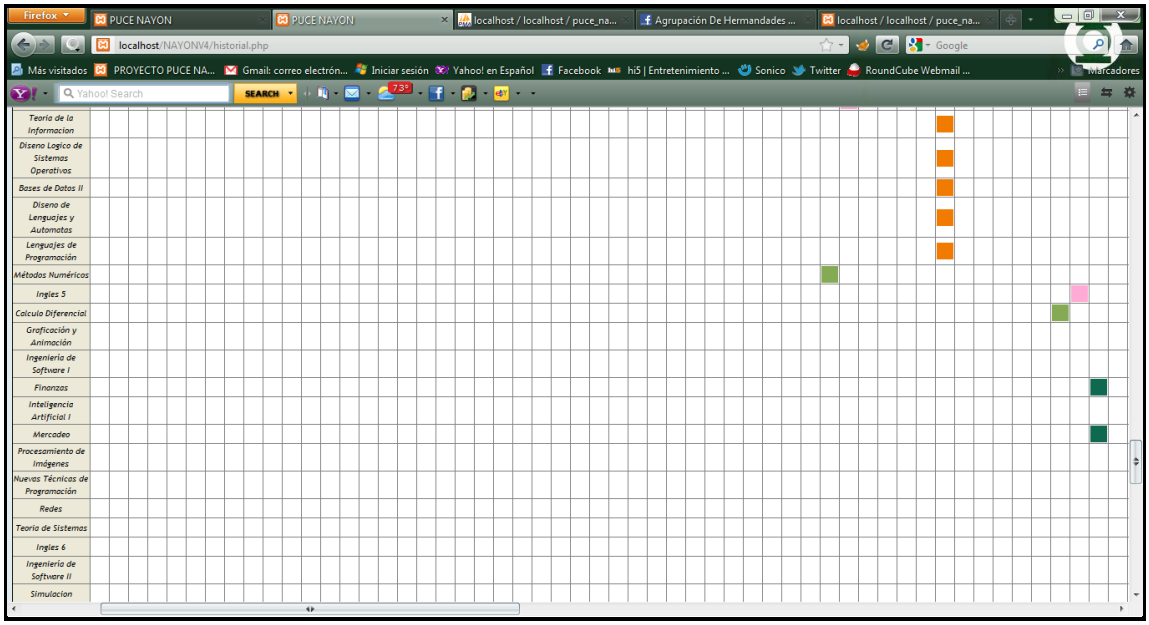
Fuente: http://localhost/NAYONV4/

Figura V-38: Pantalla 2 Kardex de Estudiante seleccionado (Ejemplo 2).



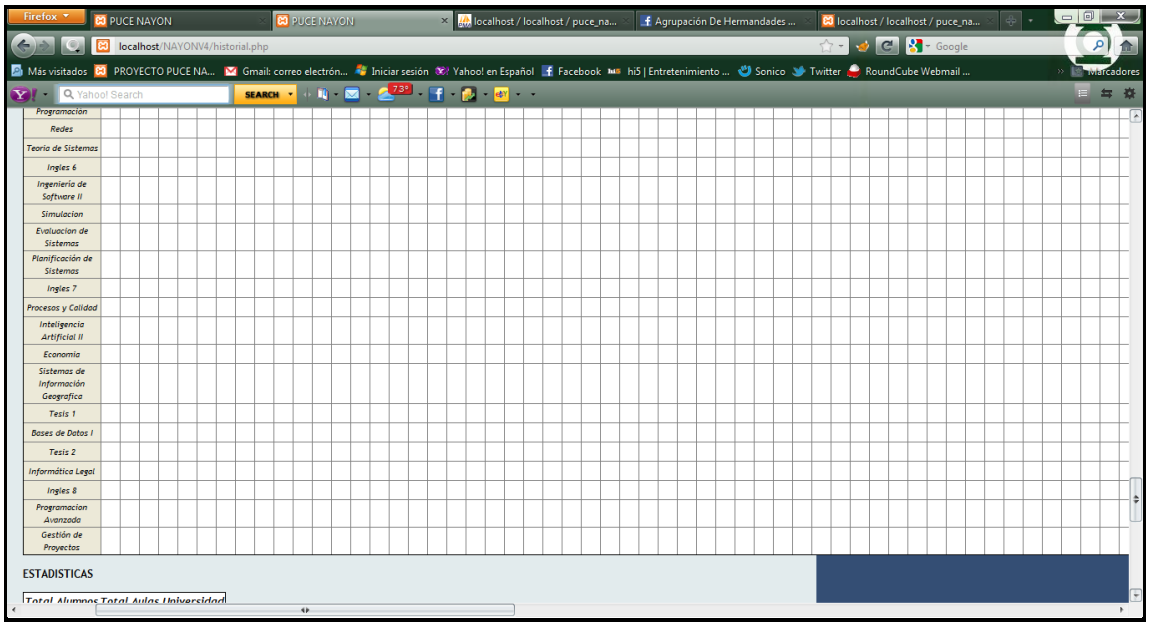
Fuente: http://localhost/NAYONV4/

Figura V-39: Pantalla 3 Kardex de Estudiante seleccionado (Ejemplo 2).



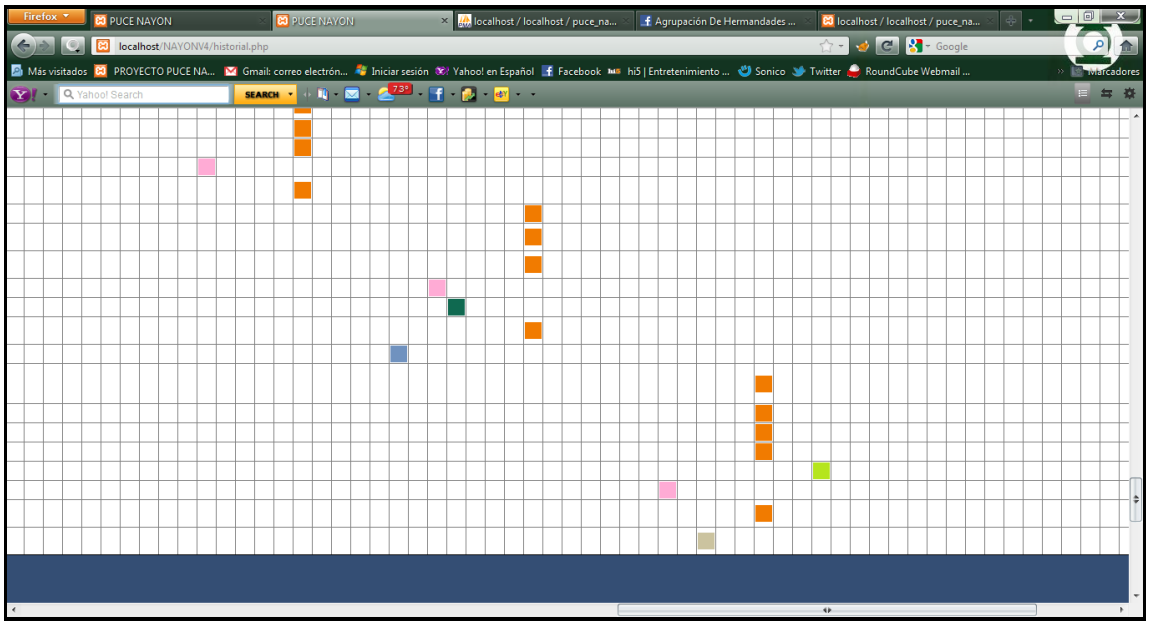
Fuente: http://localhost/NAYONV4/

Figura V-40: Pantalla 4 Kardex de Estudiante seleccionado (Ejemplo 2).



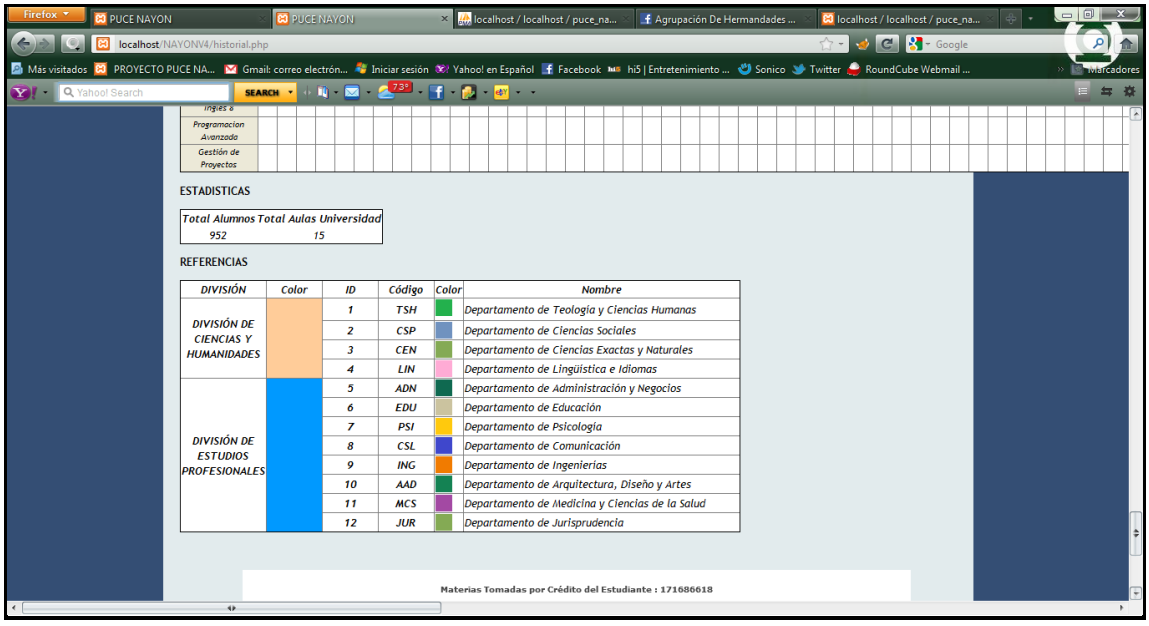
Fuente: http://localhost/NAYONV4/

Figura V-41: Pantalla 5 Kardex de Estudiante seleccionado (Ejemplo 2).



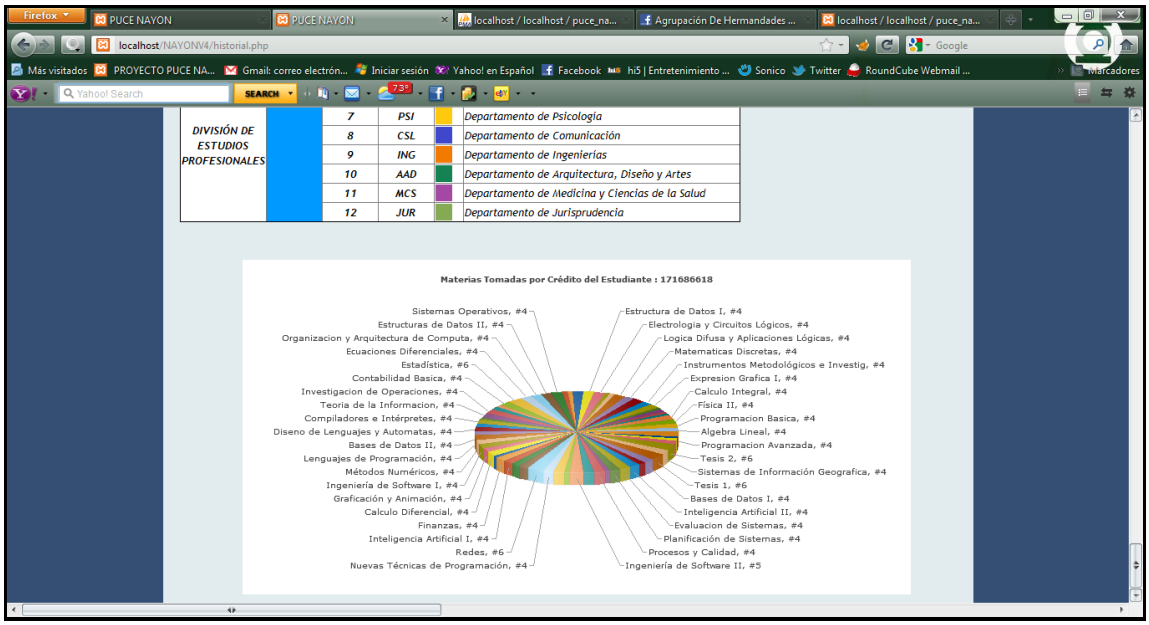
Fuente: http://localhost/NAYONV4/

Figura V-42: Referencias de Departamentos de Estudiante seleccionado (Ejemplo 2).



Fuente: http://localhost/NAYONV4/

Figura V-43: Créditos de Materias Tomadas del Estudiante seleccionado (Ejemplo 2).



Fuente: http://localhost/NAYONV4/

Al dar clic en el link volver atrás se logra regresar al menú principal.

El proceso que simula la manera en que un alumno avanza cada nivel hasta culminar su carrera académica, es generado con la ejecución de un procedure, en el cual se hace uso de tablas temporales, que en sí solo sirven para realizar consultas u operaciones con la base.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `matricularcarr`( )
BEGIN

DECLARE idAlumno_ingenieria, idCarrera_ingenieria int;
DECLARE maxAlumnos_ingenieria double;

DECLARE idAlumno_arquitectura, idCarrera_arquitectura int;
DECLARE maxAlumnos_arquitectura double;

DECLARE idAlumno_administracion, idCarrera_administracion int;
DECLARE maxAlumnos_administracion double;

DECLARE idAlumno_economia, idCarrera_economia int;
DECLARE maxAlumnos_economia double;

DECLARE idAlumno_medicina , idCarrera_medicina  int;
```

```
DECLARE maxAlumnos_medicina  double;

  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_ingenieria
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_ingenieria
(
id int(11)
);
delete from nuevos_ingenieria;
delete from carreras_ingenieria;

  insert into nuevos_ingenieria (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_ingenieria (id) select distinct ID from
carrera where NOMBRE like '%ingenieria%';

  select count(*) into maxAlumnos_ingenieria from
nuevos_ingenieria where matriculado = 0;

SET maxAlumnos_ingenieria=maxAlumnos_ingenieria*0.13;
  WHILE maxAlumnos_ingenieria > 1 DO

   select count(*) into maxAlumnos_ingenieria from
nuevos_ingenieria where matriculado = 0;

  select cedula into idAlumno_ingenieria from nuevos_ingenieria
order by RAND() limit 1;
  select id into idCarrera_ingenieria from carreras_ingenieria
order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_ingenieria,idCarrera_ingenieria,1);

  delete from nuevos_ingenieria where cedula =
idAlumno_ingenieria;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_ingenieria;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_arquitectura
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_arquitectura
(
id int(11)
);
delete from nuevos_arquitectura;
```

```
delete from carreras_arquitectura;

  insert into nuevos_arquitectura (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_arquitectura (id) select distinct ID from
carrera where NOMBRE  like '%arquitectura%';

  select count(*) into maxAlumnos_arquitectura from
nuevos_arquitectura where matriculado = 0;

SET maxAlumnos_arquitectura=maxAlumnos_arquitectura*0.124;
  WHILE maxAlumnos_arquitectura > 1 DO

  select count(*) into maxAlumnos_arquitectura from
nuevos_arquitectura where matriculado = 0;

  select cedula into idAlumno_arquitectura from
nuevos_arquitectura order by RAND() limit 1;
  select id into idCarrera_arquitectura from
carreras_arquitectura order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_arquitectura,idCarrera_arquitectura,1);

  delete from nuevos_arquitectura where cedula =
idAlumno_arquitectura;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_arquitectura;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_administracion
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_administracion
(
id int(11)
);
delete from nuevos_administracion;
delete from carreras_administracion;

  insert into nuevos_administracion (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_administracion (id) select distinct ID
from carrera where NOMBRE like '%administracion%';

  select count(*) into maxAlumnos_administracion from
nuevos_administracion where matriculado = 0;

SET maxAlumnos_administracion=maxAlumnos_administracion*0.317;
  WHILE maxAlumnos_administracion > 1 DO
```

```
  select count(*) into maxAlumnos_administracion from
nuevos_administracion where matriculado = 0;

  select cedula into idAlumno_administracion from
nuevos_administracion order by RAND() limit 1;
  select id into idCarrera_administracion from
carreras_administracion order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_administracion,idCarrera_administracion,1);

  delete from nuevos_administracion where cedula =
idAlumno_administracion;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_administracion;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_economia
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_economia
(
id int(11)
);
delete from nuevos_economia;
delete from carreras_economia;

  insert into nuevos_economia (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_economia (id) select distinct ID from
carrera where NOMBRE like '%economia%';

  select count(*) into maxAlumnos_economia from nuevos_economia
where matriculado = 0;

SET maxAlumnos_economia=maxAlumnos_economia*0.218;
  WHILE maxAlumnos_economia > 1 DO

  select count(*) into maxAlumnos_economia from nuevos_economia
where matriculado = 0;

  select cedula into idAlumno_economia from nuevos_economia
order by RAND() limit 1;
  select id into idCarrera_economia from carreras_economia order
by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_economia,idCarrera_economia,1);
```

```
  delete from nuevos_economia where cedula = idAlumno_economia;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_economia;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_medicina
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_medicina
(
id int(11)
);
delete from nuevos_medicina ;
delete from carreras_medicina ;

  insert into nuevos_medicina  (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_medicina  (id) select distinct ID from
carrera where NOMBRE like '%medicina%';

  select count(*) into maxAlumnos_medicina  from nuevos_medicina
where matriculado = 0;

SET maxAlumnos_medicina =maxAlumnos_medicina *0.21;
  WHILE maxAlumnos_medicina  > 1 DO

  select count(*) into maxAlumnos_medicina  from nuevos_medicina
where matriculado = 0;

  select cedula into idAlumno_medicina  from nuevos_medicina
order by RAND() limit 1;
  select id into idCarrera_medicina  from carreras_medicina
order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_medicina ,idCarrera_medicina ,1);

  delete from nuevos_medicina  where cedula = idAlumno_medicina;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_medicina;

  END WHILE;

END$$
```

Todas las tablas descritas como temporales, nos ayudan a realizar el avance entre cada semestre o nivel de estudio, para la simulación del progreso de un estudiante:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `matricularcarr1`( )
BEGIN

DECLARE idAlumno_ingenieria, idCarrera_ingenieria int;
DECLARE maxAlumnos_ingenieria double;

DECLARE idAlumno_arquitectura, idCarrera_arquitectura int;
DECLARE maxAlumnos_arquitectura double;

DECLARE idAlumno_administracion, idCarrera_administracion int;
DECLARE maxAlumnos_administracion double;

DECLARE idAlumno_economia, idCarrera_economia int;
DECLARE maxAlumnos_economia double;

DECLARE idAlumno_medicina , idCarrera_medicina  int;
DECLARE maxAlumnos_medicina  double;

  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_ingenieria
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_ingenieria
(
id int(11)
);
delete from nuevos_ingenieria;
delete from carreras_ingenieria;

  insert into nuevos_ingenieria (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_ingenieria (id) select distinct ID from
carrera where NOMBRE like '%ingenieria%';

  select count(*) into maxAlumnos_ingenieria from
nuevos_ingenieria where matriculado = 0;

SET maxAlumnos_ingenieria=maxAlumnos_ingenieria*0.13;
  WHILE maxAlumnos_ingenieria > 1 DO

   select count(*) into maxAlumnos_ingenieria from
nuevos_ingenieria where matriculado = 0;

  select cedula into idAlumno_ingenieria from nuevos_ingenieria
order by RAND() limit 1;
  select id into idCarrera_ingenieria from carreras_ingenieria
order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_ingenieria,idCarrera_ingenieria,1);

  delete from nuevos_ingenieria where cedula =
idAlumno_ingenieria;
```

```
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_ingenieria;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_arquitectura
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_arquitectura
(
id int(11)
);
delete from nuevos_arquitectura;
delete from carreras_arquitectura;

  insert into nuevos_arquitectura (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_arquitectura (id) select distinct ID from
carrera where NOMBRE  like '%arquitectura%';

  select count(*) into maxAlumnos_arquitectura from
nuevos_arquitectura where matriculado = 0;

SET maxAlumnos_arquitectura=maxAlumnos_arquitectura*0.124;
  WHILE maxAlumnos_arquitectura > 1 DO

  select count(*) into maxAlumnos_arquitectura from
nuevos_arquitectura where matriculado = 0;

  select cedula into idAlumno_arquitectura from
nuevos_arquitectura order by RAND() limit 1;
  select id into idCarrera_arquitectura from
carreras_arquitectura order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_arquitectura,idCarrera_arquitectura,1);

  delete from nuevos_arquitectura where cedula =
idAlumno_arquitectura;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_arquitectura;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_administracion
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_administracion
```

```
(
id int(11)
);
delete from nuevos_administracion;
delete from carreras_administracion;

  insert into nuevos_administracion (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_administracion (id) select distinct ID
from carrera where NOMBRE like '%administracion%';

  select count(*) into maxAlumnos_administracion from
nuevos_administracion where matriculado = 0;

SET maxAlumnos_administracion=maxAlumnos_administracion*0.317;
  WHILE maxAlumnos_administracion > 1 DO

  select count(*) into maxAlumnos_administracion from
nuevos_administracion where matriculado = 0;

  select cedula into idAlumno_administracion from
nuevos_administracion order by RAND() limit 1;
  select id into idCarrera_administracion from
carreras_administracion order by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_administracion,idCarrera_administracion,1);

  delete from nuevos_administracion where cedula =
idAlumno_administracion;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_administracion;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_economia
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_economia
(
id int(11)
);
delete from nuevos_economia;
delete from carreras_economia;

  insert into nuevos_economia (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_economia (id) select distinct ID from
carrera where NOMBRE like '%economia%';
```

```
  select count(*) into maxAlumnos_economia from nuevos_economia
where matriculado = 0;

SET maxAlumnos_economia=maxAlumnos_economia*0.218;
  WHILE maxAlumnos_economia > 1 DO

  select count(*) into maxAlumnos_economia from nuevos_economia
where matriculado = 0;

  select cedula into idAlumno_economia from nuevos_economia
order by RAND() limit 1;
  select id into idCarrera_economia from carreras_economia order
by RAND() limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_economia,idCarrera_economia,1);

  delete from nuevos_economia where cedula = idAlumno_economia;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_economia;

  END WHILE;


  CREATE TEMPORARY TABLE IF NOT EXISTS nuevos_medicina
(
cedula int(11),
matriculado tinyint(1)
);
  CREATE TEMPORARY TABLE IF NOT EXISTS carreras_medicina
(
id int(11)
);
delete from nuevos_medicina ;
delete from carreras_medicina ;

  insert into nuevos_medicina  (cedula, matriculado) Select
CEDULA,MATRICULADO  from alumno where matriculado = 0;
  insert into carreras_medicina  (id) select distinct ID from
carrera where NOMBRE like '%medicina%';

  select count(*) into maxAlumnos_medicina  from nuevos_medicina
where matriculado = 0;

SET maxAlumnos_medicina =maxAlumnos_medicina *0.21;
  WHILE maxAlumnos_medicina  > 1 DO

  select count(*) into maxAlumnos_medicina  from nuevos_medicina
where matriculado = 0;

  select cedula into idAlumno_medicina  from nuevos_medicina
order by RAND() limit 1;
  select id into idCarrera_medicina  from carreras_medicina
order by RAND() limit 1;
```

```
  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(idAlumno_medicina ,idCarrera_medicina ,1);

  delete from nuevos_medicina  where cedula = idAlumno_medicina;
  update alumno set MATRICULADO = 1 where CEDULA =
idAlumno_medicina;

  END WHILE;

END$$
```

También para simular el pase, se usa el siguiente procedure:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `pasan`(IN niv int )
BEGIN

         DECLARE    id, CedA, IdC, Ced, Carr, idA, maxAlumnos
int;


 CREATE TEMPORARY TABLE IF NOT EXISTS pasan
(
id int(11),
cedula int(11),
id_carrera      int(11)
);

  select count(ID) into Ced from alumno_carrera where NIVEL=niv;


 insert into pasan (id, cedula, id_carrera) Select ID, CEDULA,
ID_CARRERA from alumno_carrera where NIVEL=niv+1;

 select count(*) into maxAlumnos from pasan;


 WHILE maxAlumnos > 1 DO

  select count(*) into maxAlumnos from pasan;

  select id into idA from pasan order by RAND(maxAlumnos*0.99)
limit 1;
  select cedula into  CedA from pasan order by
RAND(maxAlumnos*0.99) limit 1;
 select id_carrera into IdC from pasan order by
RAND(maxAlumnos*0.99) limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(CedA,IdC,niv+1);

  END WHILE;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `pasan1`(IN niv int
)
BEGIN

         DECLARE    id, CedA, CedP, IdC, Ced, Carr, idA, nivel,
maxAlumnos int;


 CREATE TEMPORARY TABLE IF NOT EXISTS pasan1
(
id int(11),
cedula int(11),
id_carrera      int(11)
);

  select count(ID) into Ced from alumno_carrera where NIVEL=niv;


 insert into pasan (id, cedula, id_carrera) Select ID, CEDULA,
ID_CARRERA from alumno_carrera where NIVEL=niv;

 select count(*) into maxAlumnos from pasan;

 set nivel=niv;
 WHILE maxAlumnos > 1 DO

  select count(*) into maxAlumnos from pasan1;

  select id into idA from pasan1 order by RAND(maxAlumnos*0.99)
limit 1;
  select cedula into  CedA from pasan1 order by
RAND(maxAlumnos*0.99) limit 1;
  select cedula into  CedP from pasan1 order by
RAND(maxAlumnos*0.01) limit 1;
 select id_carrera into IdC from pasan1 order by
RAND(maxAlumnos*0.99) limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(CedA,IdC,nivel);

update materia_carrera_alumno set estado =0   where ID_ALUCAR  =
CedP;

  END WHILE;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `pasan2`(IN niv int
)
BEGIN

         DECLARE    id, CedA, CedP, IdC, Ced, Carr, idA, nivel,
maxAlumnos int;
```

```
 CREATE TEMPORARY TABLE IF NOT EXISTS pasan1
(
id int(11),
cedula int(11),
id_carrera      int(11)
);

  select count(ID) into Ced from alumno_carrera where NIVEL=niv;


 insert into pasan (id, cedula, id_carrera) Select ID, CEDULA,
ID_CARRERA from alumno_carrera where NIVEL=niv;

 select count(*) into maxAlumnos from pasan;

 set nivel=niv+1;
 WHILE maxAlumnos > 1 DO

  select count(*) into maxAlumnos from pasan1;

  select id into idA from pasan1 order by RAND(maxAlumnos*0.99)
limit 1;
  select cedula into  CedA from pasan1 order by
RAND(maxAlumnos*0.99) limit 1;
  select cedula into  CedP from pasan1 order by
RAND(maxAlumnos*0.01) limit 1;
 select id_carrera into IdC from pasan1 order by
RAND(maxAlumnos*0.99) limit 1;

  insert into alumno_carrera (CEDULA,ID_CARRERA,NIVEL) values
(CedA,IdC,nivel);

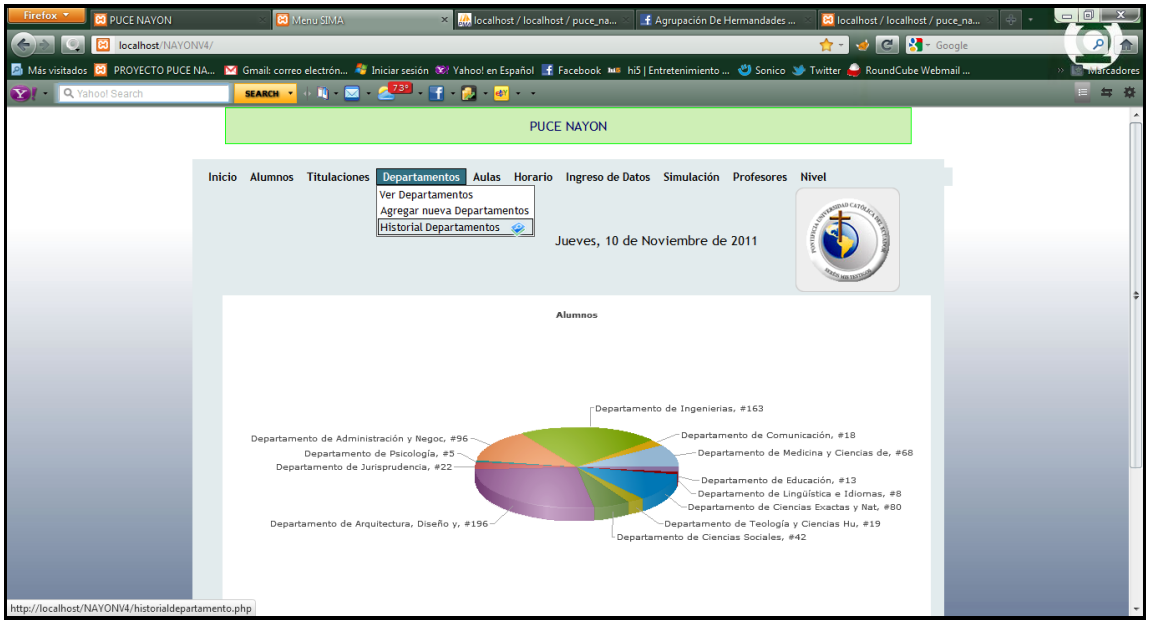
update materia_carrera_alumno set estado =0   where ID_ALUCAR  =
CedP;

  END WHILE;
END$$
```

4.2.12. HISTORIAL DEPARTAMENTO

Al dirigirse hacia el menú departamentos, se selecciona historial departamento como se muestra a continuación:

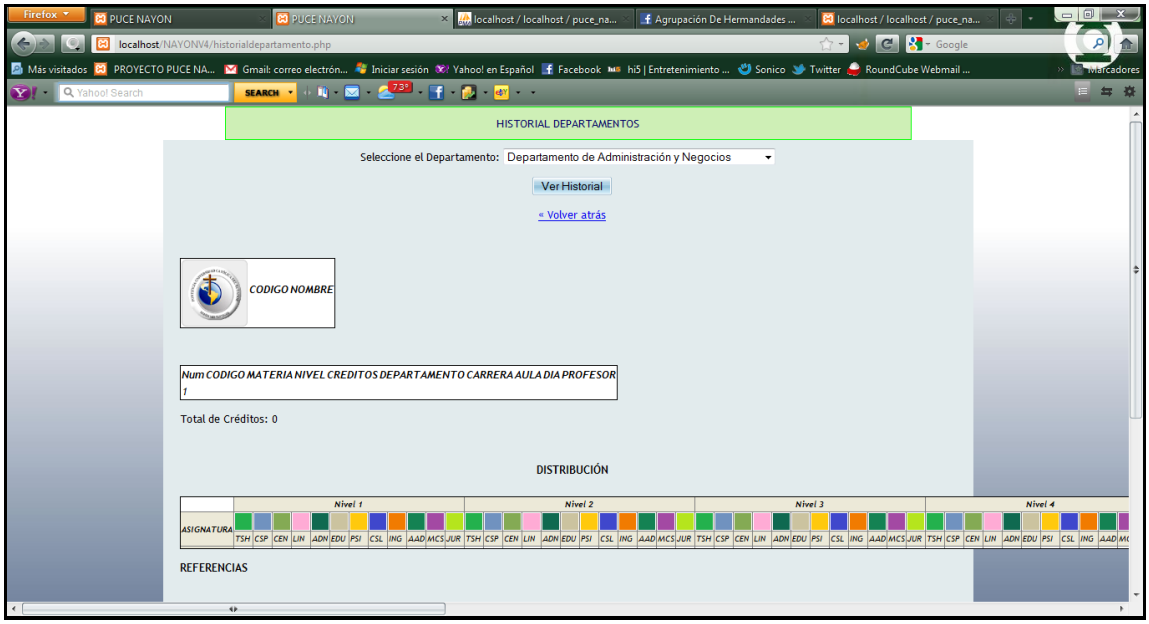
Figura V-44: Submenú Historial Departamento.



Fuente: http://localhost/NAYONV4/

Aparecerá la siguiente ventana:

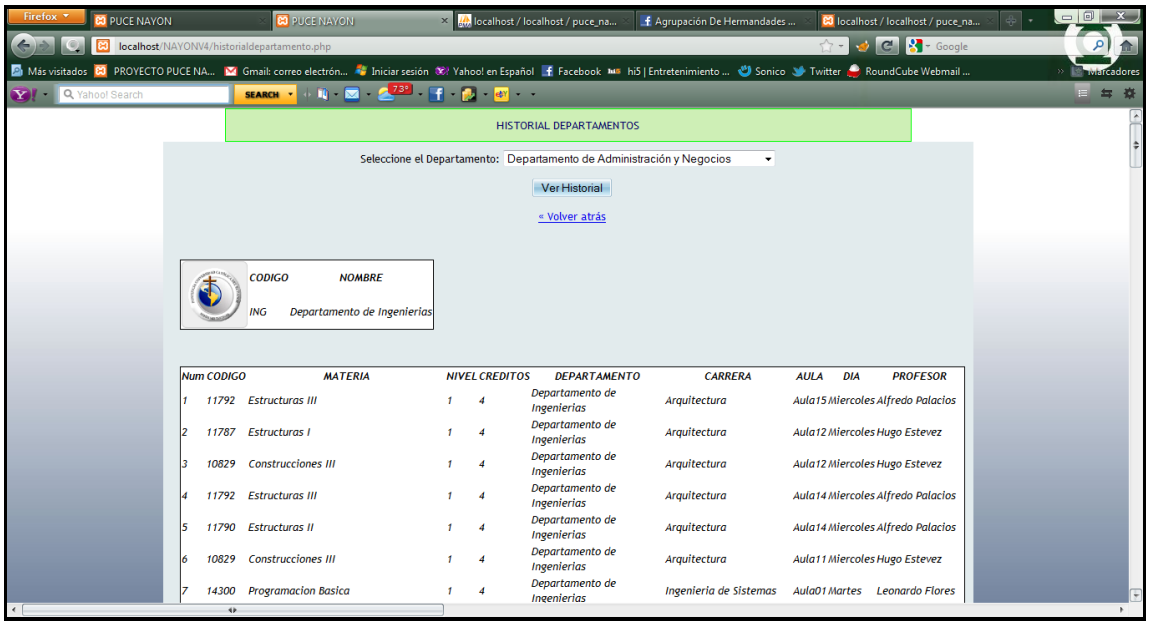
Figura V-45: Pantalla Historial Departamentos.



Fuente: http://localhost/NAYONV4/

Al seleccionar el departamento se visualiza que asignaturas por nivel están siguiendo tomadas actualmente por los estudiantes:

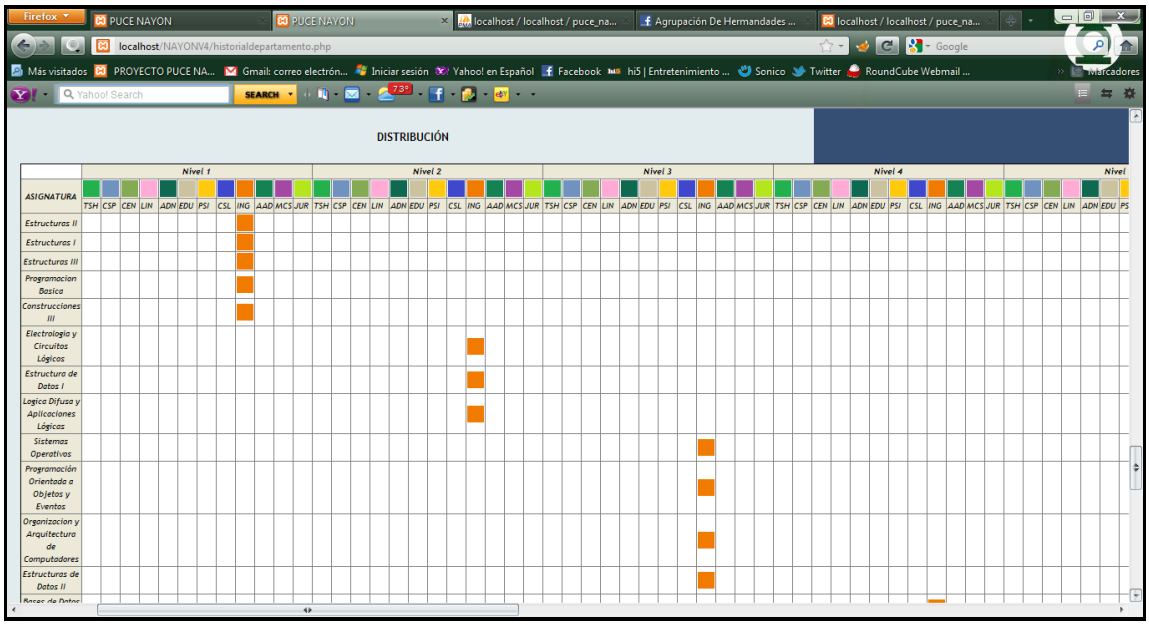
Figura V-46: Pantalla Asignaturas del Departamento seleccionado (Ingeniería como ejemplo).



Fuente: http://localhost/NAYONV4/

Se visualizan las asignaturas tomadas actualmente por los estudiantes. Adicionalmente es posible obtener de manera gráfica dichas materias con sus respectivos departamentos a los cuales pertenecen:

Figura V-47: Distribución Gráfica de Asignaturas del Departamento seleccionado (Ingeniería como ejemplo).

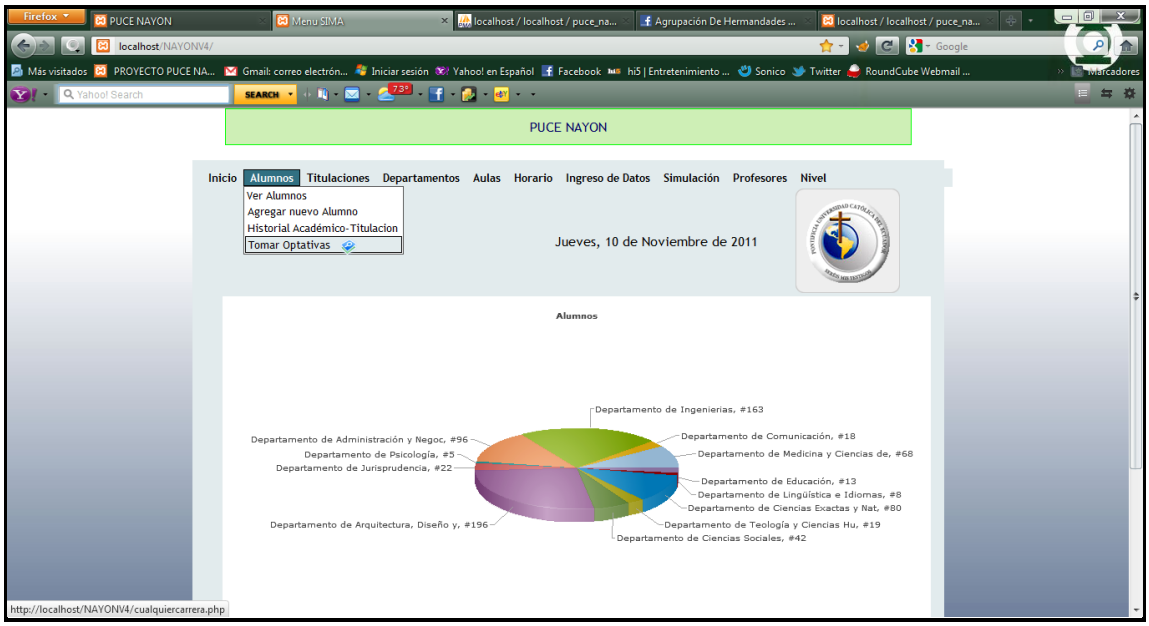


Fuente: http://localhost/NAYONV4/

5.2.1.3. ADMINISTRACIÓN DE MATERIAS OPTATIVAS

En el menú alumnos se puede seleccionar el submenú tomar optativas:

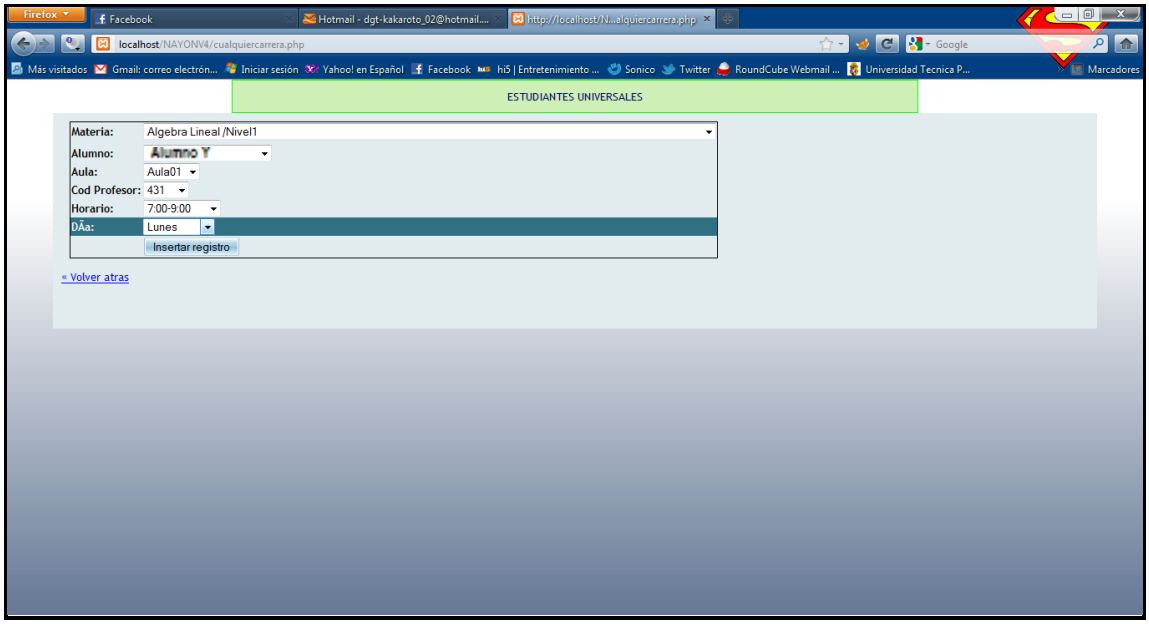
Figura V-48: Submenú Tomar Optativas.



Fuente: http://localhost/NAYONV4/

Este parte es útil para que el alumno escoja las asignaturas optativas que desee:

Figura V-49: Pantalla Tomar Optativas.



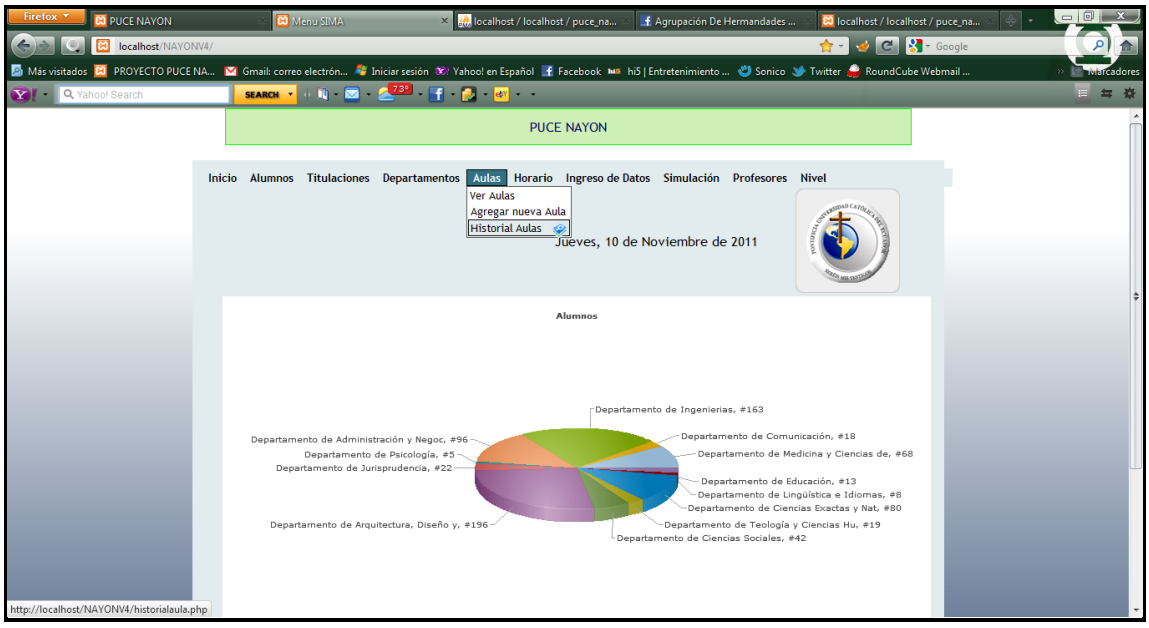
Fuente: http://localhost/NAYONV4/

Al dar clic en el link volver atrás se consigue regresar al menú principal.

5.2.1.4. HISTORIAL AULA

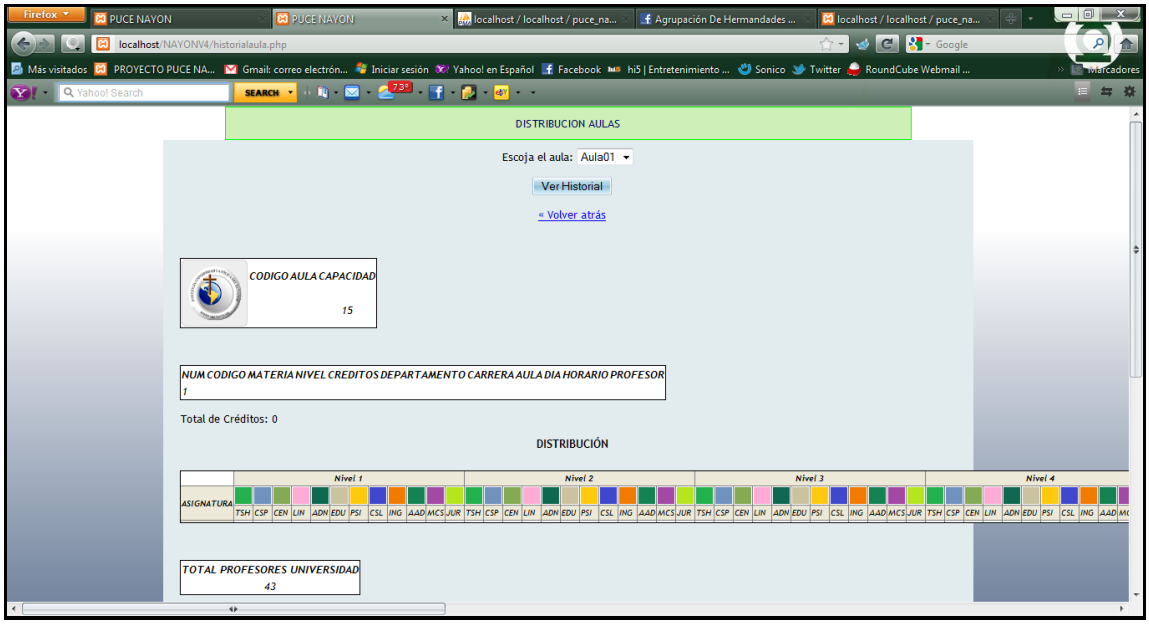
En el menú aulas se selecciona el submenú historial aula como se muestra a continuación:

Figura V-50: Submenú Historial Aulas.



Fuente: http://localhost/NAYONV4/

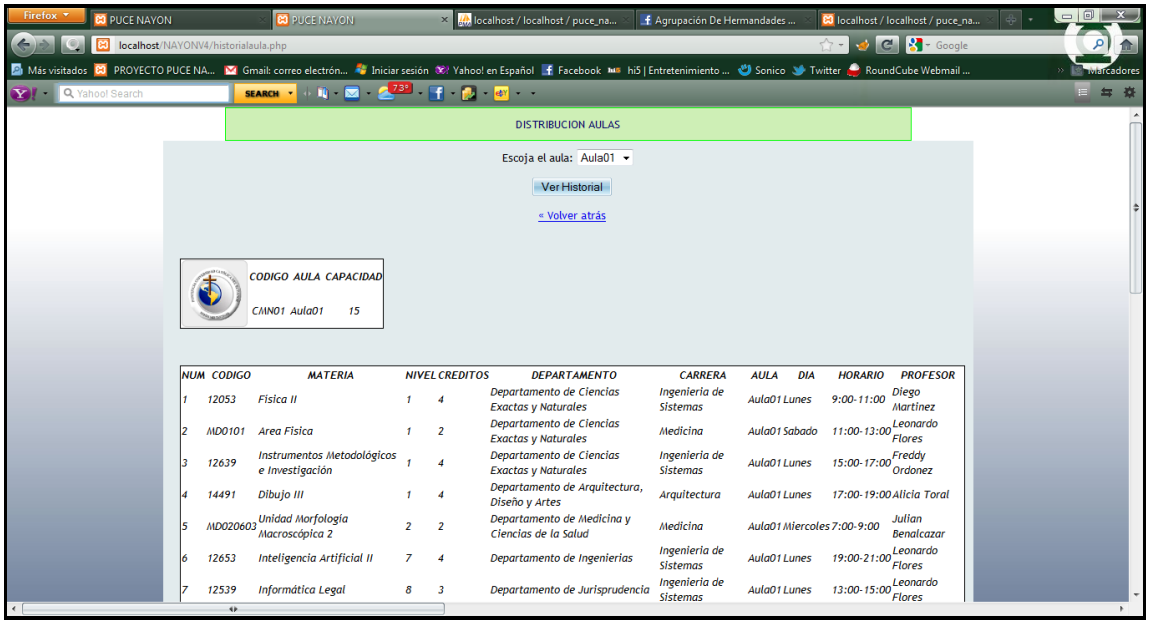
Figura V-51: Pantalla Historial Aulas.



Fuente: http://localhost/NAYONV4/

Se debe seleccionar un aula para ver su distribución:

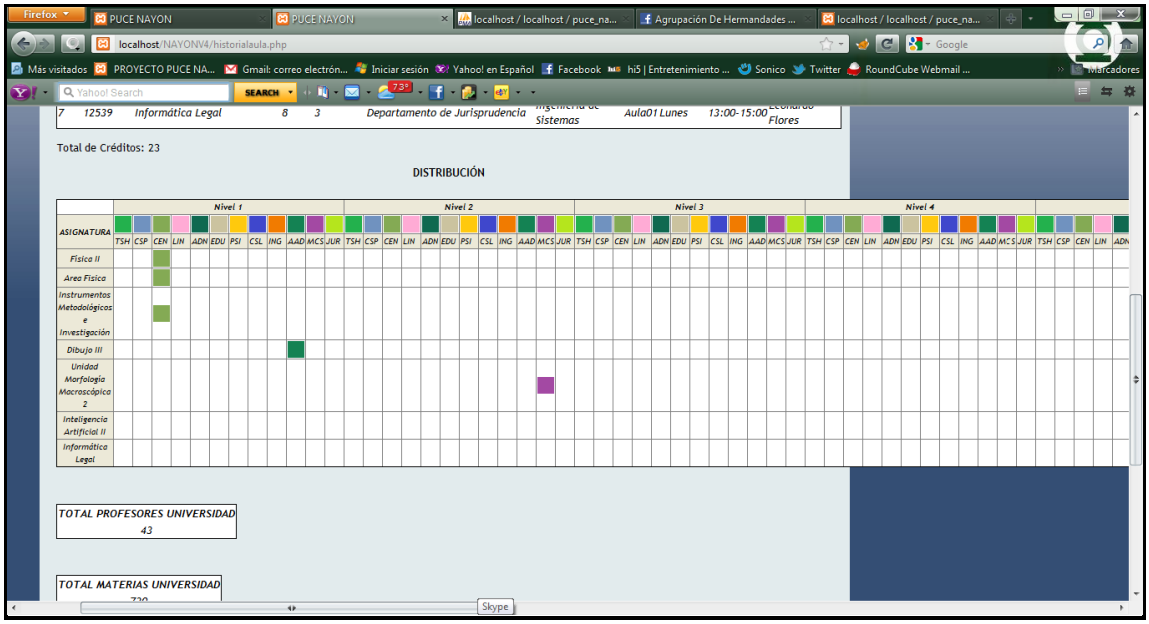
Figura V-52: Pantalla Historial del Aula seleccionada.



Fuente: http://localhost/NAYONV4/

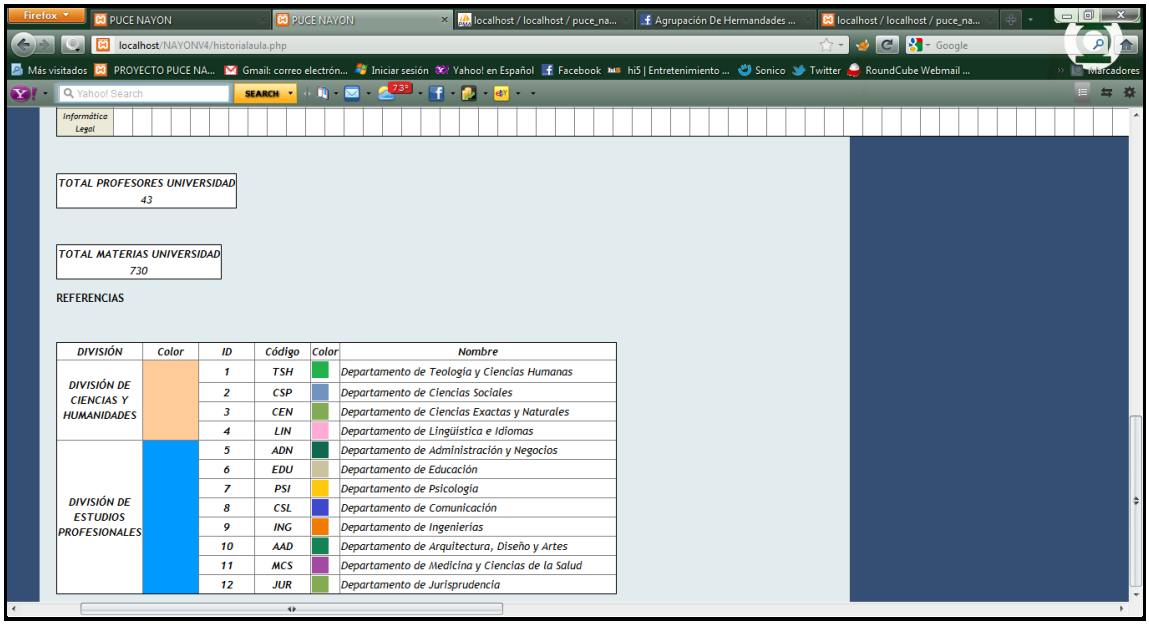
Se logra visualizar las asignaturas en las que se usa el aula, además de una distribución gráfica, estadísticas de profesores y asignaturas:

Figura V-53: Distribución Gráfica del Aula seleccionada.



Fuente: http://localhost/NAYONV4/

Figura V-54: Estadísticas de Profesores y Asignaturas.



DIVISIÓN	Color	ID	Código	Color	Nombre
DIVISIÓN DE CIENCIAS Y HUMANIDADES		1	TSH		Departamento de Teología y Ciencias Humanas
		2	CSP		Departamento de Ciencias Sociales
		3	CEN		Departamento de Ciencias Exactas y Naturales
		4	LIN		Departamento de Lingüística e Idiomas
DIVISIÓN DE ESTUDIOS PROFESIONALES		5	ADN		Departamento de Administración y Negocios
		6	EDU		Departamento de Educación
		7	PSI		Departamento de Psicología
		8	CSL		Departamento de Comunicación
		9	ING		Departamento de Ingenierías
		10	AAD		Departamento de Arquitectura, Diseño y Artes
		11	MCS		Departamento de Medicina y Ciencias de la Salud
		12	JUR		Departamento de Jurisprudencia

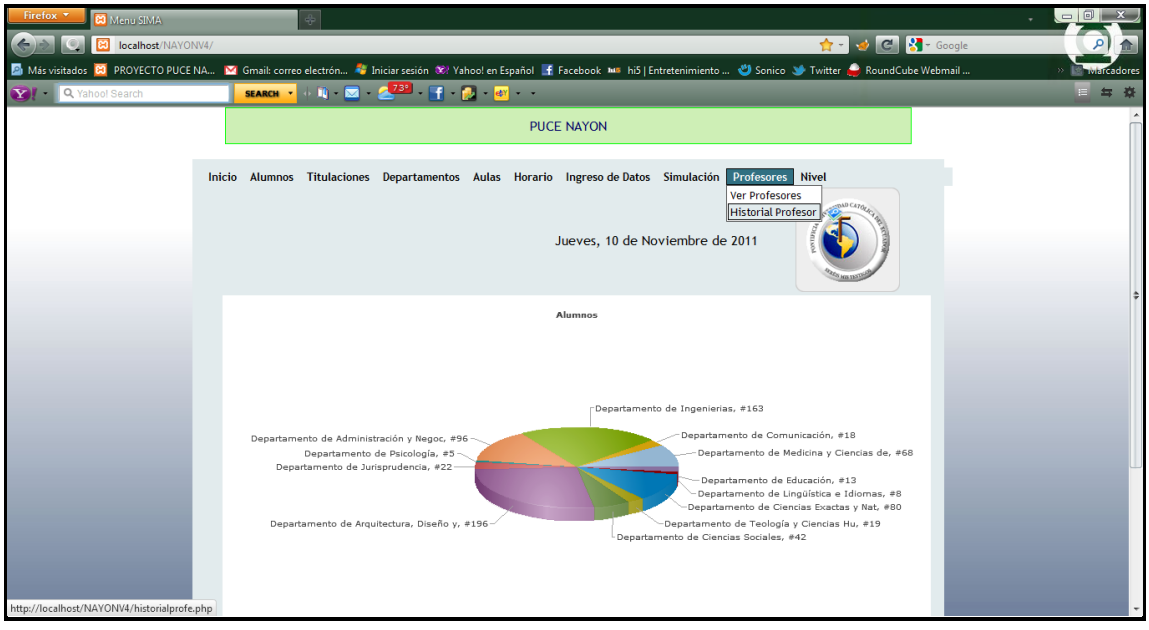
Fuente: http://localhost/NAYONV4/

Al dar clic en el link volver atrás se consigue regresar al menú principal.

5.2.15. HISTORIAL PROFESORES

Al ir hacia el menú profesores se selecciona el submenú historial profesor como se muestra a continuación:

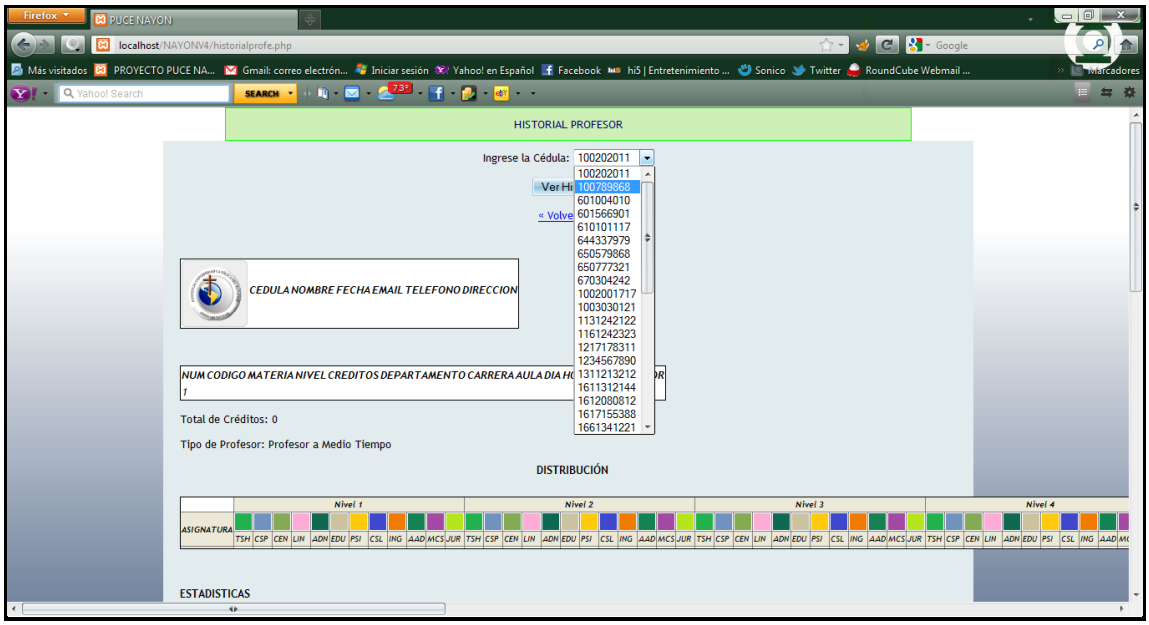
Figura V-55: Submenú Historial Profesor.



Fuente: http://localhost/NAYONV4/

Aparecerá una ventana con el historial del profesor donde será posible seleccionar la cedula del profesor:

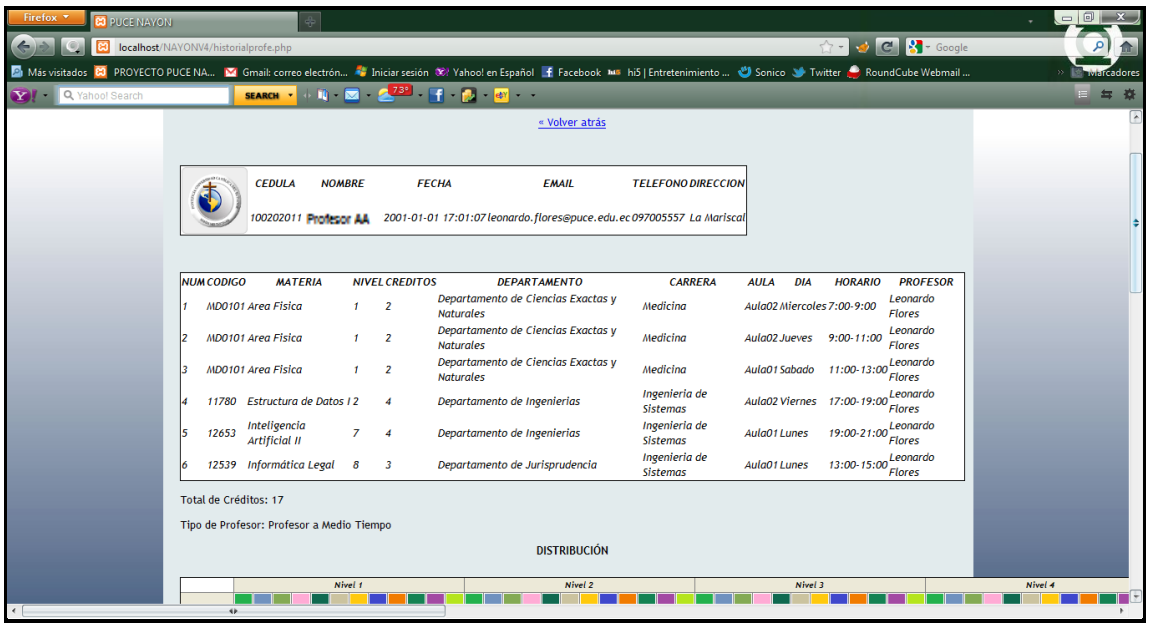
Figura V-56: Pantalla Historial Profesor.



Fuente: http://localhost/NAYONV4/

Una vez elegido un determinado profesor, se mostrará que tipo de profesor es, ya sea medio tiempo o tiempo completo, además de sus asignaturas asignadas:

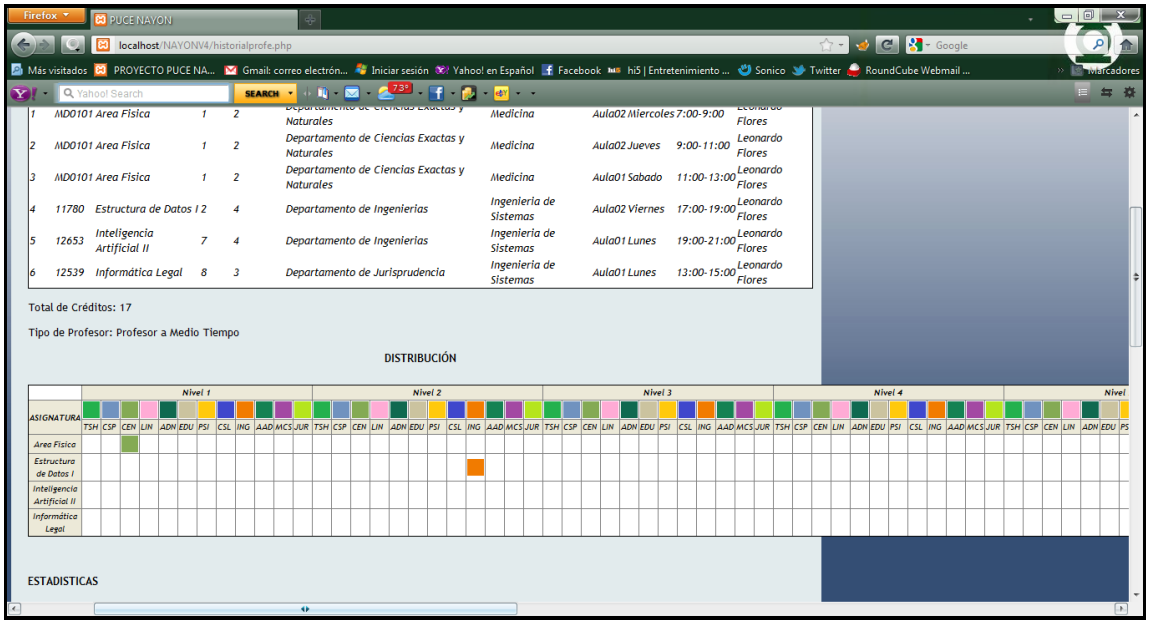
Figura V-57: Pantalla de Historial del Profesor seleccionado.



Fuente: http://localhost/NAYONV4/

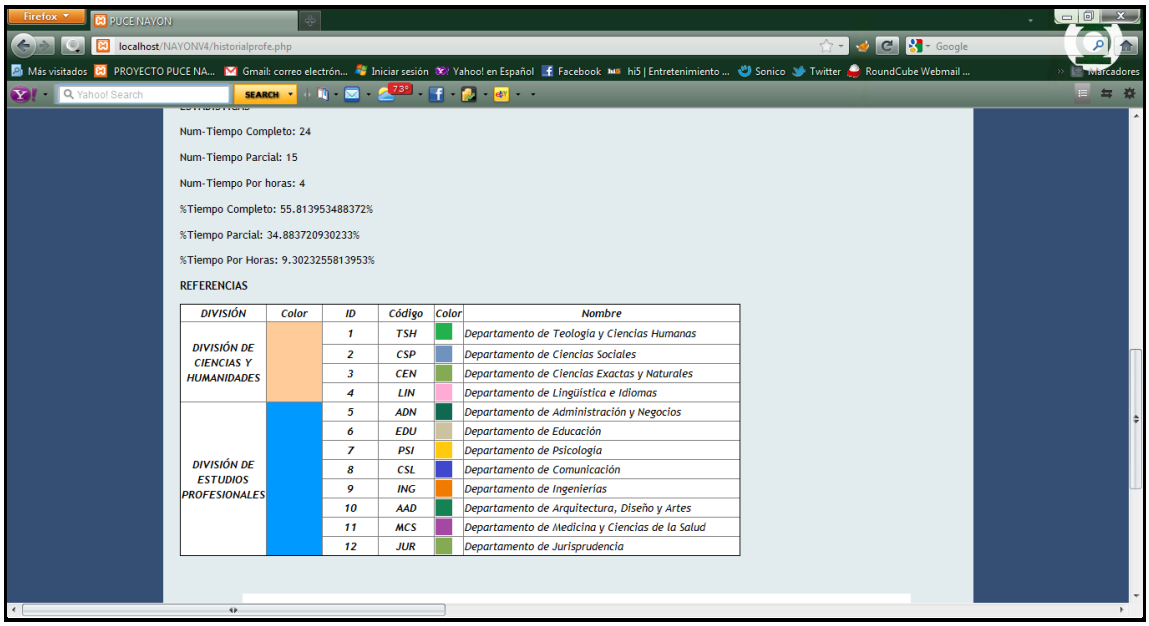
Adicionalmente de la distribución gráfica y las estadísticas generales:

Figura V-58: Distribución gráfica del Profesor seleccionado.



Fuente: http://localhost/NAYONV4/

Figura V-59: Estadísticas del Profesor seleccionado.



Fuente: http://localhost/NAYONV4/

4.3. RESULTADOS ESTADÍSTICOS

ESTADÍSTICAS DE UN ESTUDIANTE QUE TERMINA LA CARRERA DE SISTEMAS

ASIGNATURAS COMUNES	PROFESIONALIZACIÓN	OPTATIVAS	TOTAL
20	40	5	65

ESTADÍSTICAS DE UN ESTUDIANTE QUE TERMINA LA CARRERA DE ADMINISTRACIÓN DE EMPRESAS

ASIGNATURAS COMUNES	PROFESIONALIZACIÓN	OPTATIVAS	TOTAL
16	40	5	61

4.3.1. VARIABLES INDEPENDIENTES

Es cada propiedad o característica de la simulación realizada que tiene la capacidad para influir o afectar a otras variables.

4.3.1.1. DIMENSIÓN UTILIZACIÓN

La aplicación web presenta una interfaz rápida y amigable lo que permite al usuario un fácil manejo para la administración de información de alumnos, profesores, aulas, horarios, carreras académicas, niveles de estudio y departamentos estudiantiles, así como también la visualización de datos sobre la actividad y los antecedentes de los elementos que intervienen durante el tiempo requerido para la obtención de un título académico universitario.

4.3.1.2. DIMENSIÓN ESCALABILIDAD

La aplicación web tiene la capacidad de seguir creciendo, puesto que se desarrolló de una forma modular en la que se pueden agregar nuevas funciones y opciones según futuros requerimientos.

4.3.2. VARIABLES DEPENDIENTES

Es el objeto de estudio, sobre la cual se centra la simulación realizada en general.

4.3.2.1. MEDICIÓN PREVIA ANTES DE LA EJECUCIÓN DE LA APLICACIÓN WEB

Con el fin de evaluar las variables del presente proyecto, se tomó como fuente indicadores y tendencias actuales de número de estudiantes matriculados por carrera, número de estudiantes por aula, porcentaje de aprobación, porcentaje de pérdidas, porcentaje de abandono, número de aulas, entre otros, información que se generó en el primer y segundo semestre del periodo lectivo 2011 y fue suministrada por el Sub decanato de la Facultad.

5. CONCLUSIONES

1. El modelo académico "Nayón" expone como idea principal renovar y mejorar el modelo educativo bajo el cual actualmente está trabajando la PUCE, basado en el concepto de departamentalización el cual sugiere mayores ventajas frente a una organización universitaria regida por escuelas o facultades.
2. La organización universitaria departamental es un concepto el cual se refiere a estructurar una unidad académica en la que profesores especializados eduquen en un campo o disciplina determinado, con lo cual se apunta a evitar agrupar ramas las cuales no tienen asociación impidiendo así la multiplicidad, perfil bajo el cual actualmente están constituidas las diferentes facultades o escuelas.
3. Analizando todos los factores y propuestas del modelo académico "Nayón", como son la departamentalización, la asignación y distribución correcta de recursos humanos e infraestructura, entre otros, resulta notorio el beneficio que el estudiante obtiene debido a que todas estas ideas se centran en conseguir el crecimiento y desarrollo del alumno a través de la optimización de su plan de estudio.
4. Ante la necesidad de llegar a conocer comportamientos y tendencias que acogerían distintos elementos frente a una determinada situación, es importante tomar en cuenta un proceso de simulación, debido a que es un método destinado a conseguir este tipo de requerimientos.

5. Realizar una simulación a través del desarrollo de un sistema informático que muestre datos artificiales permitió a la universidad ganar experiencia en procesos de adaptación de nuevos modelos educativos, sin incurrir en los riesgos, tiempo y costos que conlleva su práctica usando recursos humanos y de infraestructura reales.
6. Logrando una correcta implementación de la metodología RAD se obtuvo la definición, diseño y modelado del sistema. Las iteraciones que incluye esta metodología junto al flujo de trabajo planteado por la misma permitieron el desarrollo de un proyecto que diera con la solución a los problemas de la universidad.
7. Los diagramas UML, como herramienta conjunta de desarrollo, hicieron posible la realización de las diferentes fases establecidas por la metodología RAD.
8. El análisis del sistema actual a través de los diagramas de casos de uso y entidad relación, facilitó la determinación de los requerimientos ya que a través de ellos se realizó la identificación, definición y recolección de los verdaderos requisitos necesarios para el desarrollo de la aplicación, que se centran en el usuario y sus necesidades.
9. Se logró establecer de forma inmediata una arquitectura del sistema sólida y lista para ser puesta en práctica, debido a que los diagramas utilizados permitieron concretar los flujos y vías con las cuales se trabajaron los datos para generar una información veraz al usuario. Por lo anteriormente descrito la fase de elaboración fue llevada a cabo de manera adecuada.
10. El adecuado diseño del modelo de la base de datos resultó fundamental en el desarrollo del software, ya que el mismo sirvió de base para erigir la arquitectura del sistema.
11. La correcta codificación del sitio web dependió en gran parte de un correcto diseño de su arquitectura.
12. Las pruebas de integración permitieron un análisis más detallado a las funcionalidades del sistema y su correcto funcionamiento.
13. La información de origen que fue ingresada al sistema para realizar los cálculos de la simulación se basa en parámetros e indicadores definidos por los historiales académicos que hoy en día maneja la universidad, por lo cual una vez generados dichos cálculos no resulta compleja la comparación que permita tomar decisiones entre el modelo educativo actual de la PUCE y el modelo académico "Nayón".

14. El comportamiento estudiantil simulado mediando el sistema, deduce la manera con la cual el tiempo designado a transcurrir durante la obtención de un título académico, es menor en los mejores casos, debido a la posibilidad de tomar la mayor parte de asignaturas requeridas en distintos departamentos a la vez.
15. Una de las características principales que conlleva la distribución universitaria mediante departamentos es la evasión de multiplicidad innecesaria de cursos de la misma materia, lo cual implica también la disminución de profesores que bajo el modelo educativo actual de la PUCE se encuentran enseñando en dichos cursos con multiplicidad.
16. La asignación y distribución optimizada de recursos humanos e infraestructura, deduce aparentemente una minoración de cantidad de aulas, con lo cual no necesariamente se abarcaría al número de estudiantes matriculados en la universidad, por lo que es necesario ampliar las opciones de horarios o designar un número mayor de materias por profesor.

6. RECOMENDACIONES

1. Realizar la debida promoción del sistema, para la participación total de los directivos y obtener de esta forma toma de decisiones basadas en una data siempre actualizada.
2. Realizar el mantenimiento preventivo de las bases de datos utilizadas con el fin de mantener información apoyada por indicadores actuales de la universidad.
3. Realizar pruebas con los usuarios finales del sistema que permitan determinar si la totalidad de sus requerimientos fueron alcanzados, en caso contrario estudiar la culminación de los requerimientos que carece el sistema.
4. Se recomienda la actualización constante del sistema de manera que al cubrir nuevos requerimientos, pueda llegar a ser un sitio web en Internet que remplace al software que actualmente lleva a cabo el proceso de matrícula y gestión de datos estudiantiles.
5. Es recomendable promover la investigación y la práctica de simuladores en todos los temas de conocimiento asociados con el desarrollo de software ya que la simulación es una herramienta educacional para enseñar a los operadores y supervisores como el sistema funcionará.

7. REFERENCIAS

7.1. BIBLIOGRAFÍA:

[A] CLARK, Burton, "Las universidades modernas: espacio de investigación y docencia", 1997.

[B] COMISIÓN ASESORA DEL RECTOR, PUCE, *"Modelo Educativo para una nueva universidad. Proyecto Nayón"*, Marzo 2011.

[C] DATE, C, *"Introducción a los sistemas de bases de datos"*, 2001.

[D] SANTOS, Pedro, *"PROLETOOL: Una herramienta para el apoyo en el estudio y la enseñanza de la asignatura Procesadores de Lenguajes*", Septiembre 2005.

[E] MATEU, Carles, *"Desarrollo de Aplicaciones Web"*, 2004.

[F] LOUDEN, Kenneth, *"Lenguajes de Programación: Principios y Prácticas"*,2002.

[G] Converse, T y Park, J, *"PHP5 and MySQL Bible"*, 2004.

[H] HEILMANN, Christian, *"Beginning JavaScript with DOM Scripting and Ajax from Novice to Professional"*, 2006.

[I] SCHMULLER, Joseph, *"Aprendiendo UML en 24 horas", (s/a).*

[J] MONTILVA, Jonás y BESEMBEL, Isabel, *"Modelado de Sistemas usando UML 2.0"*, 2007.

[K] PRESSMAN, Roger, *"Ingeniería del Software, Un Enfoque Práctico"*. 2002.

[L] CAMPOVERDE ÁVILA, Juan Manuel; OVIEDO MONTENEGRO, Franklin Sebastián, "*Diseño y Construcción de un Simulador para el Modelo Académico "Nayón" de la Pontificia Universidad Católica del Ecuador*", Diciembre 2011

7.2. INTERNET:

[1] http://cimm.ucr.ac.cr/aruiz/libros/Universidad%20y%20Sociedad/Capitulos/modelo.html

[2] http://www.puce.edu.ec/portal/content/Universidad/104?link=oln30.redirect

[3] http://webdelprofesor.ula.ve/ingenieria/hhoeger/simulacion/PARTE1.pdf

[4] http://es.wikipedia.org/wiki/P%C3%A1gina_web

[5] http://www.informaticamilenium.com.mx/Páginas/espanol/sitioweb.htm

[6] http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=500

[7] http://www.um.es/docencia/barzana/IAGP/Iagp06.pdf

[8] http://office.microsoft.com/es-es/access-help/conceptos-basicos-del-diseno-de-una-base-de-datos-HA001224247.aspx

[9] http://www.definicionabc.com/tecnologia/mysql.php

[10] http://members.fortunecity.com/miadinellie/TrabajoFinal.htm

[11] http://www.tecn.upf.es/~ocelma/cpom/per_imprimir/tema2-lenguaje_php.html

[12] http://es.wikipedia.org/wiki/Framework

[13] http://localhost/NAYONV4/ (Servidor en dónde se instalo la aplicación)

8. ANEXOS

A. MANUAL DE USUARIO

B. MANUAL TÉCNICO

ANEXO A (MANUAL DE USUARIO)

DATOS GENERALES DEL DOCUMENTO			
CÓDIGO	VERSIÓN	NOMBRE	AUTOR
MU-0001	1.0	Manual de Usuario	Juan Campoverde-Sebastián Oviedo-Luis Fernando Aguas

INTRODUCCIÓN

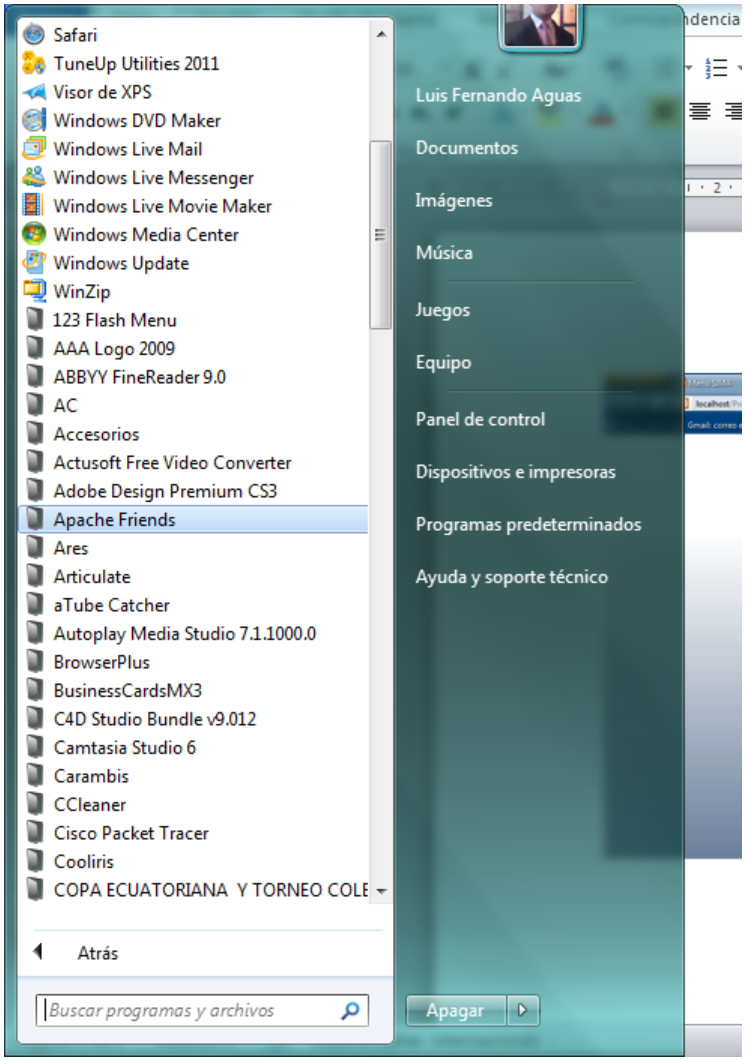
Esta herramienta realiza una simulación para el proyecto de Nayón.

INICIACIÓN DE SERVICIOS

A continuación se tiene una descripción paso a paso de las funcionalidades:

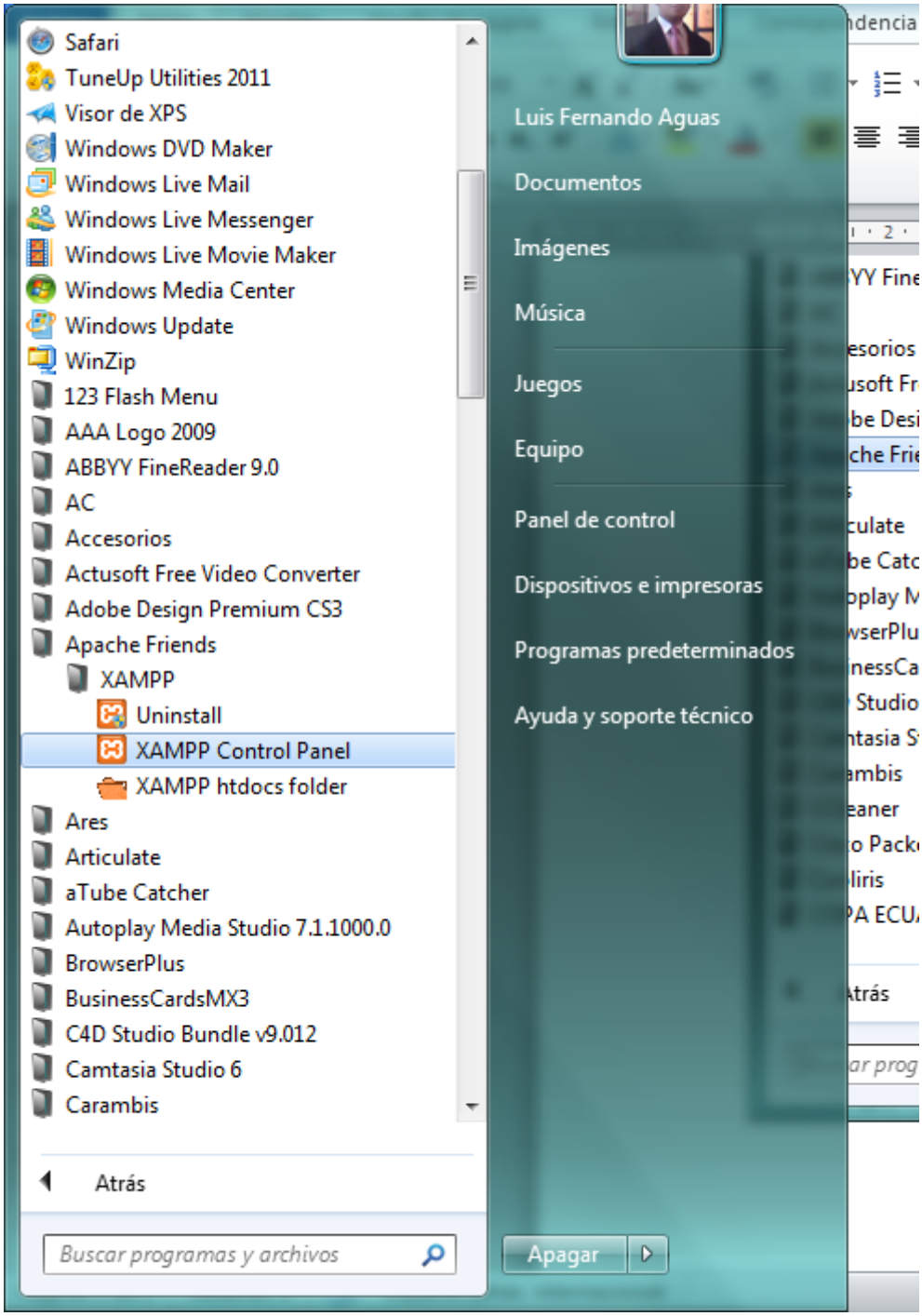
Dar clic en Menú Inicio

Ir a todos los programas



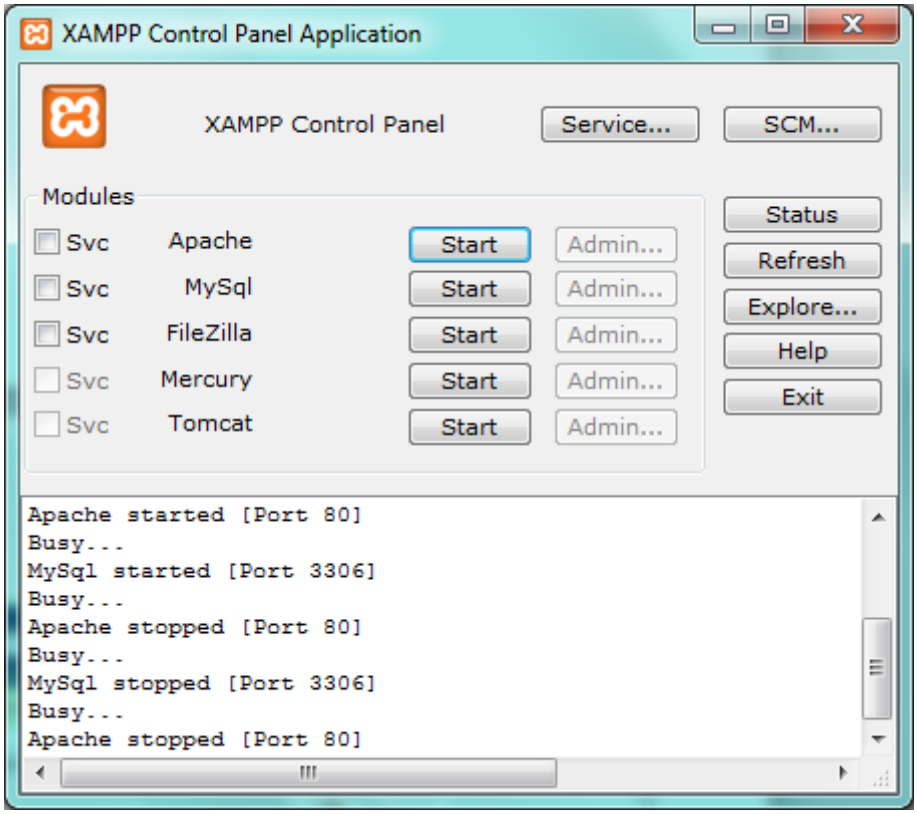
Seleccionar Apache Friends

Seleccionar XAMPP Control Panel

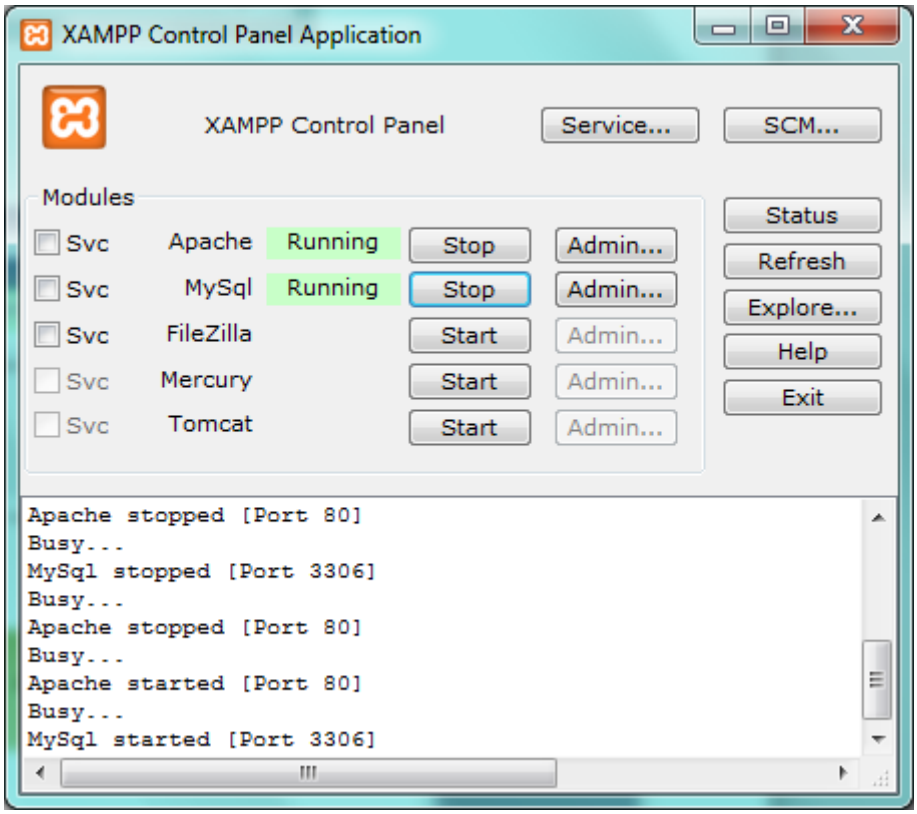


Iniciar XAMMP

Al dar clic en los botones de start tanto en Apache y MySQL



Teniendo lo siguiente:



EJECUCIÓN DEL PROGRAMA

Al abrir el navegador se debe escribir: http://localhost/NAYONV4/ con lo cual es posible visualizar el menú principal del simulador.

ANEXO B (MANUAL TÉCNICO)

DATOS GENERALES DEL DOCUMENTO			
CÓDIGO	VERSIÓN	NOMBRE	AUTOR
MT-0001	1.0	Manual Técnico	Juan Campoverde-Sebastián Oviedo-Luis Fernando Aguas

INTRODUCCION

Esta herramienta realiza una simulación para el proyecto de Nayón.

REQUERIMIENTOS DE HARDWARE

Computador Pentium IV con un mínimo de 1GB de RAM y 40 GB de disco duro.

REQUERIMIENTOS DE SOFTWARE

Navegador web (Mozilla, Internet Explorer o Equivalente)

PROCEDIMIENTO DE INSTALACIÓN

1.- Descargar el paquete Xampp desde la web de Apache Friends:

Se debe dirigir a la sección descargas par que se muestre versiones actuales y antiguas del producto, elegiremos la más reciente de XAMPP Linux, que es un paquete de más o menos 51.mb hasta el momento.

2.- Abrir una consola y como administrador descomprimimos y desempaquetar el archivo con el comando:

tar xvfz xampp-linux-1.5.3a tar.gz -C /opt

Donde 1.5.3 es la versión actual

3.- Eso es todo ahora Xampp ya está instalado en tu pc en el directorio /opt/lampp

4.- Para arrancar Xampp simplemente se teclea este comando en la consola /opt/lampp/lampp start con esto se inicia los servicios de Mysql, Apache, PHP

5.- Para probar Xampp se debe abrir el navegador y escribir: http://localhost
Aparecerá la pantalla de bienvenida de Xampp con algunos links para visualizar el estado de los servicios e instalar o desinstalar aplicaciones que vienen con el paquete.

6.- El paquete se instala por defecto sin configuraciones de seguridad, lo cual podría ser fatal para un ambiente de producción puesto que:

-El administrador del mysql(root) no tiene password.
-El demonio del Mysql es accesible vía Internet.
-ProFtpd(el servicio para subir archivos al servidor) usar el password lampp para el usuario "nobody".
-PhpMyAdmin es accesible vía Internet.

-MySQL Y Apache funcionan con el mismo usuario (nobody)

Para agregar la seguridad simplemente e escribe en la consola:
/opt/lampp/lampp security

Con este comando el sistema de seguridad solicitará los password de los servicios.

7.- Si las aplicaciones de PHP, funcionan con la versión PHP4 o PHP5, no hay ningún problema puesto que cada uno elige con que versión trabajar, con este comando se soluciona el problema: /opt/lampp/lampp php4 o /opt/lampp/lampp php5 Si no se sabe con qué versión se está trabajando, entonces se debe escribir:

/opt/lampp/lampp phpstatus

8.- Copiar la carpeta llamada seguimiento en la carpeta htdocs.

REQUERIMIENTOS DE HARDWARE (CLIENTE)

Computador Pentium II con un mínimo de 256 MB de RAM y 10 GB de disco duro.

REQUERIMIENTOS DE SOFTWARE (CLIENTE)

Navegador web (Mozilla, Internet Explorer o Equivalente)

CONFIGURACIÓN DE LA BASE DE DATOS

PhpMyAdmin, es una herramienta gráfica programada en PHP que nos permite manejar la administración de MYSQL desde nuestro navegador web.

La URL de acceso a PhpMyAdmin, a introducir en nuestro navegador, desde una instalación local y usando el paquete de Xampp (o similares), suele ser la siguiente:

http://localhost/phpmyadmin/

Crear una base de datos llamada: puce_sima_db, teniendo en cuenta las siguientes consideraciones:

- Asignarle un nombre único
- Seleccionar un "Cotejamiento de las conexiones MySQL" , apropiado, se mantiene el cotejamiento por defecto



Una vez creada la db, es conveniente crear un usuario distinto a "root" y asignarle solo privilegios específicos para la base de datos; es decir, que carezca de privilegios globales de algún tipo y solo pueda hacer "lo que quiera" dentro de esa base de datos creada anteriormente.

Para ello, desde la página de inicio de PhpMyAdmin, pulsar sobre la pestaña o enlace "Privilegios" y dar clic al enlace "Agregar un nuevo usuario", el cual conducirá a una nueva pantalla desde la que podremos configurar los privilegios de este usuario.

Lo demás, es rellenar un nombre de usuario, servidor (usualmente "localhost") y contraseña.

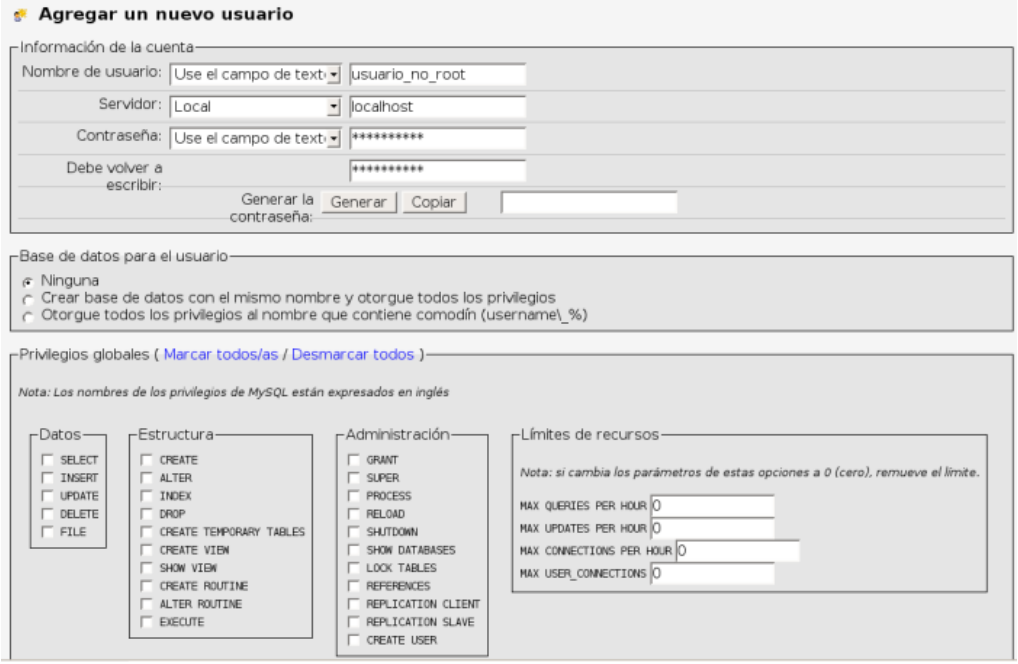
El usuario por defecto en la base es seguimiento y la contraseña es abc123@

Si se desea cambiar este nombre de usuario, se tiene que abrir el archivo dbConnect.php, encontrado en la carpeta seguimiento en la siguiente línea de código:

```
<?
//CONECION BASE DE DATOS
//mysql_connect("HOST","USUARIO","PASS");

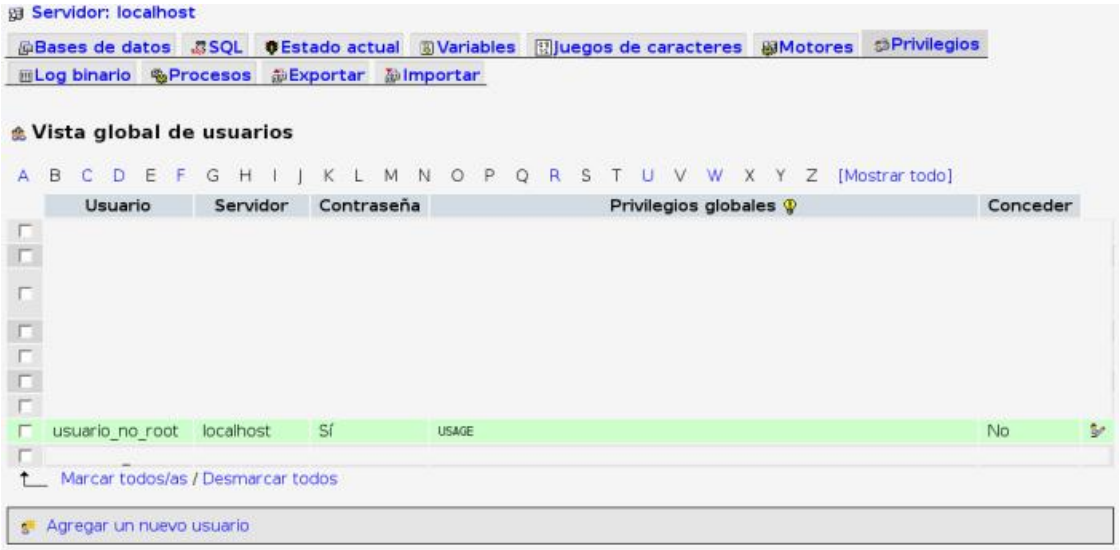
$conn = mysql_connect("localhost","puce_nayon","abc123");
mysql_select_db("puce_nayon ",$conn);

?>
```



Bien, se ha creado una db y hemos creado un usuario que, por ahora, no tiene privilegio de acceso a ninguna de las dbs. Lo que nos falta ahora por hacer, es el relacionar al usuario con las dbs que nos interese y asignarle unos privilegios específicos.

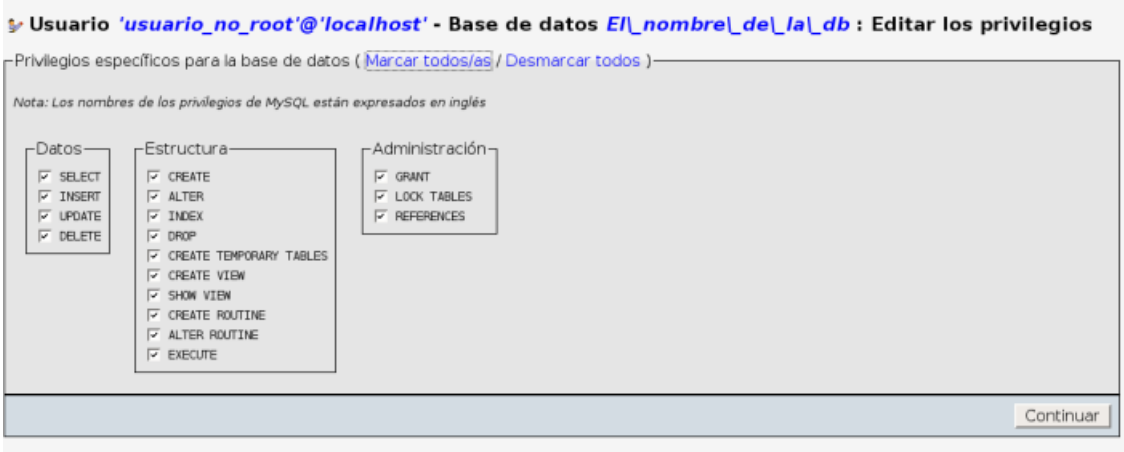
Desde la página de inicio de phpmyadmin, se puede editar los privilegios del usuario desde la pestaña o enlace "Privilegios" al pulsar sobre su "icono con un lápiz" (véase la próxima imagen)



Lo cual conducirá a otra pantalla (la de edición del usuario y sus privilegios) Ahí, de los "Privilegios globales", continuar sin seleccionar ninguno, y, en la lista de selección desplegable de título "Añadir privilegios a esta base de datos:" que se encuentra en el área de "Privilegios específicos", se tiene que seleccionar la DB que habíamos creado con anterioridad.



Ahora desde esta nueva pantalla, "seleccionar todas" las casillas de privilegios específicos y dar clic botón "Continuar"



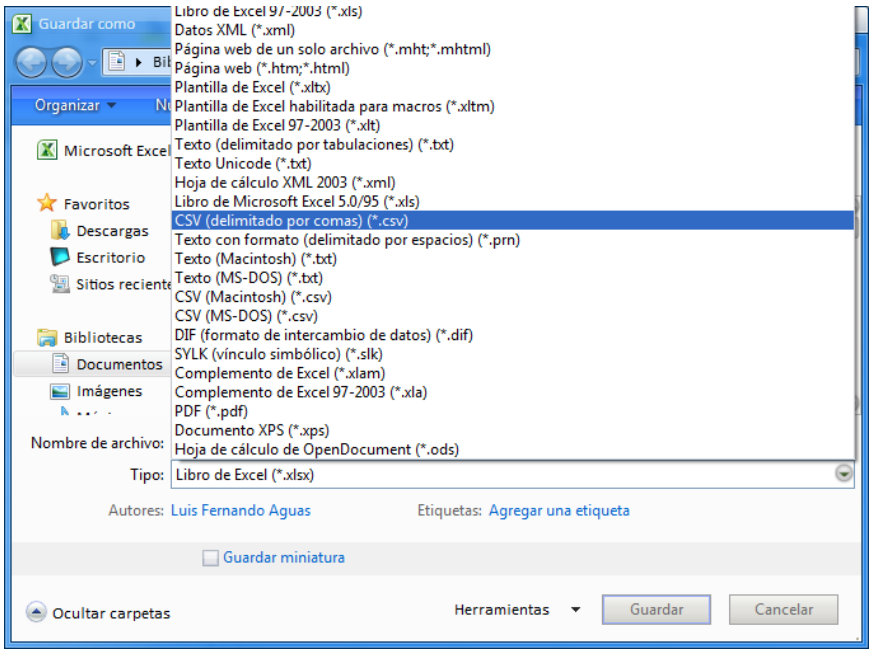
MODALIDAD DE LICENCIAMIENTO DEL SISTEMA

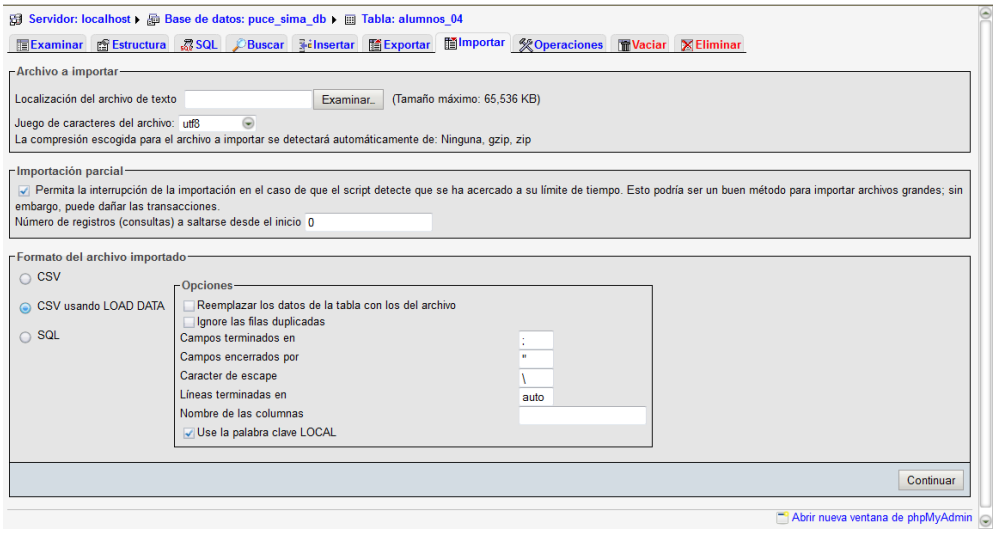
La licencia es de propiedad exclusiva de la universidad.

CANTIDAD DE LICENCIAS

La cantidad de licencias será en función de los usuarios que tenga el sistema, cabe mencionar que el sistema funcionará en la red interna de la universidad.

PROCEDIMIENTO DE RESPALDO DE LA APLICACIÓN




Servidor: localhost
Base de datos: puce_sima_db
Tabla: alumnos_04
Examinar
Estructura
SQL
Buscar
Insertar
Exportar
Importar
Operaciones
Vaciar
Eliminar
Archivo a importar
Localización del archivo de texto
Examinar..
(Tamaño máximo: 65,536 KB)
Juego de caracteres del archivo:
utf8
La compresión escogida para el archivo a importar se detectará automáticamente de: Ninguna, gzip, zip
Importación parcial
Permita la interrupción de la importación en el caso de que el script detecte que se ha acercado a su límite de tiempo. Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.
Número de registros (consultas) a saltarse desde el inicio
0
Formato del archivo importado
CSV
CSV usando LOAD DATA
SQL
Opciones
Reemplazar los datos de la tabla con los del archivo
Ignore las filas duplicadas
Campos terminados en
;
Campos encerrados por
"
Caracter de escape
\
Líneas terminadas en
auto
Nombre de las columnas
Use la palabra clave LOCAL
Continuar
Abrir nueva ventana de phpMyAdmin