



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“IMPLEMENTACIÓN DE UN SISTEMA BASADO EN MINERÍA DE
DATOS PARA LA OBTENCIÓN DE LAS PREFERENCIAS
ESTUDIANTILES DE NIVEL SUPERIOR PARA
LA PLANIFICACIÓN DE MATERIAS”

TESIS DE GRADO

Previa a la Obtención del Título de:
INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentado por:

MARLON ALFONSO ALTAMIRANO DI LUCA

LUIS ALBERTO CEVALLOS CAVERO

IVÁN ALBERTO GONZALEZ CRUZ

GUAYAQUIL – ECUADOR

2009

AGRADECIMIENTO

A Dios, motor de nuestras vidas, por todas y cada una de las bendiciones recibidas a lo largo de nuestro caminar. A nuestros padres por su amor, dedicación y sacrificio continuo en pro de nuestro bienestar y superación.

A nuestros amigos, familiares y demás seres queridos por su cariño y apoyo incondicional.

Al Ing. Fabricio Echeverría, amigo y director de nuestra tesis, por compartir sus conocimientos y hacer posible con su colaboración el desarrollo de la misma.

A nuestros apreciados profesores por colaborar en nuestro desarrollo ético y profesional.

DEDICATORIA

A Dios.

A nuestros padres.

A nuestros seres queridos.

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

MARLON ALTAMIRANO DI LUCA.

C.I. 0911393353

LUIS CEVALLOS CAVERO

C.I. 0915740138

IVÁN GONZÁLEZ CRUZ

C.I. 0920903580

TRIBUNAL DE GRADUACIÓN

MSC.ARMANDO ALTAMIRANO
PRESIDENTE DEL TRIBUNAL

ING.FABRICIO ECHEVERRÍA.
DIRECTOR DE TESIS

INC. JORGE ARAGUNDI
MIEMBRO PRINCIPAL

MBA. ANA TAPIA.
MIEMBRO PRINCIPAL

ÍNDICE GENERAL

INTRODUCCIÓN	2
CAPÍTULO 1: EL PROBLEMA	
1.1 Planteamiento del Problema	7
1.2. Delimitación del Problema	9
1.3. Evaluación	10
1.4. Interrogantes de la Investigación	11
1.5. Objetivos	13
1.5.1. Objetivo General	13
1.5. 2. Objetivos Específicos	13
1.6. Justificación e Importancia	14
CAPÍTULO 2: MARCO TEÓRICO	
2.1. Recolección de Datos	15
2.2. Construcción Teórica	19
2.3. Recursos	23
2.4. Posibles usuarios del Sistema	23
2.5. Análisis de los Datos Ingresados	24
CAPÍTULO 3: FASES DEL PROYECTO	
3.1. Ciclo virtuoso de la minería de datos	20
3.2. Fases	27
3.2.1. Fase 1: Captura de información.	28
3.2.2. Fase 2: Análisis	29
3.2.3. Fase 3: Planeación y desarrollo.	30
3.2.4. Fase 4: Demostración y uso.	32
3.3. Flujo de información en el ciclo de minería de datos.	34
CAPÍTULO 4: ANÁLISIS DEL SISTEMA	
4.1. Requerimientos funcionales	37
4.2. Requerimientos no funcionales	40
4.3. Análisis de tecnologías	41
4.4. Descripción de casos de uso y escenarios	45
4.5. Análisis de la interacción Hombre - Máquina	52

CAPITULO 5: DISEÑO Y MANEJO DEL SISTEMA

5.1. Diseño de la arquitectura del sistema	56
5.2. La correcta lógica para una correcta capa	59
5.3. Arquitectura de SADPOL.	60
5.4 DISEÑO DEL ESQUEMA ADMINISTRADOR	62
5.5 DISEÑO DEL ESQUEMA ESTUDIANTE	63
5.6. Diseño de interfaz	76
5.7. Flujo de ventanas y layouts	76
5.8. Diseño de página web	79
5.9. Diseño del plan de prueba	85

CAPITULO 6: PRUEBAS Y ANÁLISIS

6.1. Implementación	88
6.2. Resultados	90
6.3. Análisis de pruebas	91
CONCLUSIONES Y RECOMENDACIONES	92
ANEXOS	94
ANEXO 1 GUIA DE USO DEL SISTEMA	95
ANEXO 2 MANUAL DEL USUARIO 1 ADMINISTRADOR	99
ANEXO 3 MANUAL DEL USUARIO 2 ESTUDIANTE	104
BIBLIOGRAFÍA	107

ÍNDICE DE FIGURAS

Figura 1.4. Proceso de Minería de Datos.	12
Figura 2.4. Posibles usuarios del sistema.	24
Figura 3.1. Ciclo virtuoso de la minería de datos.	27
Figura 3.2. Flujo de información en el ciclo virtuoso de la minería	34
Figura 4.1. Tareas del usuario Administrador.	39
Figura 4.2. Tareas del usuario Estudiante.	40
Figura 4.3. Aplicación del lado del servidor y del cliente.	44
Figura 5.1. Arquitectura de las 3 capas.	56
Figura 5.2. Herramientas para diseño del sistema.	61
Figura 5.3. Modelo de datos del esquema administrador	62
Figura 5.4. Modelo de datos del esquema estudiante.	63
Figura 5.5. Flujo de ventanas para un usuario estudiante	77
Figura 5.6. Flujo de ventanas para un administrador del Sistema.	78
Figura 5.7. Áreas en las que se dividen las pantallas del sitio	80
Figura 5.8. Pantalla de bienvenida al usuario que ingresa al Sistema.	82
Figura 5.9. Pantalla de formulario.	83
Figura 5.10. Figura con el flujo	84
Figura A. Pantalla principal del sistema.	95
Figura A1 Acceso autorizado	97
Figura A2 mensaje de error	98
Figura A3 Nombre de usuario	98
Figura A2.1 Log de transacciones	102
Figura A2.2 Ingresar o modificar estudiantes	103
Figura A3.1. Ingreso o selección de un registro para modificar sus datos.	105
Figura A3.2. Modificación y Eliminación	106

ÍNDICE DE TABLAS

Tabla 4.3.a.	Lenguajes de programación.	42
Tabla 4.3.b.	Características del webserver APACHE.	43
Tabla 4.3.c.	Características de postgresQL.	43
Tabla 5.4.a	Tabla Administrador.administrador.	64
Tabla 5.4.b.	Tabla Administrador.transacciones.	66
Tabla 5.5.a.	Tabla Estudiante.administrador.	68
Tabla 5.5.b.	Tabla Estudiante.carrera.	68
Tabla 5.5.c.	Tabla Estudiante.especializacion.	69
Tabla 5.5.d.	Tabla Estudiante.facultad.	69
Tabla 5.5.e.	Tabla Estudiante.dias.	69
Tabla 5.5.f.	Tabla Estudiante.eleccion.	70
Tabla 5.5.g.	Tabla Estudiante_estudiante.	71
Tabla 5.5.h.	Tabla Estudiante_factor.	72
Tabla 5.5.i.	Tabla Estudiante_eleccion_new.	73
Tabla 5.5.j.	Tabla Estudiante.horario.	73
Tabla 5.5.k.	Tabla Estudiante.horas.	74
Tabla 5.5.l.	Tabla Estudiante.estudiante_materias_aprobadas.	74
Tabla 5.5.m.	Tabla Estudiante.materias.	75
Tabla 5.5.n.	Tabla Estudiante.nivel.	76

TEMA Y PROPUESTA DE TESIS:

“Implementación de un sistema basado en minería de datos para la obtención de las preferencias estudiantiles de nivel superior para la planificación de materias”

INTRODUCCIÓN

Hoy en día, al igual que en el resto de ámbitos de la sociedad, las tecnologías de la información y las comunicaciones han irrumpido en el ámbito educativo. Así, es posible capturar y recopilar de forma muy sencilla y con coste reducido todo tipo de información: datos administrativos de matriculaciones en colegios, institutos o universidades, expedientes académicos informatizados de los estudiantes y registro de actividades.

La **minería de datos educativos** (Educational Data Mining, **EDM**) es el proceso de transformar los datos en bruto recopilados por los sistemas de enseñanza en información útil que pueda utilizarse para tomar decisiones informadas. [Cecily Heiner, Ryan Baker y Kalina Yacef, Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006). Jhongli, Taiwan].

La revolución digital ha hecho posible que la información digitalizada sea fácil de capturar, procesar, almacenar, distribuir, y transmitir. Con el importante progreso en informática y en las tecnologías relacionadas

y la expansión de su uso en diferentes aspectos de la vida, se continúa recogiendo y almacenando en bases de datos gran cantidad de información.

Todas las herramientas tradicionales de minería de datos asumen que los datos que se usarán para construir los modelos contienen la información necesaria para lograr el propósito buscado: obtener suficiente conocimiento que pueda ser aplicado a la ESPOL (o problema) para obtener un beneficio (o solución), para el cliente (estudiante).

La gran cantidad de bases de datos académicas que acumulan información de estudiantes de nivel superior y de catedráticos serían de mayor utilidad, si esta minería de datos es puesta al alcance de los estudiantes con la implementación de un sistema que permita elegir según las preferencias de los usuarios sus materias. La minería de datos (DT) comprende un conjunto de técnicas, algoritmos y métodos que permiten a partir de un conjunto de información alcanzar razonamientos y proposiciones de ámbito general, partiendo de algo particular.

La minería de datos no es nueva, tiene nacimiento en los años 60' consolidándose en los 80'. Según nuestras investigaciones y sus siglas en inglés Data Mining es el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Las herramientas de Data Mining predicen futuras tendencias y comportamientos, permitiendo la toma de decisiones.

La información resultante tras trabajar con la minería de datos, es aplicable a tomar decisiones, formando parte del conjunto de tecnologías del mundo actual. Grandes empresas e instituciones públicas, privadas y de educación, ya han creado y alimentan bases de datos especialmente diseñadas para proyectos de minería de datos, en las que centralizan información potencialmente útil de todas sus áreas de trabajo, estrategias de marketing, etc.

La minería de datos por lo general contempla las siguientes fases:

1. Determinación de los objetivos. Trata de la delimitación de los objetivos que se desea alcanzar.

2. Pre-procesamiento de los datos. Se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos. Es la etapa que consume más de la mitad del tiempo del proyecto.

3. Determinación del modelo. Se comienza realizando unos análisis estadísticos de los datos, y después se lleva a cabo una visualización gráfica de los mismos para tener una primera aproximación. Según los objetivos planteados y la tarea, que debe llevarse a cabo, pueden utilizarse algoritmos desarrollados en diferentes áreas de la Inteligencia Artificial.

4. Análisis de los resultados. Verifica si los resultados obtenidos son coherentes y los coteja con los obtenidos por los análisis estadísticos y de visualización gráfica. El usuario determina si son novedosos y si le aportan un nuevo conocimiento que les permita considerar sus decisiones.

La finalidad de estas fases que son las características básicas de un sistema en minería de datos, es que el estudiante a quien denominamos el usuario, tenga acceso al nicho de la información catedrática e itineraria de la Escuela Superior Politécnica del Litoral,

para que esta sea una herramienta de la nueva tecnología que avanza a pasos agigantados, que le ayude en forma óptima a la planificación de materias al tiempo y horario a su elección con el catedrático que tuviere conveniente, creando así una planificación de selección de materias que facilite el uso de su tiempo sin afectar también la organización ya preestablecida en la plataforma del sistema, esto ayudará también como una estrategia de marketing al hacer que la institución sea la pionera en brindar soluciones estudiantiles enfocados al tema de nuestro proyecto.

CAPÍTULO 1

EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

La Escuela Superior Politécnica del Litoral ESPOL, cuenta con alrededor de 10 000 estudiantes procedentes de todas las provincias y lugares del país, con los más diversos orígenes sociales y académicos; sin que, hasta el momento, se hayan realizado estudios que evalúen la influencia de estos factores en su formación posterior. Por lo que estos factores no son tomados en cuenta a la hora de realizar el proceso de organización de actividades en la planificación de cátedras, horarios de clases, para los estudiantes presentes y futuros de la Universidad. Lo que crea debido a la diversidad de la población de usuarios estudiantiles de toda índole y clases sociales, una necesidad como es la de planificar su elección de horas y materias de estudio conforme a sus prioridades y preferencias, que no afecten su estilo de vida y que lo lleven a una formación óptima profesional. Mientras que en otros casos se puede identificar a los alumnos con mayor disponibilidad de sujetarse a planificaciones ya preestablecidas, que pudieran formar parte de proyectos o grupos de investigación, o simplemente una organización de profesores con la información conveniente para que

puedan brindarle atención diferenciada a sus estudiantes en aras de fomentar el pleno desarrollo de sus capacidades y dándole así cumplimiento al objetivo primordial de la Universidad, que es el de formar profesionales en todas las especializaciones cada vez mejor preparados.

Toda la información personal y docente de los estudiantes, desde hace mucho tiempo se encuentra digitalizada y se mantiene en históricos que no brindan mayor utilidad que la con la aplicación de esta tecnología se puede mejorar.

Es por esto que en la Universidad se hace necesario tener métodos eficientes y automáticos para explorar las grandes bases de datos, procesando de forma rápida y fiable la información para encontrar patrones de conocimiento apropiados para resolver un problema.

El objetivo fundamental de este trabajo está orientado a determinar las preferencias estudiantiles al momento de elegir las materias con el fin de que el sea un planificador de las mismas vínculo que existe entre el origen y procedencia social de los estudiantes de la ESPOL con sus

resultados académicos mediante la aplicación de técnicas de agrupación y reglas de asociación de minería de datos.

El presente trabajo plantea la aplicación de una tecnología basada en la implementación de un sistema en minería de datos, por lo que en la misma tratamos de definir qué es, a que aporta al conocimiento y que utilidad brinda al desarrollo de soluciones para los usuarios (estudiantes a nivel superior) en la selección de materias de acuerdo a sus planificaciones y acorde a sus necesidades.

1.2. DELIMITACIÓN DEL PROBLEMA

El desarrollo de nuestra tesis se encuentra delimitado en todo el campus de la Escuela Superior Politécnica del Litoral (ESPOL), en la ciudad de Guayaquil, cuyo propósito será una primordial herramienta para cada usuario de la universidad, estudiante vía on-line (Internet) y en el caso del administrador con la aplicación de la tecnología de la minería de datos en el sistema considerado para su uso en el acceso de la base de datos, para la respectiva coordinación de los horarios de clases y las materias preferidas y seleccionadas por los estudiantes.

1.3. EVALUACIÓN

Este proyecto va a ser evaluado mediante:

- Revisión continua y permanente de contenidos
- Contrastar información mediante comparación de fuentes de información.
- Aplicación de instrumentos adecuados

Factibilidad.- La investigación es factible porque ofrece una posible solución a una necesidad actual entre los estudiantes de la ESPOL, además este proyecto cuenta con el recurso humano, intelectual, tecnológico y económico necesario para su desarrollo.

Claro.- Que se esta aplicando en una Institución.

Preciso.- Por los resultados obtenidos.

Permanente.- Que se lo esta realizando desde la aprobación del tema y durante todo el proceso.

Objetivo.- Que es perceptible no se presta a subjetividades ni emociones.

Cualitativa.- Puede ser expresada en términos de calidad.

Cuantitativa.- Puede expresar en términos de cantidad.

1.4. INTERROGANTES DE LA INVESTIGACIÓN

El presente proyecto responde a las siguientes interrogantes que se dieron a través de nuestras investigaciones:

*** ¿Qué es una minería de datos?**

Consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

*** ¿Para qué sirve esta tecnología?**

Dicha tecnología en nuestro trabajo nos sirvió como medio importante para saber las preferencias estudiantiles y así poder conocer cuales son los requerimientos en cuanto al horario y sugerencia de profesor para la buena planificación.

*** ¿Dónde se aplica una minería de datos?**

Existe una lista muy extensa de donde se aplica la minería de datos, como por ejemplo; comportamiento en Internet, recursos humanos, patrones de fuga, ingeniería eléctrica, etc, siendo lo principal el análisis de datos.

*** ¿Cuáles son los beneficios de aplicar una minería de datos?**

Dentro de los beneficios al aplicar técnicas de minería de datos esta la buena planificación y buen uso de la información adquirida, siendo el único beneficiado el usuario final.

*** ¿Qué fases contempla la minería de datos?**

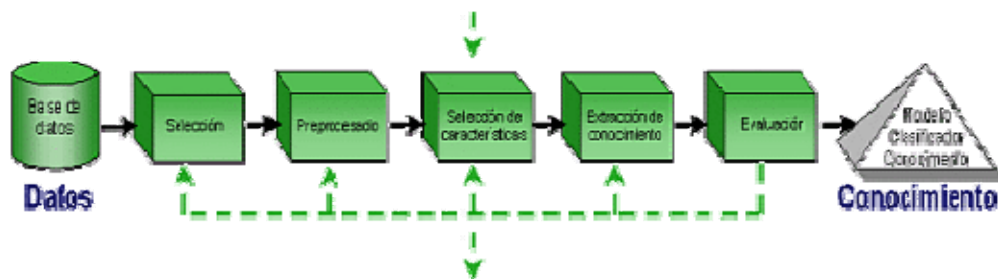


Figura 1.4. Proceso de minería de datos.

Fuente: <http://www.daedalus.es/mineria-de-datos/proceso-de-mineria-de-datos/>

Los pasos a seguir para la realización de un proyecto de minería de datos son siempre los mismos, independientemente de la técnica específica de extracción de conocimiento usada.

*** ¿Cubrirá las expectativas la aplicación de un sistema de minería de datos a todos los usuarios de la ESPOL?**

Sí, la aplicación de dicho sistema dentro de la comunidad politécnica creará una cultura de planificación ordenada teniendo como resultado las preferencias del estudiante a la hora de escuchar clases.

*** ¿Cómo sería diseñado el sistema?**

En cuanto al diseño debería tener una interacción hombre-máquina bastante simple y entendible a la vista, sin opción a confundirse ya que su buen manejo se releja en la obtención de datos reales y concretos.

1.5 OBJETIVOS

1.5.1. OBJETIVO GENERAL

- Crear un sistema que obtenga las preferencias estudiantiles para la planificación de materias, utilizando la tecnología de minería de datos, extraer información útil a partir de datos del sistema académico de la ESPOL, para este fin y dar a conocer una panorámica general de lo que es una minería de datos y las aplicaciones de las herramientas de manera óptima para la creación del sistema propuesto en nuestra tesis.

1.5.2. OBJETIVOS ESPECÍFICOS

- Utilizar los modelos discriminantes para el desarrollo de la clasificación de las preferencias.

- Definir los DATAMARTS donde se concentren los datos operativos, valores de hecho y conocimientos extraídos de los cuestionarios electrónicos.
- Implementar las interfaces web necesarias para: ingreso de datos de los estudiantes, operación de los DATAMARTS y ambiente de toma de decisiones.

1.6. JUSTIFICACIÓN E IMPORTANCIA.-

El desarrollo de nuestro proyecto aportará a la comunidad politécnica con un sistema donde el estudiante podrá dar a conocer su disponibilidad de tiempo y a su vez recomendar horarios, profesores con los cuales quisiera escuchar clases, creando una cultura de planificación en base a recolección de información.

Pensando no solo en el bienestar del usuario sino también de la institución que lo administra se podría tener una visión diferente de la demanda que se tiene ya que el buen manejo de los datos obtenidos permitirá la implementación de nuevos horarios de clases generando indirectamente el crecimiento del recurso humano con el aporte de fuentes de empleo.

CAPÍTULO 2

MARCO TEÓRICO

2.1 RECOLECCIÓN DE DATOS

Toda la información necesaria para alimentar el sistema que se propone en este proyecto se encuentra en la base de datos SISTEMA ACADÉMICO, de la ESPOL; por lo que no fue necesario integrar varios orígenes de datos para su recolección. Los atributos seleccionados para realizar el proyecto de minería se encontraban en varias vistas dentro de la base de datos.

El sistema será un anexo en el sistema Censo Académico en Línea (CENACAD) que hará referencia a un formulario personal para cada alumno que ingrese su respectivo número de matrícula, el sistema presentará el respectivo flujo del alumno señalando de forma puntual las materias que ya ha tomado y las materias que esta tomando en el actual semestre permitiendo visualizar las materias que no ha tomado aún; así disminuirémos el ingreso de elecciones erróneas por parte del alumno, dentro del formulario una vez escogido las materias se le pedirá al usuario que escoja con que profesor desearía tomar cada una

de las materias elegidas con el horario que le parezca mas conveniente.

Este módulo del sistema no pondrá restricciones en el ámbito de la asignación de profesor y el horario que decida el usuario ya que nuestro objetivo es conocer también preferencias del alumno para su respectivo análisis.

Para asociar en una sola tabla los datos personales de los estudiantes; se integra utilizando la técnica ETL (Extraer, Transformar y Cargar Extract, Transform and Load), Este proceso permite mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en un data warehouse para su respectivo análisis. [Wikipedia, Enciclopedia Libre, (es.wikipedia.org/wiki/ETL)]. Extraeremos información que se obtienen de datos históricos a partir de números de matrículas, correspondientes a estudiantes matriculados en la Universidad en el período comprendido entre los años 1998 aproximadamente al 2008.

La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

La fase de transformación aplica una serie de reglas o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. En ciertos casos pueden ser necesarias aplicar algunas de las siguientes transformaciones:

- Seleccionar sólo ciertas columnas para su carga (las columnas con valores nulos no se cargan).
- Traducir códigos (si la fuente almacena una "H" para Hombre y "M" para Mujer en el destino tiene que guardar "1" y "2" según el caso).
- Obtener listado de materias a elegir.
- Calcular totales de múltiples filas de datos (número de materias elegidas).
- Generación de campos clave en el destino.
- Dividir una columna en varias (columna "Nombre: García Miguel"; pasar a dos columnas "Nombre: Miguel" y "Apellido: García").
- La aplicación de cualquier forma, simple o compleja, de validación de datos, y la consiguiente aplicación de la acción que en cada caso se requiera:
 - Datos OK: Entregar datos a la siguiente etapa (Carga).

- Datos erróneos: Ejecutar políticas de tratamiento de excepciones (rechazar el registro completo, dar al campo erróneo un valor nulo).

La fase de carga es el momento en el cual los datos de la fase anterior (**transformación**) son cargados en el sistema. Emplearemos un **data warehouse** para mantener el historial de los registros de manera que se pueda hacer una auditoria de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

Existen dos formas básicas de desarrollar el proceso de carga:

- Acumulación simple: Consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el data warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio de la magnitud considerada.
- Rolling: Se aplica para mantener varios niveles de granularidad. Se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo en alguna o varias de las dimensiones de la magnitud almacenada (Wikipedia, Enciclopedia Libre).

Somos concientes que la tecnología actual y su creciente demanda, el desarrollo de tecnologías de minería de datos más avanzadas para interpretar la información y el conocimiento de los datos distribuidos por todo el mundo. En este siglo la demanda continuará creciendo, y la ESPOL, no podría quedarse atrás al ser una de las Universidades de mayor prestigio, debe ir a la par siempre al avance con la tecnología.

2.2. CONSTRUCCIÓN TEÓRICA

Se denomina Minería de Datos al conjunto de técnicas y herramientas aplicadas al proceso no trivial de extraer y presentar conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a partir de grandes conjuntos de datos, con objeto de predecir de forma automatizada tendencias y comportamientos; y describir de forma automatizada modelos previamente desconocidos [Piatetski-Shapiro et al., 1991; Chen et al., 1996; Mannila, 1997]. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

Mediante los modelos extraídos utilizando técnicas de minería de datos se aborda la solución a problemas de predicción, clasificación y segmentación.

En general, las tareas de la minería de datos se pueden clasificar en dos categorías: minería de datos descriptiva y minería de datos predictiva [Piatetski-Shapiro et al., 1996; Han, 1999].

Fases o procesos:

Un proceso típico de minería de datos consta de los siguientes pasos generales:

1. **Selección del conjunto de datos**, tanto en lo que se refiere a las variables dependientes, como a las variables objetivo.
2. **Análisis de las propiedades de los datos**, en especial la presencia de valores atípicos y ausencia de datos (valores nulos).
3. **Transformación del conjunto de datos de entrada**, se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos.
4. **Seleccionar y aplicar la técnica de minería de datos**, se construye el modelo predictivo, de clasificación o segmentación.
5. **Evaluar los resultados** contrastándolos con un conjunto de datos previamente reservado para validar la generalidad del modelo.

Técnicas:

Estas técnicas que provienen de la inteligencia artificial y de la estadística, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener resultados.

Usaremos el modelo estadístico que es una expresión simbólica en forma de igualdad o ecuación que se emplea en todo diseño experimental para indicar los diferentes factores que modifican la variable de respuesta.

Según el objetivo del análisis de los datos, los algoritmos utilizados se clasifican en supervisados y no supervisados.

- Algoritmos supervisados (o predictivos): predicen un dato (o un conjunto de ellos) desconocido a priori, a partir de otros conocidos.
- Algoritmos no supervisados (o deductivos): se descubren patrones y tendencias en los datos.

Criterios para aplicar los métodos de la minería de datos

- Factibilidad económica-organizativa: no se conocen métodos alternativos, no existen problemas de violación de la información.

- Factibilidad técnica: se dispone de suficientes datos, estos contienen rasgos relevantes, con poco ruido y se domina la aplicación de los métodos.

Utilidad de la minería de datos

Sabiendo que la minería constituye un paso más en el análisis de los datos para apoyar la toma de decisiones. Esta es una técnica de complemento, para un análisis más avanzado de los datos y extraer más información de ellos.

Además podemos citar los tiempos de respuestas, el manejo de gran volumen de datos para obtener un modelo válido es un inconveniente ya que implica gran cantidades de tiempo de procesos y se requiere un respuesta en tiempo real.

Gracias a la valiosa utilidad de la minería de datos hemos podido integrar algoritmos y resultados obtenidos en sistemas operacionales. De este modo, el conocimiento obtenido con estas técnicas no queda limitado por la visión que el usuario tiene del problema.

2.3. RECURSOS

DISPONIBILIDAD DE LA INFORMACIÓN

En la siguiente lista, se presentan una serie de recursos que resultan útiles en el desempeño de nuestra investigación.

- **Libros**

- T. Hastie, R. Tibshirani, J. Friedman "The elements of Statistical Learning: Data Mining, Inference, and Prediction". Este libro describe las ideas más importantes en el área de Data Mining manteniendo un marco conceptual con gran coherencia y elegancia.

- **Software y lenguajes de programación:**

- Smarty, PHP, Ajax, Postgress, XML

2.4. POSIBLES USUARIOS DEL SISTEMA

Este sistema combina el uso de dos usuarios, un administrador o usuario 1 que es aquel que únicamente está encargado de alimentar la base de datos, manipular, clasificar, organizar y realizar un análisis discriminante del usuario 2 o estudiante que es aquel que visualiza,

ingresa, selecciona y acepta según sus preferencias las materias según los horarios establecidos por el usuario 1.

Para mayor comprensión lo hemos graficado en el siguiente cuadro:

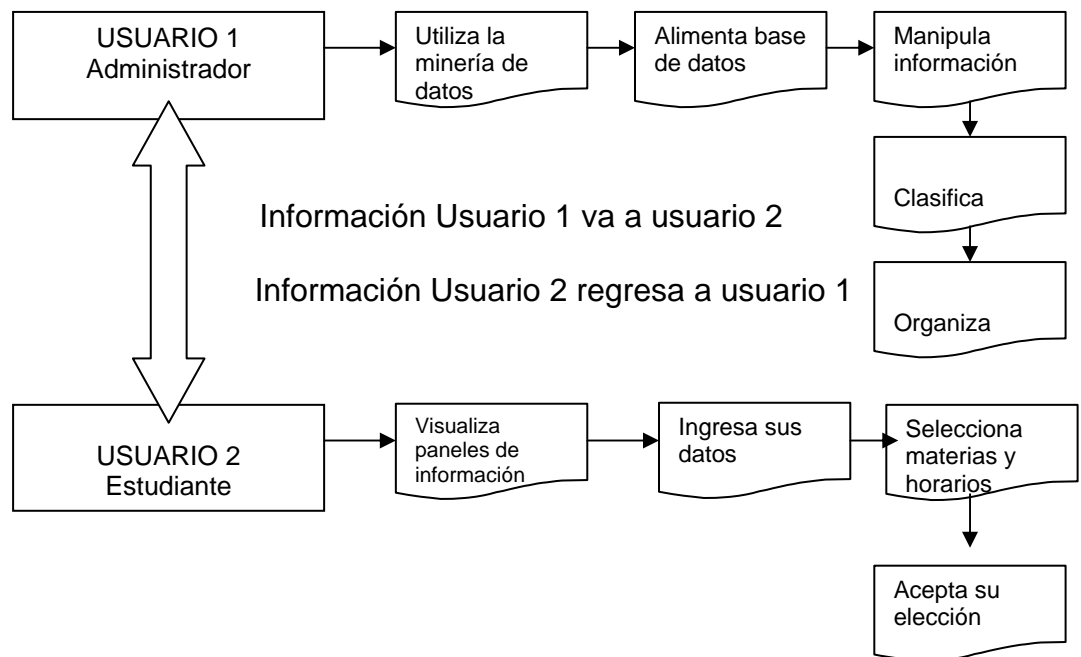


Figura 2.4. Posibles Usuarios del sistema.

2.5. ANÁLISIS DE LOS DATOS INGRESADOS.-

Los datos de los cuales disponemos de los estudiantes (provistos por la universidad) son los siguientes:

- Nombre del estudiante.
- Apellidos del estudiante.
- Matrícula del estudiante

- Factor P del estudiante
- Edad del estudiante
- Semestre en que está el estudiante
- Materias aprobadas por el estudiante
- Materias que actualmente está tomando el estudiante

Para realizar el análisis de clasificación de un estudiante, se dispone de algunos métodos relacionados con el análisis discriminante, el método escogido fue el análisis de Fisher.

La distribución F de Fisher. Una variable aleatoria se distribuye según el modelo de probabilidad **F de Fisher con (m,n) grados de libertad** , donde **m** y **n** son enteros positivos, si su función de densidad es la siguiente :

$$g(f) = \frac{\Gamma\left(\frac{m+n}{2}\right)\left(\frac{m}{n}\right)^{\frac{m}{2}} f^{\frac{m-2}{2}}}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)\left(1+\frac{m}{n}f\right)^{\frac{m+n}{2}}}$$

$$, f > 0, \text{ donde } \Gamma(p) = \int_0^{\infty} e^{-x} x^{p-1} dx$$

El modelo de probabilidad del cociente entre varianzas muestrales, en poblaciones normales e independientes. Si X_1, X_2, \dots, X_m ; e Y_1, Y_2, \dots, Y_n son muestras aleatorias de tamaños m y n extraídas de poblaciones normales $N(\mu_x, \sigma_x)$ y $N(\mu_y, \sigma_y)$, respectivamente, entonces

$\frac{\sigma_y^2 S_x^2}{\sigma_x^2 S_y^2}$ sigue una ley de probabilidad F de Fisher

con $(m-1, n-1)$ grados de libertad, donde $S_x^2 = \sum_{i=1}^m \frac{(X_i - \bar{X})^2}{m-1}$

y $S_y^2 = \sum_{j=1}^n \frac{(Y_j - \bar{Y})^2}{n-1}$ son las varianzas muestrales.

CAPÍTULO 3

FASES DEL PROYECTO

3.1. Ciclo virtuoso de la minería de datos

El proceso de la minería de datos es un ciclo, ya que los resultados obtenidos pueden alimentar nuevamente los procesos efectuados.

El ciclo virtuoso en esencia involucra:

Capturar, analizar, planear, actuar y revisar.

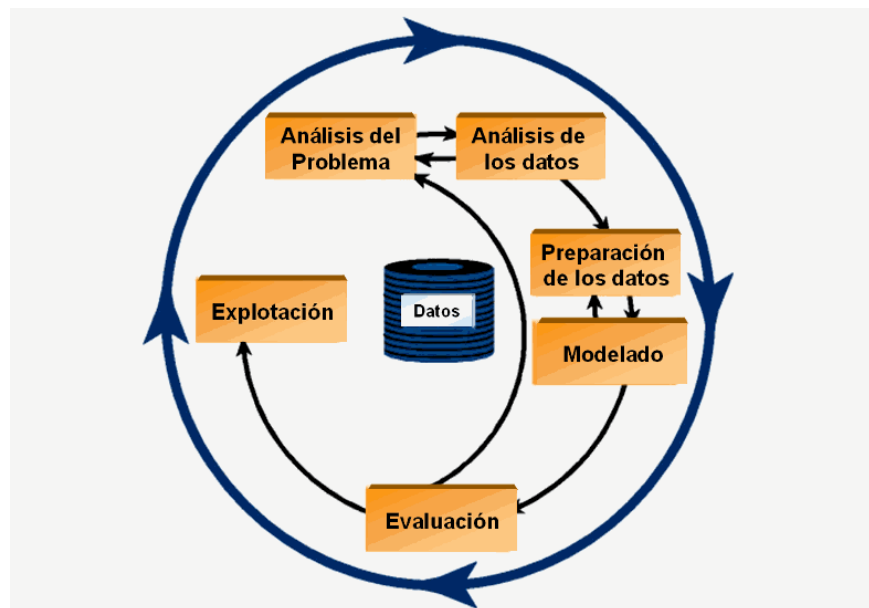


FIGURA 3.1.- Ciclo virtuoso de la minería de datos.

Fuente : <http://www.monografias.com/trabajos27/estimacion-coste/Image1100.gif>

Para realizar nuestro sistema hemos efectuado los siguientes pasos conforme el ciclo virtuoso de la minería de datos

3.2. Fases

3.2.1 Fase 1

Captura de información:

Para poder realizar la captura de la información debemos preocuparnos por identificar entre los datos disponibles, cuales de ellos son de interés para cumplir con los objetivos de nuestro trabajo. Recordando que existe información necesaria, la cual no podemos dejar a un lado. Además debemos observar qué datos son sensibles y afectan en mayor medida a los objetivos trazados.

Usando la base de datos de la ESPOL, donde se encuentra toda la información requerida de los estudiantes y partiendo desde el número de matrícula obtenemos información primordial como lo son:

- Factor P.
- Promedio.
- Edad.
- Nivel académico. (Ej: Nivel 100, Nivel 200)
- Historial de materias aprobadas.

Además tomamos los datos de preferencias de los estudiantes (en el sistema), esto nos ayudará a tener la opinión personalizada de los individuos.

Podemos disponer de un inmenso caudal de conocimiento basado en los datos obtenidos, sin embargo explotar dicho conocimiento es nuestro objetivo, de allí la idea de seleccionar la información necesaria entre toda la información disponible.

Una vez obtenida la información se procede al siguiente paso.

3.2.2 Fase 2.

Análisis:

Los datos que poseemos deben ser procesados y para esto involucramos lo siguiente:

- Tecnologías a usar: php, postgres, ajax, javascript, smarty.
- Orientación a objetivos: se define y se crean los objetivos.
- Modelamiento de la base de datos (para el respectivo almacenamiento de datos).

El análisis de los datos tienen su fundamento en los algoritmos aplicados para poder realizar la búsqueda de conocimiento previamente definido en la fase anterior. Estos algoritmos son planeados y elegidos con el fin de extraer de los datos el conocimiento fundamental para nuestro análisis.

Recordemos que estos algoritmos son aplicados de manera automática por el sistema con el único propósito de que se ejecute de manera

sistemática siguiendo los patrones establecidos, sin incluir la apreciación o punto de vista del operador o usuario en ese instante. Esta premisa debe ser aplicada, pues es fundamental en la minería de datos, obligatoriamente se necesita un trabajo automatizado, si existe alguna intervención humana durante el proceso éste ya no es considerado como minería de datos.

La razón por la cual se considera de importancia este punto radica en el objetivo de respetar la integridad del conocimiento extraído, para no incluir puntos de vista y apreciaciones que no han sido consideradas válidas durante la fase inicial. Se pone en práctica esta premisa pues el ser humano tiende a cambiar de opinión siendo influenciado por las circunstancias que lo rodean en un determinado instante o momento.

3.2.3 Fase 3

Planeación y Desarrollo.

En esta fase llevamos a ejecución lo propuesto en las fases anteriores buscando cumplir con los requerimientos establecidos para el diseño de las estructuras donde almacenaremos la información obtenida de los datos, respetando la consistencia y accesibilidad de la información.

Deseamos brindar la posibilidad de acceder a la información sin poder cambiarla o modificarla, y presentando los datos de manera comprensiva. Es por eso que procedemos al desarrollo de nuestra tesis con los siguientes pasos:

- Creación de base php.
- Creación del sitio web.
- Creación de clases en php y funciones en smarty.
- Creación de procesos en la base de datos.

Durante esta fase establecemos los procedimientos por medio de los cuales realizamos nuestro análisis discriminante (esto es transparente al usuario), generamos la información requerida para poder hacer predicciones.

Un proceso de minería de datos va más allá de la extracción y el análisis de los datos, también aplica el uso de estos datos para ser interpretados, ser almacenados y darle uso las veces que sea necesario para la toma de decisiones.

Pensando en el almacenamiento de los datos y el acceso a ellos, diseñamos tablas para guardar los análisis ya realizados para futuras consultas y toma de decisiones basada en la información obtenida.

3.2.4 Fase 4

Demostración y uso.

Pruebas realizadas para la verificación de los resultados antes de la respectiva sustentación.

Para esta fase ya disponemos de las estructuras de información obtenida a través de los datos previstos. Además ya contamos con nuestro sistema y las conexiones web establecidas para disponer de la información existente en nuestro repositorio de base de datos y la posibilidad de almacenar los análisis realizados.

Para realizar las pruebas necesitamos recordar que la implementación de nuestra tesis consta de 2 módulos:

- Módulo Estudiante.
- Módulo Administrador.

Pruebas realizadas:

MÓDULO ESTUDIANTE:

Se procede a las siguientes verificaciones:

- Ingreso de estudiante con su respectivo usuario y contraseña.

- Carga del flujo correspondiente a la carrera.
- Materias disponibles para elegir (Efecto drag and drop).
- Presentación de materia con opción de elegir profesor y horario deseado.
- Validaciones correspondientes a los horarios (número de horas).
- Posibilidad de poder modificar en caso de error antes de concluir con elección.
- Guardar preferencias de estudiante en el repositorio.

MÓDULO ADMINISTRADOR:

Verificaciones efectuadas:

- Ingreso de administrador con su usuario y contraseña.
- Carga de tablas de jornadas con la información establecida para el análisis a partir de día elegido, facultad y materia.
- Cálculo y presentación de los valores obtenidos por la función discriminante y sus respectivos centroides.
- Guardas análisis discriminante desarrollado.
- Carga de análisis anterior elegido para consulta.
- Verificación de estadística de profesores más elegidos y sus horarios.

3.3. Flujo de información en el ciclo de minería de datos.

En el ciclo virtuoso de la minería de datos los datos son procesados y almacenados de acuerdo al siguiente esquema, respetando siempre la integridad de los mismos.

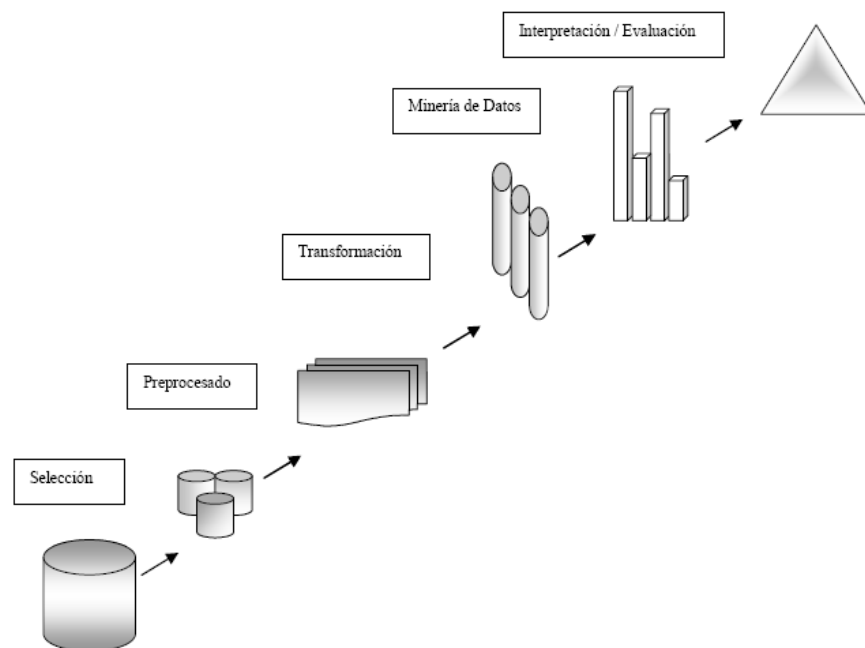


FIGURA 3.2.- Flujo de información en el ciclo virtuoso de la minería.

Además teniendo en cuenta la modernización que viven los registros académicos y siendo una necesidad la obtención de preferencias estudiantiles con respecto a la buena planificación de las materias cuya finalidad es la de nuestro proyecto.

Este trabajo luego de un análisis y viendo la necesidad del estudiante universitario utiliza los modelos discriminantes para el desarrollo de la clasificación de las preferencias, en el mismo se define los DATAMARTS donde se concentran los datos operativos, valores de hecho y conocimientos extraídos de los cuestionarios electrónicos. Además implementamos una interfaz Web necesaria para: ingreso de datos de los estudiantes, operación de los DATAMARTS y ambiente de toma de decisiones.

Lo que se quiere con un sistema como el nuestro es la de proporcionar la extracción del conocimiento de los estudiantes con respecto a las preferencias académicas a medida que va transcurriendo su ciclo de rol estudiantil.

Reflejar las tendencias de elección por parte de los alumnos; es una herramienta útil que facilite la toma de decisiones para los respectivos coordinadores. Para un futuro semestre; de que materias y que profesores los alumnos desean tomar se podría determinar con un grado de confiabilidad la cantidad de alumnos que desean tomar una materia para el próximo semestre tomando en cuenta dependencias estadísticas del mismo ya sea porque esta tomando una materia que le permite tomar otra después o por el status del mismo alumno si se

encuentra a prueba o esta es una materia que se encuentra en un nivel superior al siguiente semestre; sin embargo conocer las tendencias del alumnado por parte de los coordinadores facilitaría la labor de los mismos al abrir un tope de materias que se demandan.

El presente trabajo presenta un sistema que servirá de soporte para el coordinador a la toma de decisiones de este tipo de procesos con una respectiva toma de información y procesamiento de la misma para dar una selección óptima y eficiente de la cantidad demandada de materias y profesores para un siguiente semestre.

CAPÍTULO 4

ANÁLISIS DEL SISTEMA

En esta sección se realizará el análisis tanto de los requerimientos funcionales involucrados en el ciclo de vida del sistema.

4.1. Requerimientos funcionales.-

Aquí se definirán los requerimientos funcionales del sistema, los cuales delimitarán el alcance del proyecto e indicarán el comportamiento observable y la funcionalidad del mismo.

Esta aplicación nos facilita realizar un análisis discriminante con el cual podremos desarrollar una predicción y diseñar claramente planificación de materias para un respectivo semestre con el fin de simplificar el proceso actual de planificación; Para este objetivo se crea la parte administrativa con la cual se podrá guardar los datos generados en un Data Warehouse para poder hacer uso de los mismos en el análisis posterior.

Disponemos de dos tipos de usuarios:

- Usuario 1.- Administrador
- Usuario 2.- Estudiante

Tareas Usuario1 ó Administrador:

Teniendo claras las tareas que el Administrador puede realizar, las mismas que fueron mencionadas en el ítem **3.2.4**, en especial luego de realizar un análisis discriminante podrá obtener los datos para la función discriminante y el valor del centroide con el cual podrá realizar un análisis predictivo.

Previo al análisis discriminante el administrador podrá elegir la opción de realizar un análisis predictivo, a partir del cual podrá clasificar a un estudiante nuevo en alguno de los grupos ya creados, teniendo como seguridad la predicción ya realizada.

Como usuario Administrador podrá ingresar nuevos estudiantes al sistema o modificar a estudiantes ya existentes, pudiendo obtener los datos del estudiante en el sistema.

Los análisis y predicciones del Administrador podrán ser guardados en una base de datos para ser usados posteriormente si este desea hacerlo, por eso se le dará la opción de guardar las tareas realizadas durante su sesión.

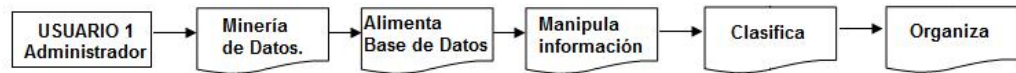


Figura 4.1. Tareas del Usuario Administrador.

Tareas Usuario2 ó Estudiante:

El Usuario2 al ingresar al sistema con su usuario y contraseña, se le cargará las opciones de planificación y contacto.

En la planificación se le cargará el flujo de materias respectivo a su carrera con el cual podrá hacer la selección de las materias que podrá tomar en su próximo semestre. Tendrá habilitada sólo las materias de las cuales dispondrá de acuerdo a los requerimientos respectivos a cada materia.

Una vez que haya escogido las materias se le presentará la opción de elegir con que profesor desea tomarla (asumiendo los profesores escogidos previamente para el dictado de la materia), y también la opción de elegir el horario en el que le gustaría tomar las clases (Validando las horas que se requieren para cada materia).

Con la opción de contactos el estudiante podrá informar de algún requerimiento especial o alguna anomalía en su registro al

administrador, por medio del cual se le enviará un e-mail de notificación al administrador del sistema.

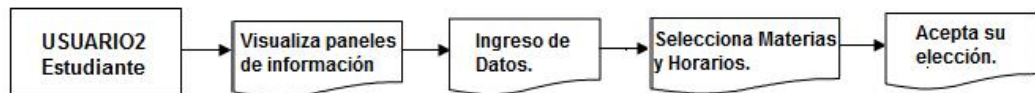


Figura 4.2. Tareas del Usuario Estudiante.

4.2. Requerimientos no funcionales

Luego de definir los requerimientos funcionales de nuestro sistema debemos también definir aquellos sistemas no funcionales, los mismos que se definen como aquellos que tienen que ver con la seguridad y confiabilidad de la información.

- **Validación de transacciones;** hace referencia al ingreso al sistema y a los cambios que se puedan realizar los diferentes tipos de usuarios del sistema en cada uno de sus ambientes.
- **Presentación de la información;** aquellos datos que son presentados al usuario final y que no se pueden prestarse a confusiones, mucho menos el ser erróneos.
- **Actualización de la información;** la actualización de datos de estudiantes y la clasificación de los mismos debe realizarse periódicamente.

- **Respaldo de la información;** Realizar un buen y respectivo respaldo de datos obtenidos, para así no tener pérdidas en la información..

4.3. Análisis de tecnologías

En lo que respecta al desarrollo de nuestro sistema hemos usado tecnologías tanto del lado del cliente, como del lado del servidor e incluso tecnologías cliente/servidor, que detallamos a continuación.

En lo que respecta al lenguaje de programación la siguiente tabla nos muestra las características y a que grupo pertenecen:

Lenguaje	Características	Tipo
JAVASCRIPT	Es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.	cliente
CSS (Cascading Style Sheets)	Es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS los resultados finales de la página son diferentes, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores... Incluso podemos definir nuestros propios estilos en un archivo externo a nuestras páginas; así, si en algún momento queremos cambiar alguno de ellos, automáticamente se nos actualizarán todas las páginas vinculadas de nuestro sitio.	cliente

PHP	Es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.	servidor
PL/PGSQL	Es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comando SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las funciones SQL básicas.	servidor
DHTML	<p>Una nueva capacidad de la que disponen los navegadores modernos, por la cual se puede tener un mayor control sobre la página que antes.</p> <p>Cualquier página que responde a las actividades del usuario y realiza efectos y funcionalidades se puede englobar dentro del DHTML, pero en este caso nos referimos más a efectos en el navegador por los cuales se pueden mostrar y ocultar elementos de la página, se puede modificar su posición, dimensiones, color, etc.</p> <p>DHTML nos da más control sobre la página, gracias a que los navegadores modernos incluyen una nueva estructura para visualizar en páginas web denominada capa. Las capas se pueden ocultar, mostrar, desplazar, etc.</p>	cliente/servidor
XML	XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.	cliente/servidor

Tabla 4.3.a. Lenguajes de Programación.

Lo que respecta al uso de un WebServer (Servidor Web) hemos usado el conocido APACHE, el mismo que es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan:

APACHE	<p>Multiplataforma</p> <p>Es un servidor de web conforme al protocolo HTTP/1.1</p> <p>Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.</p> <p>Basado en hebras en la versión 2.0</p> <p>Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.</p> <p>Se desarrolla de forma abierta</p> <p>Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.</p>
---------------	---

Tabla 4.3.b. Características del WebServer APACHE.

Siguiendo en la descripción de herramientas y tecnologías usadas en el desarrollo de nuestro proyecto, tenemos a continuación una breve descripción de nuestro sistema de base de datos relacional orientada objetos de software libre, postgresQL.

POSTGRESQL	<p>Implementación del estándar SQL92/SQL99.</p> <p>Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP ...), cadenas de bits, etc. También permite la creación de tipos propios.</p> <p>Incorpora una estructura de datos array.</p> <p>Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.</p> <p>Permite la declaración de funciones propias, así como la definición de disparadores.</p> <p>Soporta el uso de índices, reglas y vistas.</p> <p>Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.</p> <p>Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.</p>
-------------------	--

Tabla 4.3.c. Características de PostgreSQL.

En cuanto a las seguridades podemos citar que hemos escogido postgresSQL porque tiene la propiedad de soporte del protocolo de comunicación encriptado por SSL.

A continuación se muestra un ejemplo del funcionamiento de las páginas dinámicas y la interacción de los lenguajes de programación tanto en el lado del cliente como en el lado del servidor.

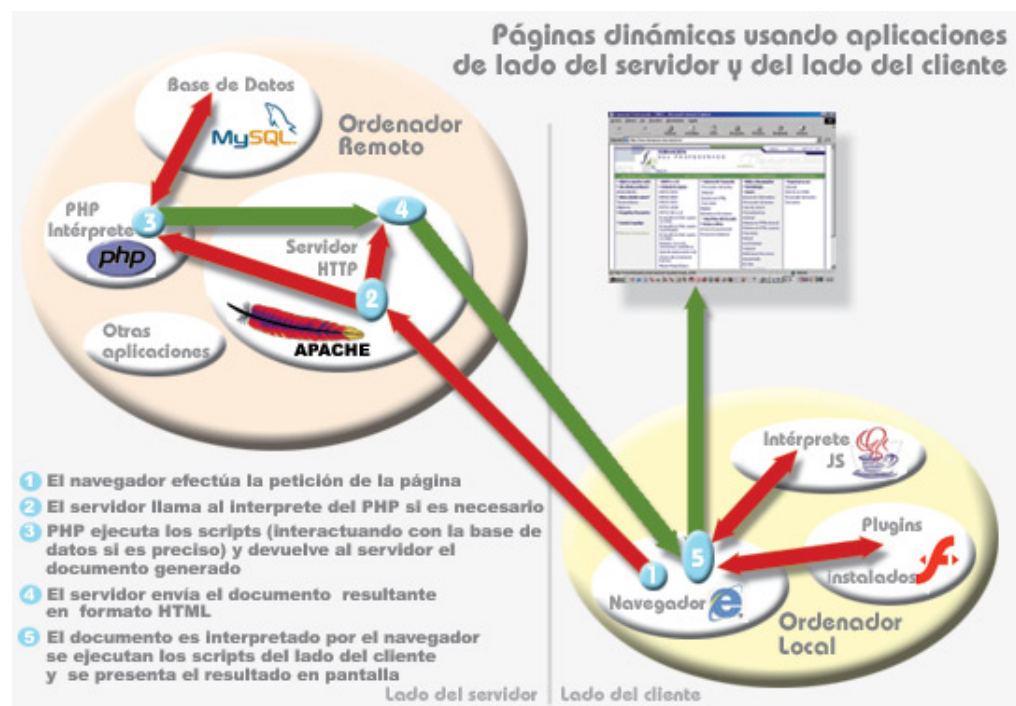


Figura 4.3. Aplicaciones del lado del servidor y del cliente.

Fuente: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

4.4. Descripción de casos de uso y escenarios

Este proyecto de tesis ha sido realizado con la finalidad de entregar una aplicación capaz de mejorar la planificación de materias tomando en cuenta las preferencias de los estudiantes según el nivel y las materias que estén dispuesto a tomar; para ello existen algunas secuencias a seguir detalladas a continuación.

Los casos de uso para el Sistema Académico Descriptivo Politécnico (**SADPOL**) son los siguientes:

- 1) Usuario ingresa al sistema
- 2) Usuario estudiante elije materias a planificar
- 3) Usuario estudiante elimina materias elegidas
- 4) Usuario estudiante elige profesor
- 5) Usuario estudiante realiza planificación
- 6) Usuario estudiante envía comentario al administrador
- 7) Usuario administrador realiza análisis descriptivo
- 8) Usuario administrador realiza análisis predictivo
- 9) Usuario administrador ingresa estudiante
- 10) Usuario administrador modifica estudiante
- 11) Usuario administrador guarda cálculos realizados

Nombre: 1. Usuario ingresa al sistema.

Descripción: Un usuario intenta ingresar al sistema a través de su nombre de usuario y contraseña.

Escenarios:

- 1.1 Acceso autorizado a un estudiante al sistema.
- 1.2 Acceso autorizado de un administrador al sistema.
- 1.3 Acceso no autorizado a una persona cuyo nombre de usuario es incorrecto.
- 1.4 Acceso no autorizado a una persona cuyo nombre de usuario es correcto, pero la contraseña es inválida.
- 1.5 Acceso no autorizado por error de procesamiento en la base de datos del sistema.

Nombre: 2. Usuario estudiante elige materias a planificar

Descripción: Un usuario estudiante elige sus materias para poder realizar planificación del semestre. El estudiante arrastra las materias desde donde esta el flujo dibujado hacia el área de selección.

Escenarios:

- 2.1 El estudiante arrastra de forma correcta las materias.
- 2.2 El estudiante no fija las materias sobre el área disponible.
- 2.3 El estudiante escoge materias que no están disponibles.
- 2.4 El flujo de la carrera de los estudiantes no es cargada de forma

correcta de la base de datos.

Nombre: 3. Usuario estudiante elimina materias elegidas

Descripción: Un usuario estudiante elimina las materias seleccionadas para planificación del semestre. El usuario arrastra las materias desde el área de selección hacia el área de eliminación..

Escenarios:

- 3.1 El estudiante arrastra de forma correcta las materias.
- 3.2 El estudiante no fija las materias sobre el área disponible.
- 3.3 El estudiante escoge materias que no están dentro del área seleccionada.
- 3.4 El proceso de arrastrar no está habilitado.

Nombre: 4. Usuario estudiante elige profesor

Descripción: Un usuario estudiante escoge al profesor con el que desea tomar la materia seleccionada.

Escenarios:

- 4.1 El estudiante marca la casilla de verificación del profesor que desee.
- 4.2 El estudiante no marca de manera apropiada la casilla de verificación.
- 4.3 No existen profesores disponibles para realizar la selección.

Nombre: 5. Usuario estudiante realiza planificación

Descripción: Un usuario estudiante escoge el horario durante la semana en el cual desea tener las clases de su materia dando clic en cada casillero del horario (Teniendo como referencia el limitante de horas de la materia).

Escenarios:

5.1 El estudiante marca la casilla de horarios que desea.

5.2 El estudiante no marca de manera apropiada la casilla de horario.

5.3 El estudiante excede de las horas correspondientes a la materia.

5.4 El estudiante sobrepasa las horas correspondientes a la materia.

5.5 No existen horarios disponibles para realizar la selección.

Nombre: 6. Usuario estudiante envía comentario al Administrador

Descripción: Un usuario estudiante envía una observación al Administrador del sistema indicando algún caso en particular que no haya sido tomado en cuenta durante el análisis (algún comentario de apreciación).

Escenarios:

6.1 El estudiante envía comentario al Administrador de manera

apropiada.

6.2 El estudiante no ingresa su nombre en el formulario.

6.3 El estudiante no ingresa su correo en el formulario.

6.4 El estudiante no ingresa su comentario en el formulario.

6.5 El formulario no puede ser enviado por fallas en la conexión.

Nombre: 7. Usuario Administrador realiza análisis descriptivo

Descripción: Un usuario administrador mediante el sistema realiza el cálculo del análisis descriptivo generando la función discriminante y los centroides a partir de las materias establecidas en un flujo.

Escenarios:

7.1 El Administrador tiene los datos apropiados y realiza el análisis.

7.2 El Administrador no dispone de los datos de la materia para generar análisis.

7.3 El Administrador no posee la matriz de variables para generar análisis.

7.4 El análisis no es generado por problemas en el cálculo estadístico.

Nombre: 8. Usuario Administrador realiza análisis predictivo

Descripción: Un usuario Administrador mediante el sistema realiza el cálculo del análisis predictivo por medio de la función discriminante y los centroides que ha calculado anteriormente o que tiene guardados.

Escenarios:

- 8.1 El administrador tiene los datos apropiados y realiza predicción.
- 8.2 El administrador no dispone de los datos necesarios para la predicción.
- 8.3 El análisis no es generado por problemas en el cálculo estadístico.

Nombre: 9. Usuario Administrador ingresa estudiante

Descripción: Un usuario administrador ingresa los datos de un estudiante nuevo al sistema.

Escenarios:

- 9.1 El Administrador ingresa de forma apropiada los datos en el formulario.
- 9.2 El Administrador no ingresa los datos de forma apropiada en el formulario.
- 9.3 El formulario no genera el flujo correspondiente a la carrera por problemas en conexión a la base de datos.

Nombre: 10. Usuario Administrador modifica estudiante

Descripción: Un usuario administrador modifica los datos de un estudiante existente en el sistema.

Escenarios:

- 10.1 El Administrador ingresa de forma apropiada los datos nuevos en el formulario.
- 10.2 El Administrador no ingresa los datos de forma apropiada en el formulario.
- 10.3 El formulario no genera el flujo correspondiente a la carrera por problemas en conexión a la base de datos.
- 10.4 El Administrador no guarda los datos de la modificación realizada al estudiante existente por problemas de conexión a la base de datos.

Nombre: 11. Usuario Administrador guarda cálculos realizados

Descripción: Un usuario Administrador guarda los cálculos realizados sea de descripción o predicción en el sistema.

Escenarios:

- 11.1 El Administrador guarda el análisis realizado en su sesión.
- 11.2 El Administrador no guarda el análisis realizado en su

sesión.

11.3 El Administrador no guarda el análisis realizado en su sesión por problemas en conexión a la base de datos.

4.5. Análisis de la interacción Hombre - Máquina

Si bien es cierto, la Interacción Hombre - Máquina trata acerca del buen diseño de sistemas computacionales que les permita a los usuarios desarrollar sus actividades de manera productiva y segura. Luego de haber realizado un análisis de todas las herramientas y tecnologías usadas en el desarrollo de nuestro proyecto podemos concluir que nuestro sistema cumple con los principios fundamentales de la Interacción Hombre-Máquina, como los son:

- Visibilidad.
- Permisividad.
- Usabilidad.
- Consistencia.
- Navegabilidad.
- Retroalimentación.
- Interfaz significativa.
- Seguridad.

- Diseño minimalista.
- Codificación gráfica.

Describiendo brevemente dichos principios y haciendo referencia a nuestro proyecto, tenemos:

Visibilidad: Todas las opciones y controles del sistema deben ser fácilmente distinguibles para los usuarios. El uso de herramientas como AJAX y JavaScript nos permite que el usuario vea de una forma agradable y sencilla el manejo del sistema.

Permisividad (affordance): Sabiendo que se refiere a las manipulaciones de objetos para obtener un resultado fácil de entender, gracias al uso de lenguajes de programación como DHTML se pudo proporcionar nombres de menús y opciones, así como de controles que tengan concordancia con lo que el usuario espera como resultado de utilizar estos objetos.

Usabilidad: El sistema es fácil de aprender y de usar, especialmente para novatos, con el objetivo de que las tareas de los usuarios sean más eficientes, efectivas y seguras. Por ejemplo, cuando los usuarios necesiten hacer ingresos de nuevos registros, sólo debe ser necesario

llenar los campos que se requieren, enviar la información, y obtener el resultado de la operación en pocos pasos.

Consistencia: El diseño del sistema guarda consistencia tanto con los estándares de diseño Web bajo los cuales los usuarios se encuentran familiarizados, así como dentro del mismo sistema, esto es que todos los colores, mensajes de error y resultado, fuentes y tamaños de las letras, enlaces y controles siguen un mismo formato en todas las páginas del sistema.

Navegabilidad: El sistema proporciona que los menús permitan la exploración general de todas las opciones que brinda el sistema a través de hipervínculos fáciles de identificar. Se ha escogido el uso de menús pull-down para presentar las opciones de los menús.

Retroalimentación (feedback): El sistema indica al usuario en qué sesión se encuentra, qué acción ha sido ejecutada o se encuentra en estado de ejecución, y cuáles son las operaciones permitidas a las que tiene acceso.

Interfaz significativa (familiaridad): Este principio se relaciona con la importancia de que los usuarios puedan familiarizarse con los objetos y

la interface del sistema una vez que ya lo han aprendido a utilizar, proporcionando nombres significativos (fáciles de recordar) de opciones y controles.

Seguridad: El sistema no permitirá ingresar a páginas internas digitando su URL mientras no haya iniciado una sesión. Además el uso de postgres hace que la comunicación sea seguro mediante el uso del protocolo de encriptación.

Para el reconocimiento de errores se proporcionan mensajes claros, en lenguaje natural, y fáciles de entender cuando se produzca un error.

Diseño minimalista: El sistema presenta a los usuarios únicamente la información relevante, como el estado del sistema y la operación que se va o está ejecutando.

Codificación gráfica: El uso de los colores dentro del diseño de las páginas Web permite al usuario establecer una separación entre la cabecera, los menús con sus opciones, y la parte central donde se muestran las operaciones y resultados del sistema. Hemos evitado el uso excesivo de colores (polución de colores) y la combinación de colores fuertes por la dificultad de interpretar la interfase del sistema que esto conlleva.

CAPÍTULO 5

DISEÑO Y MANEJO DEL SISTEMA

5.1. Diseño de la arquitectura del sistema

Arquitectura de las 3 capas

Generalmente la arquitectura se refiere a dividir por partes la funcionalidad de la función dentro de componentes separados basados en que es lo que harán y agrupándolos en clase de componentes dentro de una simple capa lógica.

La mayoría de los módulos que se crearon para el sistema tienen componentes que están en 3 capas desde la aplicación del servidor:

- La capa de presentación.
- La capa de negocio
- La capa de los datos.

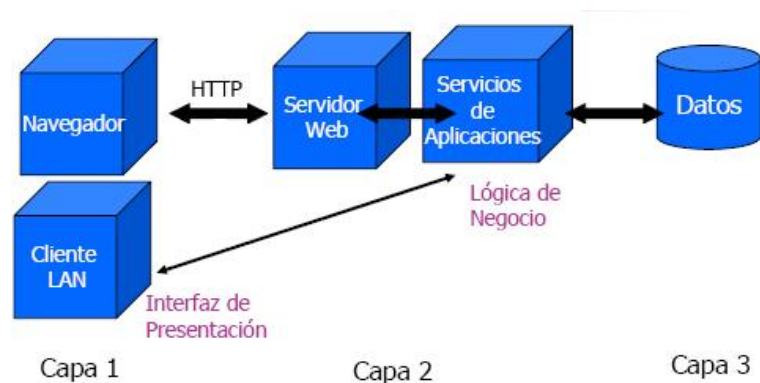


Figura 5.1. Arquitectura de las 3 capas.

Fuente: <http:// triana.escet.urjc.es/>

La capa de presentación es aquella que contiene los elementos de la interfaz de usuario del sitio e incluye toda la lógica que manejará la interacción del visitante o el usuario. Esta capa hace que todo el sitio se vea vivo y la manera del diseño es una parte crucial para el éxito del sistema porque nuestra aplicación es un sitio web y su capa de presentación está compuesta por páginas web dinámicas.

La capa de negocios (también llamada la capa intermedia) recibe pedidos desde la capa de presentación y retorna un resultado a la capa de presentación dependiendo sobre la lógica del negocio que esta contenga.

Casi cualquier evento que suceda en la capa de presentación usualmente resulta un llamado a la capa de negocios (excepto eventos que son manejados localmente por la capa de presentación como un simple ingreso de datos, etc). Casi siempre la capa de negocios necesita llamar a la capa de datos para información que se encuentra disponible para responder a un pedido de la capa de presentación.

La Capa de datos (a veces también llamada como, capa de base de datos), es responsable de manejar los datos de la aplicación y enviar a la capa de negocios cuando esta lo solicita, casi cada cliente pide un

resultado en la capa de datos que está siendo procesada para información (excepto cuando estos datos ya se encuentran almacenados en cache a nivel de la capa de negocios o a nivel de la capa de presentación, es muy importante tener un sistema de base de datos veloz.

Estas capas son puramente lógicas – no hay limitantes sobre la ubicación física de cada capa.

En teoría existe libertad de colocar la aplicación e implícitamente todas sus capas en un único servidor o se pueden colocar cada capa en separados servidores si la aplicación lo permite. Una importante limitante en el modelo de arquitectura de 3 capas es que la información debe fluir en secuencia y en un orden entre capas. La capa de presentación solo permitirá el acceso a la capa de negocios y no puede acceder directamente a la capa de datos.

La capa de negocios es el “cerebro” en la mitad de la comunicación con las otras capas y procesa y coordina todo el flujo de información si es que la capa de información accede directamente a la capa de datos, las reglas de la arquitectura de 3 capas serán rotas. Cuando se incrementa una arquitectura de 3 capas se debe estar consciente y obedecer sus reglas para adquirir sus beneficios.

5.2. La correcta lógica para una correcta capa

Ya que cada capa contiene su propia lógica, a veces puede resultar un poco tedioso, cuando exactamente dibujar una línea entre las capas. En un escenario por ejemplo, en lugar de leer el usuario valido de la capa de negocios y decidir si es un usuario valido o no dependiendo de su estado se puede tener un método de la capa de datos llamado “es usuario” y leer directamente sin la capa de negocios.

En muchas otras circunstancias se podría tener la opción de reemplazar alguna lógica de alguna capa en otra o en ambas. Más aún, existen ocasiones en las que incluso pensamos que es la manera correcta (con respecto a la arquitectura) para implementar algo, podrías escoger romper la regla para ganar algo de rendimiento. Como una regla general si es que el rendimiento se mejora de esta manera, esta bien los límites establecidos entre las capas, pero solo un poco. De otra manera, mantener todas las reglas del negocio en la capa del medio, es preferible porque genera una aplicación limpia y fácil de mantener.

5.3. Arquitectura de SADPOL⁽²⁾.

Implementar una arquitectura 3 capas ayudará a alcanzar los objetivos mencionados de rendimiento, flexibilidad y mantenimiento. La disciplina de código impuesto para este sistema puede parecer un poco rígida a primera vista. Separando las mayores partes de la aplicación en separados componentes pequeños, permitirá su reusabilidad.

Otra ventaja de la arquitectura de 3 capas, es que si se la implementa correctamente todo el sistema será flexible para los cambios. Cuando se hace un pequeño cambio en una capa, las otras capas usualmente no resultan afectadas a veces, incluso casos extremos. Por ejemplo, si por alguna razón el sistema de base de datos es cambiado (el administrador decide usar Oracle en lugar de PostgreSQL) lo único que necesita es actualizar un poco la capa de datos al igual que la capa de negocios.

Para diseñar y crear el sistema se utiliza.-

- **Php** para generar contenido web dinámico
- **Smarty** para separar plantillas del código.
- **Postgresql** para almacenar datos en el website
- Se implementa la integración de la base de datos usando **adodb**

⁽²⁾SADPOL.- Sistema de Análisis Discriminante Politécnico.

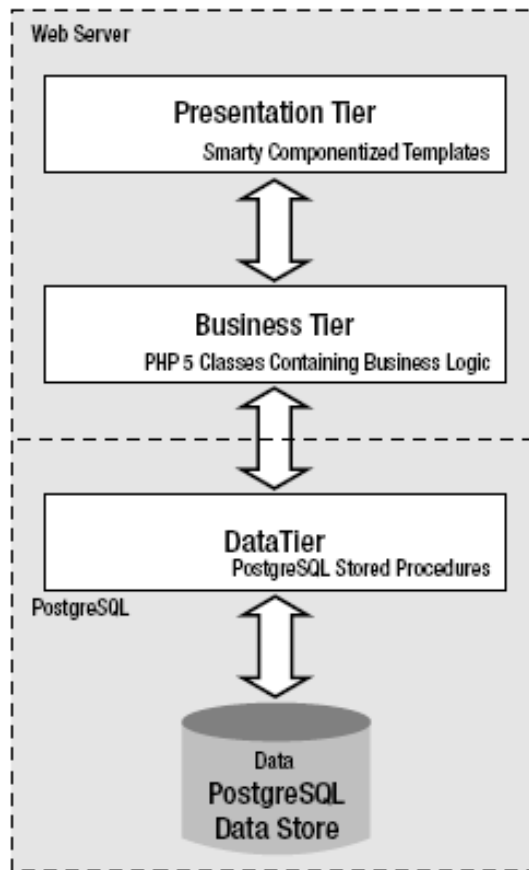


Figura 5.2. Herramientas para diseño del sistema.
Fuente: <http://geekswithblogs.net>

5.4 DISEÑO DEL ESQUEMA ADMINISTRADOR

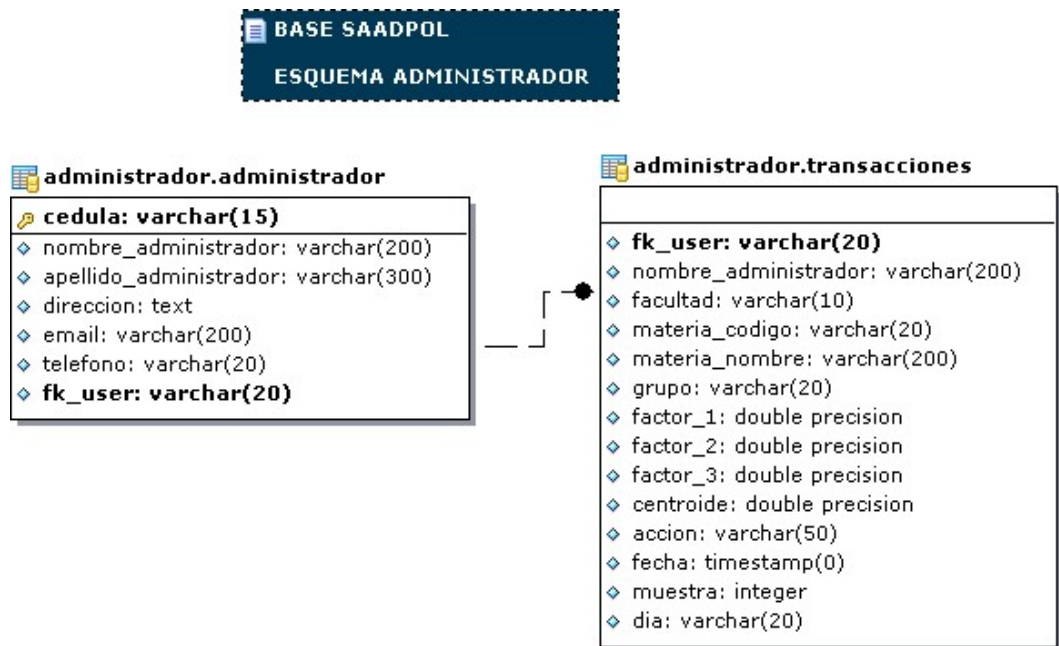


Figura 5.3. Modelo de datos del esquema administrador.

5.5 DISEÑO DEL ESQUEMA ESTUDIANTE

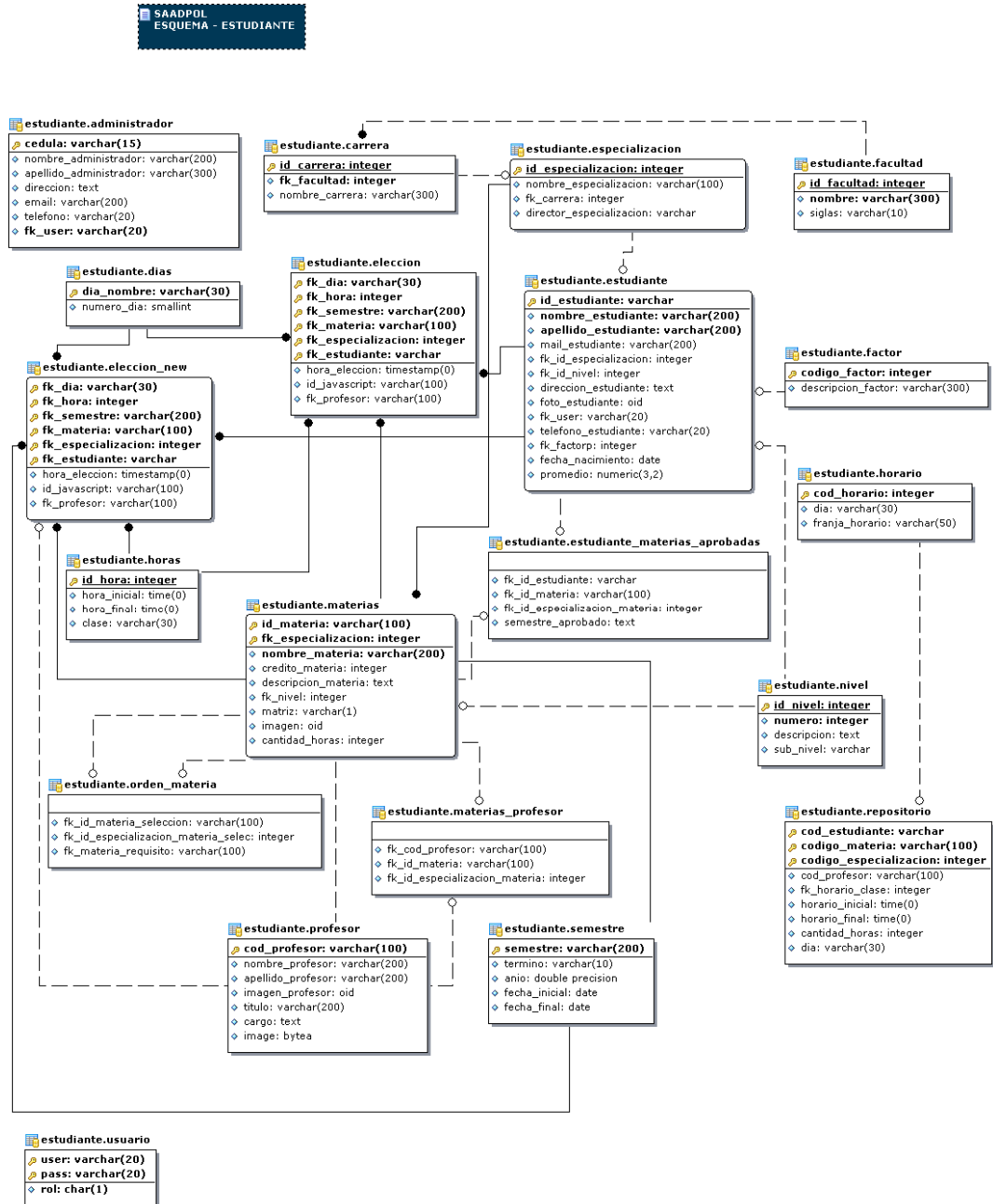


Figura 5.4. Modelo de datos del esquema estudiante.

El esquema de Administrador, tal como se aprecia en la **Figura 5.4.**

poseé las siguientes tablas:

- Administrador.administrador
- Administrador.transacciones

La tabla Administrador.administrador es la que almacena la información de los datos del administrador del sistema.

Administrador.administrador	
Campo.	Descripción.
cedula	Clave primaria. Campo de tipo Varchar (), con longitud máxima de 15 caracteres, almacena el número de cédula del usuario identificado como administrador.
nombre_administrador	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre del usuario identificado como administrador.
apellido_administrador	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena el apellido del usuario identificado como administrador.
direccion	Campo de tipo Text (), almacena la dirección del usuario identificado como administrador.
email	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el mail del usuario identificado como administrador.
telefono	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena el número telefónico del usuario identificado como administrador.
fk_user	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena identificador propio de la tabla transacciones. Es la clave para poder acceder a la tabla mencionada.

Tabla 5.4.a. Tabla Administrador.administrador.

La tabla `Administrador.transacciones` es la que almacena la información de todas las transacciones (análisis y predicciones) que haya realizado el administrador en el sistema. Tiene los siguientes campos:

Administrador.transacciones	
Campo.	Descripción.
fk_user	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena identificador propio de la tabla transacciones. Es la clave para poder acceder a la tabla mencionada.
nombre_administrador	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre del usuario identificado como administrador.
facultad	Campo de tipo Varchar (), con longitud máxima de 10 caracteres, almacena el nombre de la facultad a la cual pertenece la materia a la que se le realizó el análisis discriminante.
materia_codigo	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena el código de la materia a la que se le realizó el análisis discriminante.
materia_nombre	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre de la materia a la que se le realizó el análisis discriminante.
grupo	Campo de tipo Varchar(), con longitud máxima de 20 caracteres, almacena el nombre del grupo al cual se le efectuó el análisis discriminante.
factor_1	Campo de tipo Double (), almacena el valor numérico número uno correspondiente a la función discriminante la cual es obtenida en el análisis realizado por el administrador.
factor_2	Campo de tipo Double (), almacena el valor numérico número dos correspondiente a la función discriminante la cual es obtenida en el análisis realizado por el administrador.
factor_3	Campo de tipo Double (), almacena el valor numérico número tres correspondiente a la función discriminante la cual es obtenida en el análisis realizado por el administrador.
centroide	Campo de tipo Double (), almacena el valor numérico correspondiente al centroide obtenido en el cálculo discriminante.

accion	Campo de tipo Varchar(), con longitud máxima de 50 caracteres, almacena el tipo de análisis que fue efectuado por el administrador.
fecha	Campo de tipo timestamp(), almacena la fecha del sistema en el que fue realizado el análisis por el administrador.
muestra	Campo de tipo Integer(), guarda el número de individuos para el cual se realizó el análisis discriminante.
dia	Campo de tipo Varchar(), con longitud máxima de 20 caracteres, almacena el nombre del día para el cual fue hecho el análisis discriminante.

Tabla 5.4.b. Tabla Administrador.transacciones

El esquema de Estudiante, tal como se aprecia en la **Figura 5.5**, posee las siguientes tablas:

Estudiante.administrador

Estudiante.carrera

Estudiante.especializacion

Estudiante.facultad

Estudiante.dias

Estudiante.eleccion

Estudiante.estudiante

Estudiante.factor

Estudiante.eleccion_new

Estudiante.horario

Estudiante.horas

Estudiante.estudiante_materias_aprobadas

Estudiante.materias

Estudiante.nivel

Estudiante.orden_materia

Estudiante.materias_profesor

Estudiante.repositorio

Estudiante.profesor

Estudiante.semestre

Estudiante.usuario

Estudiante.prueba

La tabla ESTUDIANTE.administrador es la que almacena la información de los datos del administrador del sistema. Tiene los siguientes campos:

Estudiante.administrador	
Campo.	Descripción.
cedula	Clave primaria. Campo de tipo Varchar (), con longitud máxima de 15 caracteres, almacena el número de cédula del usuario identificado como administrador.
nombre_administrador	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre del usuario identificado como administrador.
apellido_administrador	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena el apellido del usuario identificado como administrador.
direccion	Campo de tipo Text (), almacena la dirección del usuario identificado como administrador.
email	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el mail del usuario identificado como administrador.
telefono	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena el número telefónico del usuario identificado como administrador.

fk_user	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena identificador propio de la tabla transacciones. Es la clave para poder acceder a la tabla mencionada.
---------	--

Tabla 5.5.a. Tabla Estudiante.administrador

La tabla Estudiante.carrera es la que almacena la información de los datos de la carrera del estudiante. Tiene los siguientes campos:

Estudiante.carrera	
Campo.	Descripción.
Id_carrera	Clave primaria. Campo de tipo Integer (), almacena un identificar numérico que corresponde a cada carrera (este valor es único para cada carrera).
Fk_facultad	Campo de tipo Integer (),almcena un identificador de la facultad a la que pertenece la carrera, es una clave para tener acceso a la tabla facultad.
Nombre_carrera	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena el nombre de la carrera.

Tabla 5.5.b. Tabla Estudiante.carrera.

La tabla Estudiante.especializacion es la que almacena la información de los datos de la especialización del estudiante. Tiene los siguientes campos:

Estudiante.especializacion	
Campo.	Descripción.
Id_especializacion	Clave primaria. Campo de tipo Integer (), almacena un identificar numérico que corresponde a cada especialización (este valor es único para cada especialización).
Fk_carrera	Campo de tipo Integer (), almacena un identificador de la carrera a la que pertenece la especialización, es una clave para tener acceso a la tabla carrera.

Nombre_ especializacion	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena el nombre de la especialización.
Director_ especializacion	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena el nombre del director de la especialización.

Tabla 5.5.c. Tabla Estudiante.especializacion.

La tabla Estudiante.facultad es la que almacena la información de los datos de la facultad del estudiante. Tiene los siguientes campos:

Estudiante.facultad	
Campo.	Descripción.
Id_facultad	Clave primaria. Campo de tipo Integer (), almacena un identificar numérico que corresponde a cada facultad (este valor es único para cada facultad).
Nombre_ facultad	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena el nombre de la facultad.
Siglas	Campo de tipo Varchar (), con longitud máxima de 10 caracteres, almacena las siglas de la facultad.

Tabla 5.5.d. Tabla Estudiante.facultad

La tabla Estudiante.dias es la que almacena la información de los datos de los días de la semana. Se creó para dar un valor numérico a cada día de la semana. Tiene los siguientes campos:

Estudiante.dias	
Campo.	Descripción.
Id_nombre	Clave primaria. Campo de tipo Varchar (), con longitud máxima de 30 caracteres, almacena los nombres de los días de la semana.
Numero_dia	Campo de tipo Smallint (), almacena un numero identificador para cada día de la semana.

Tabla 5.5.e. Tabla Estudiante.dias.

La tabla Estudiante.eleccion es la que almacena la elección de las materias que el estudiante desea tomar el próximo semestre. Tiene los siguientes campos:

Estudiante.eleccion	
Campo.	Descripción.
Fk_dia	Campo de tipo Varchar (), con longitud máxima de 30 caracteres, almacena un identificador del día para poder tener acceso a la tabla día.
Fk_hora	Campo de tipo Integer (), almacena un identificador de la hora para poder tener acceso a la tabla hora.
Fk_semestre	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena un identificador del semestre para poder tener acceso a la tabla semestre.
Fk_materia	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador de la materia para poder tener acceso a la tabla materia.
Fk_especializacion	Campo de tipo Integer (), almacena un identificador de la especialización para poder tener acceso a la tabla especialización.
Fk_estudiante	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador del estudiante para poder tener acceso a la tabla estudiante.
hora_eleccion	Campo de tipo timestamp (), almacena la hora del sistema en el que el usuario estudiante realizó su selección de materia.
id_javascript	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificar con la inicial del día y la hora que seleccionó el estudiante su horario de preferencias.
fk_profesor	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador del profesor para poder tener acceso a la tabla profesor.

Tabla 5.5.f. Tabla Estudiante.eleccion.

La tabla Estudiante_estudiante es la que almacena los datos del estudiante que desea hacer sus predicciones, contiene campos:

Estudiante_estudiante	
Campo.	Descripción.
Id_estudiante	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador el cual es único para cada estudiante.
nombre_estudiante	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre del usuario identificado como estudiante.
apellido_estudiante	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el apellido del usuario identificado como estudiante.
mail_estudiante	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el mail del usuario identificado como estudiante.
fk_id_especializacion	Campo de tipo Integer (), almacena un identificador de la especialización para poder tener acceso a la tabla especialización.
fk_id_nivel	Campo de tipo Integer (), almacena un identificador del nivel para poder tener acceso a la tabla nivel.
direccion_estudiante	Campo de tipo Text (), almacena la dirección del usuario identificado como estudiante.
foto_estudiante	Campo de tipo Oid, almacena la foto del usuario identificado como estudiante.
fk_user	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena identificador propio de la tabla transacciones. Es la clave para poder acceder a la tabla mencionada.
telefono_estudiante	Campo de tipo Varchar (), con longitud máxima de 20 caracteres, almacena el número telefónico del usuario identificado como estudiante.
fk_factorp	Campo de tipo Integer (), almacena identificador propio de la tabla factor. Es la clave para poder acceder a la tabla mencionada.
fecha_nacimiento	Campo de tipo Date, determina la fecha de nacimiento de nacimiento.
Promedio	Campo de tipo numeric (3,2), registra el promedio del estudiante.

Tabla 5.5.g. Tabla Estudiante_estudiante.

La tabla estudiante_factor es la que almacena los datos del factor p al cual se le va a dar valor a cada estudiante. Tiene los siguientes campos:

Estudiante_factor	
Campo.	Descripción.
codigo_factor	Clave primaria. Campo de tipo Integer (), almacena un número correspondiente a cada identificar de factor p, este valor es único para cada factor p.
descripcion_factor	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena la descripción correspondiente al factor p.

Tabla 5.5.h. Tabla Estudiante_factor

La tabla estudiante_eleccion_new: es la que almacena los datos de una elección nueva realizada por cada estudiante. Tiene los siguientes campos:

Estudiante_eleccion_new	
Campo.	Descripción.
Fk_dia	Campo de tipo Varchar (), con longitud máxima de 30 caracteres, almacena un identificador del día para poder tener acceso a la tabla día.
Fk_hora	Campo de tipo Integer (), almacena un identificador de la hora para poder tener acceso a la tabla hora.
Fk_semestre	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena un identificador del semestre para poder tener acceso a la tabla semestre.
Fk_materia	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador de la materia para poder tener acceso a la tabla materia.
Fk_especializacion	Campo de tipo Integer (), almacena un identificador de la especialización para poder tener acceso a la tabla especialización.

Fk_estudiante	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador del estudiante para poder tener acceso a la tabla estudiante.
hora_eleccion	Campo de tipo timestamp (), almacena la hora del sistema en el que el usuario estudiante realizó su selección de materia.
id_javascript	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador con la inicial del día y la hora que seleccionó el estudiante su horario de preferencias.
fk_profesor	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador del profesor para poder tener acceso a la tabla profesor.

Tabla 5.5.i. Tabla Estudiante_eleccion_new.

La tabla estudiante.horario es la que almacena los datos del horario que es parte de la selección que hace el estudiante en sus preferencias. Tiene los siguientes campos:

Estudiante.horario	
Campo.	Descripción.
codigo_horario:	Clave primaria. Campo de tipo Integer (), almacena un número correspondiente a cada identificador del horario, este valor es único para cada horario.
descripcion_factor:	Campo de tipo Varchar (), con longitud máxima de 300 caracteres, almacena la descripción correspondiente al factor p.

Tabla 5.5.j. Tabla Estudiante.horario.

La tabla estudiante.horas es la que almacena los datos del inicio y el final de cada horario establecido que es parte de la selección que hace el estudiante en sus preferencias. Tiene los siguientes campos:

estudiante.horas	
Campo.	Descripción.
id_hora	Clave primaria. Campo de tipo Integer (), almacena un número correspondiente a cada identificar de hora, este valor es único para cada hora.
hora_inicial	Campo de tipo Time (), almacena la hora de inicio de cada espacio horario.
hora_final	Campo de tipo Time (), almacena la hora final de cada espacio horario.
Clase	Campo de tipo Varchar (), con longitud máxima de 30 caracteres, almacena la clase a la que corresponde el horario.

Tabla 5.5.k. Tabla Estudiante.horas.

La tabla Estudiante.estudiante_materias_aprobadas es la que almacena los datos de las materias aprobadas por cada estudiante que desea hacer sus predicciones. Tiene los siguientes campos:

Estudiante.estudiante_materias_aprobadas	
Campo.	Descripción.
fk_Id_estudiante	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificar el cual sirve para tener acceso a la tabla estudiante.
fk_id_materia	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un identificador el cual sirve para tener acceso a la tabla materia.
fk_id_especializacion_materia	Campo de tipo Integer (), almacena un identificador el cual sirve para tener acceso a la tabla especialización de la materia.
semestre_aprobado	Campo de tipo Text (), almacena el último semestre aprobado por el estudiante.

Tabla 5.5.I. Tabla Estudiante.estudiante_materias_aprobadas.

La tabla Estudiante.materias es la que almacena los datos de las materias correspondientes a cada flujo de cada carrera. Tiene los siguientes campos:

Estudiante.materias	
Campo.	Descripción.
id_materia:	Campo de tipo Varchar (), con longitud máxima de 100 caracteres, almacena un valor único el cual es clave primaria para identificar a cada materia.
fk_especializacion:	Campo de tipo Integer (), almacena un identificador de la especialización para poder tener acceso a la tabla especialización.
nombre_materia:	Campo de tipo Varchar (), con longitud máxima de 200 caracteres, almacena el nombre de cada materia correspondiente a un flujo.
credito_materia:	Campo de tipo Integer (), almacena un valor el cual corresponde al número de créditos que tiene cada materia fijado por las facultades.
descripcion_materia:	Campo de tipo Text (), almacena un comentario correspondiente a la materia a la cual se refiere.
fk_nivel:	Campo de tipo Integer (), almacena un identificador del nivel para poder tener acceso a la tabla nivel.
matriz:	Campo de tipo Varchar (), almacena un identificador para esta tabla materias.
imagen:	Campo de tipo Oid, almacena una imagen correspondiente a la materia.
cantidad_horas:	Campo de tipo Integer (), almacena el número de hora que le corresponde a la materia.

Tabla 5.5.m. Tabla Estudiante.materias.

La tabla `Estudiante.nivel` es la que almacena los datos correspondientes a nivel del flujo de cada carrera. Tiene los siguientes campos:

Estudiante.nivel	
Campo.	Descripción.
id_nivel:	Campo de tipo Integer (), almacena un identificador único de nivel.
numero:	Campo de tipo Integer (), almacena un número el cual corresponde al valor del nivel.
descripcion:	Campo de tipo Text (), almacena un comentario correspondiente al nivel que se refiere.
sub_nivel:	Campo de tipo Varchar (), almacena un identificador del subnivel correspondiente al nivel al que se refiere.

Tabla 5.5.n. Tabla Estudiante.nivel

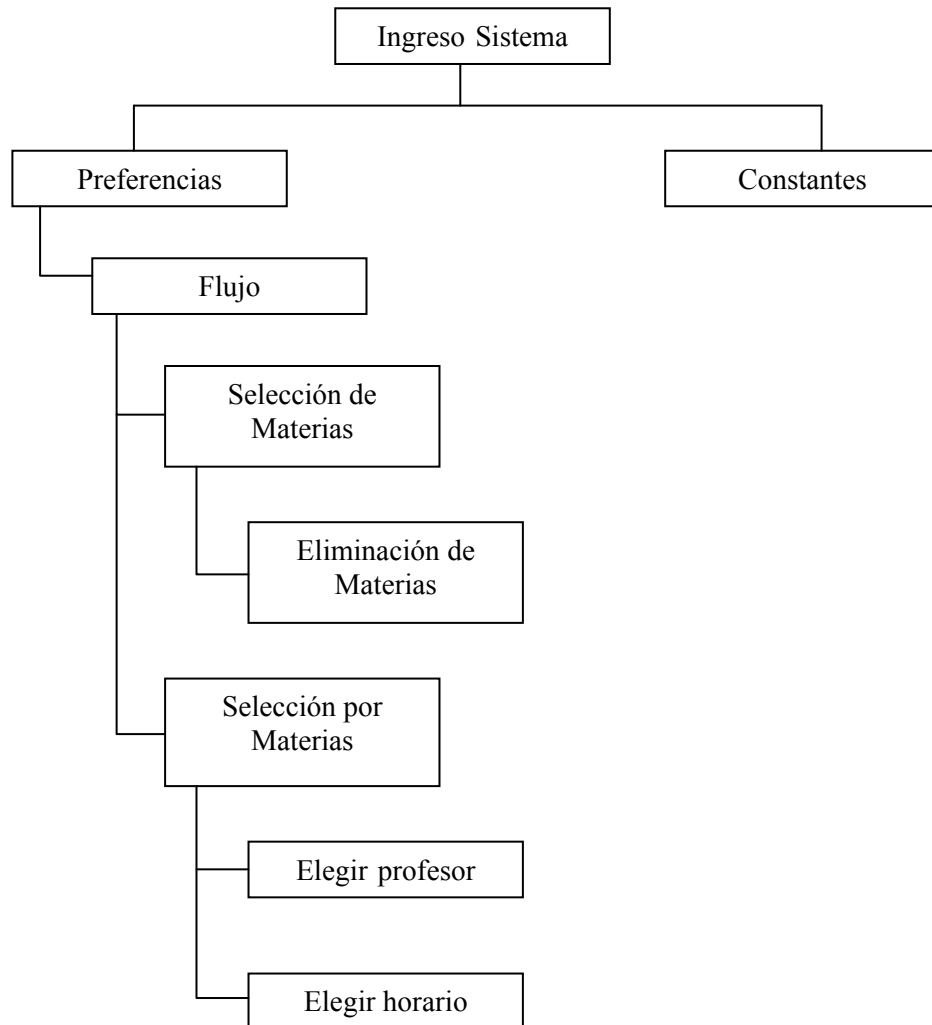
5.6. Diseño de interfaz

En esta sección se definirán los flujos de ventanas con las opciones disponibles para los usuarios del sistema dependiendo del tipo de usuario al que pertenezcan además del diseño de la interacción con el usuario.

5.7. Flujo de ventanas y layouts

El flujo de ventanas describe la secuencia de las ventanas que el usuario observará a la vez que indican la funcionalidad requerida para el sistema. Permite ayudar a establecer la interfaz de usuario y la describe en términos de las tareas que se esperan que sean realizadas

por éstos, sin necesidad de detallar todos los casos excepcionales, como el manejo de errores.



La Figura 5.5. Flujo de ventanas para un usuario estudiante

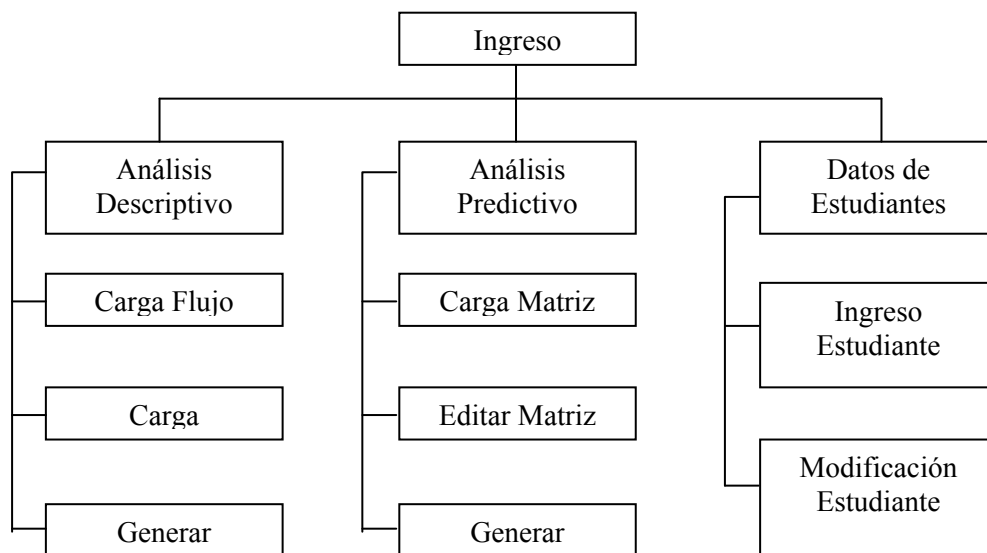


Figura 5.6. Flujo de ventanas para un administrador del sistema.

Los layouts de cada ventana serán presentados dentro del Anexo 2, que se refiere al manual del usuario.

El layout define la apariencia real de una ventana, describiendo sus partes visuales reales, como vistas, menús, botones, textos, campos, iconos, etc., y su ubicación dentro de una pantalla. Los layouts definen la conexión y secuencia de las ventanas.

5.8. Diseño de páginas web

Presentamos el diseño de las páginas del sitio Web desarrollado para la demostración de este proyecto de tesis, el cual sigue las indicaciones previstas en el análisis de la interacción.

En primer lugar, todas las páginas deberán guardar consistencia en cuanto a la disposición de los elementos de la pantalla. Así, en la parte superior estará fijo el logo del sitio, mientras que en el lado izquierdo se ubicará el menú de opciones. El contenido de este menú dependerá del usuario que esté navegando el sitio, bajo esto estará un menú con las opciones del usuario. El contenido de este menú dependerá del usuario que este navegando en el sitio. El usuario verá su identificación con nombre en cada página.

Adicionalmente, en la página de ingreso se dará una opción de inicio de sesión del sistema donde podrán logonearse tanto los estudiantes como los administradores. El sector principal de la página tendrá un contenido variable dependiendo de la opción elegida por el usuario.



Figura 5.7. Áreas en las que se dividen las pantallas del sitio

Como se puede apreciar en la **Figura 5.8** en el área de trabajo aparecen las ofertas más recientes, las cuales incluyen una imagen del inmueble acompañada de una breve descripción y el valor de su arriendo, que son las características que les interesan a las personas que navegan por el sitio.

Así mismo, para que los usuarios desarrollen un buen modelo mental se ha hecho uso de colores que contrasten para poder diferenciar las

áreas en las que se dividen las pantallas del sitio, así como colores diferenciados para el menú, y los títulos de cada área, todo esto con el fin de asegurar la visibilidad del sistema.

Con la presencia de los menús en todas las páginas se asegura la navegabilidad del sitio, ya que permiten pasar de una opción a otra en un solo paso. Los nombres de las opciones deben cumplir con el principio de permisividad explicado en el análisis de la interacción Hombre - Máquina.

Cuando un usuario registrado en el sistema haya ingresado se le presentará la pantalla que se muestra en la Figura 5.9. En esta página el título del área de trabajo indica que el usuario se encuentra en el módulo de administración del sistema y se muestran el nombre del usuario y su rol. En el área de menús aparecen las opciones a las que el usuario tiene acceso dependiendo del grupo al que pertenece, y se presenta el enlace para cerrar la sesión.

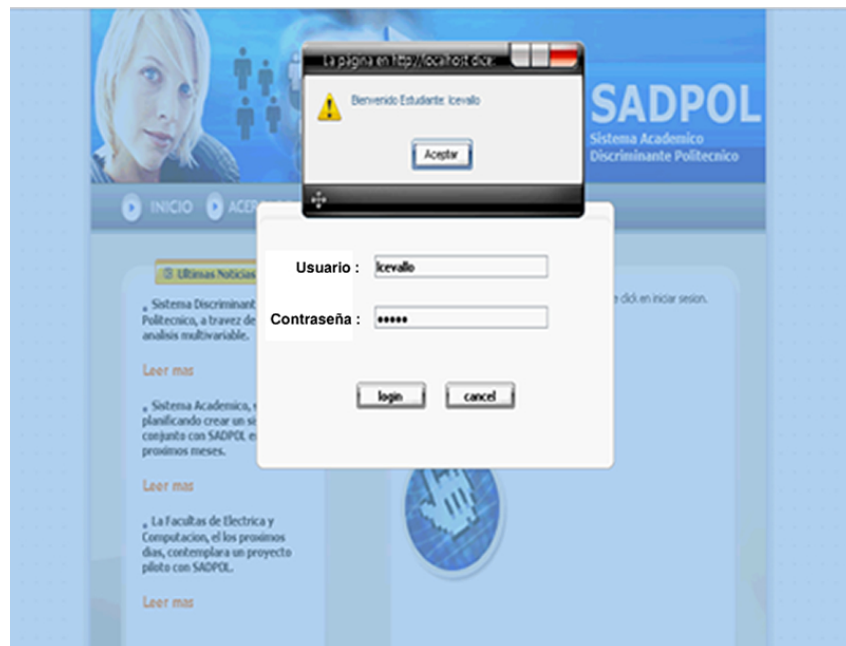


Figura 5.8. Pantalla de bienvenida al usuario que ingresa al sistema

Quando el usuario hace uso de una de las opciones del menú, se le presentará un formulario que le permitirá interactuar con el sistema. Los nombres de los campos de los formularios proporcionarán una terminología que cumplan con el principio de la familiaridad, para que los usuarios reconozcan fácilmente el uso de cada uno de estos campos.

Los formularios serán ubicados en el área principal y en la parte superior, el título deberá indicar la transacción que se va a realizar. Luego se mostrarán todos los campos que pertenecen al formulario, y

finalmente el botón que permitirá ejecutar la acción requerida. Este tipo de diseño será utilizado para todos los formularios del sistema.



The screenshot shows the 'Contactanos' (Contact Us) page of the SADPOL system. The header features a blue banner with a woman's face on the left, a network diagram in the center, and the SADPOL logo on the right. Below the banner is a navigation menu with buttons for 'INICIO', 'ACERCA DE', 'PLANIFICACION', and 'CONTACTANOS', along with a 'Salir' button. The main content area has a yellow 'Contactanos' tab. On the left, there is a circular icon of a book and a list of contact information: 'Profesor encargado: Fabricio Echeverria, fechever@espol.edu.ec' and 'Colaboradores: Luis Cevallos, lcevallo@fiiec.espol.edu.ec, Ivan Gonzales, lgonzales@fiiec.espol.edu.ec, Marlon Altamirano, maltamir@fiiec.espol.edu.ec'. The main form area contains a heading 'Tus Sugerencias' and a sub-heading 'Para poder ayudarnos al crecimiento de este sistema, podra enviarnos sus sugerencias por este medio.' Below this are input fields for 'Nombre:' and 'Correo:', a large text area for suggestions, and 'Enviar' and 'Limpiar' buttons.

Figura 5.9. Pantalla de formulario

Cuando la transacción ha sido realizada, el resultado de la misma aparecerá con un mensaje de realización o error. Este mensaje deberá ser lo suficientemente entendible para permitirle al usuario obtener una buena retroalimentación y reconocimiento de errores, dado el caso, de la acción ejecutada.

Por último, Si un usuario estudiante va a realizar su preferencia en materia de semestre se le presentara una pantalla con el flujo cargado, con el cual podrá arrastrar las materias a la que desea aplicar su preferencia como se muestra en la figura 5.11

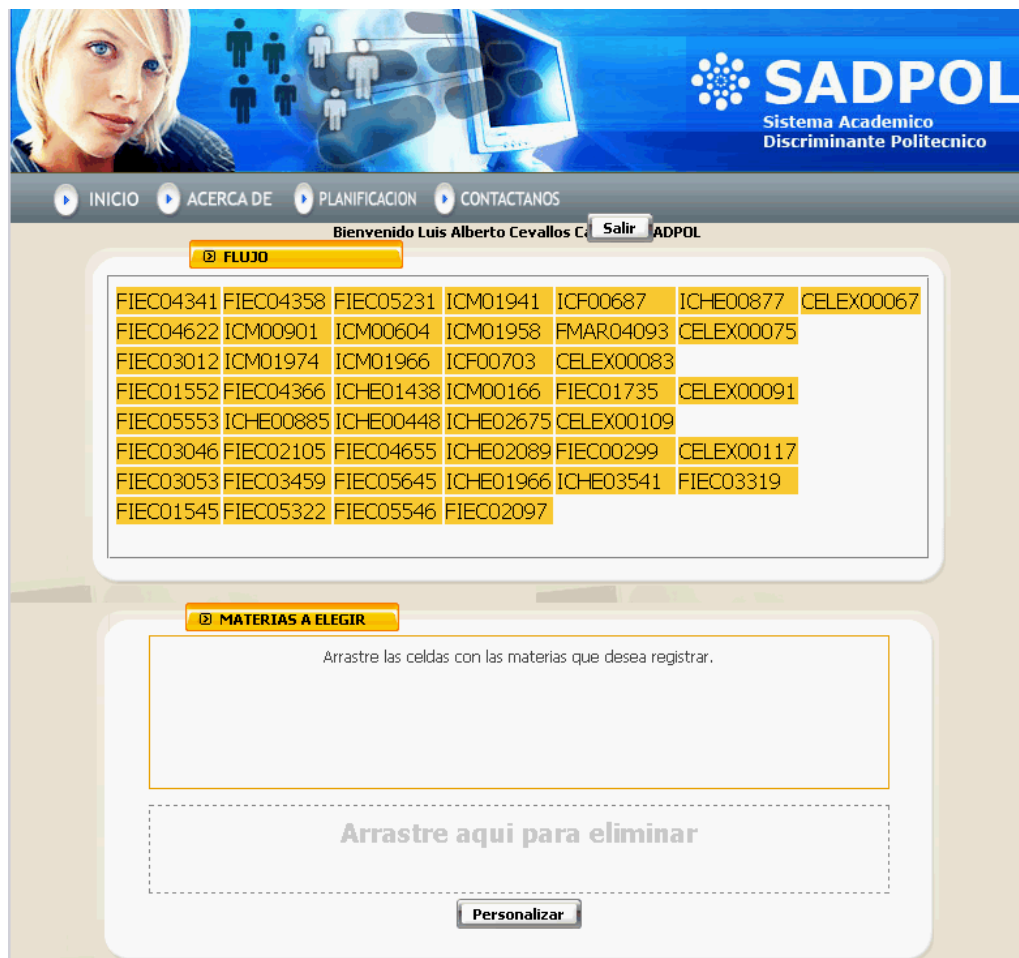


Figura 5.10. Flujo de materias.

Esto es lo que se llama un efecto Drag and drop (arrastrar y soltar). El cual brindará la facilidad de usuario para realizar la selección de las materias.

5.9. IMPLEMENTACIÓN Y PRUEBAS

5.9.1. Procesos de implementación

En esta sección se hace una descripción del proceso de implementación de la aplicación demostrativa de este proyecto de tesis, una vez realizadas las etapas de análisis y diseño del sistema.

Se han definido ciertas reglas en la programación de la aplicación para facilitar el mantenimiento y entendimiento del sistema. Las reglas adoptadas son:

Los nombres de las clases empiezan con letra mayúscula y deben ser significativos.

Los nombres de los métodos empiezan con letra mayúscula por cada palabra que la compone, por ejemplo, CrearContrato.

Los nombres de las variables de clase y método empiezan con letra minúscula.

El código debe estar debidamente documentado, incluyendo descripciones para las clases y los métodos. Tal como se explicó en el análisis de tecnologías, el lenguaje de programación escogido para el desarrollo del

proyecto es PHP, que es un lenguaje especialmente enfocado en el desarrollo de aplicaciones Web con contenido dinámico.

El diseño de las páginas Web, tal como se definió en la sección de herramientas de desarrollo, fue hecho en Macromedia Dreamweaver, siguiendo los principios de diseño de la interfaz establecidos en el capítulo anterior.

Dentro del proceso de instalación y configuración de las herramientas de desarrollo, en primer lugar se procedió a la instalación del servidor Web Apache y del módulo de PHP (XAMPP). Se usa este Framework porque ya viene configurado el PHP para trabajar con el Apache Server, evitando así reconfigurar el mismo. Adicionalmente se instaló el IDE Easy Eclipse for Lamp y el Netbeans IDE early access for PHP para el desarrollo de la lógica de programación, y la base de datos PostgreSQL para el almacenamiento de los datos. También se agregó las librerías de Smarty y ADODB para el patrón de diseños MVC.

Luego de realizado el proceso de instalación y configuración de las herramientas de desarrollo, se procedió a la codificación del módulo de conexión a la base de datos desde una página PHP. Las pruebas de conexión se realizaron utilizando el Easy Eclipse for Lamp y programando la clase Database para que establezca la conexión a la base de datos, tomando como referencia el ADODB.

A continuación se programaron las páginas y las clases en PHP, haciendo pruebas de las transacciones de inserción, modificación, eliminación y consulta de registros desde y hacia la base de datos.

Luego del desarrollo de la lógica de programación se realizaron las pruebas para comprobar el correcto funcionamiento del sistema y eliminar los errores que no habían sido detectados.

5.10. Plan de pruebas

En lo que concierne al plan de pruebas realizamos pruebas de usabilidad, las mismas que nos permitió probar la facilidad con la cual los usuarios podían operar la aplicación. Siendo los objetivos principales los siguientes:

- Determinar si la interfaz del usuario es lo suficientemente intuitiva tanto para usuarios que tienen experiencia en aplicaciones web como para los que no la tienen.
- Determinar si los usuarios pueden usar nuestro sistema completando satisfactoriamente el proceso de logon y de selección de preferencias.
- Determinar si la aplicación requiere modificaciones para poder cumplir con los objetivos anteriores.

Los usuarios de pruebas fueron estudiantes ESPOL siendo expertos y no expertos en operar sistemas de aplicaciones web. Dichas pruebas fueron realizadas en el lapso de 3 días, se seleccionaron cerca de 10 estudiantes,

un número mayor de estudiantes con poca experiencia en operar dichas aplicaciones para así tener una retroalimentación positiva de la usabilidad de nuestro sistema.

Luego de obtener los resultados de dichas pruebas, es posible derivar conclusiones importantes, orientadas al mejoramiento de la aplicación para que al ser operadas se adapte más a las costumbres de los usuarios reales. De todas las sugerencias sobre los cambios para hacer de la interfaz más intuitiva o amigable podemos destacar las siguientes:

- Modificar la forma de presentar el respectivo flujo de materias del estudiante.
- Cambiar los colores que se usaron en la interfaz de tal manera no cause un impacto visual negativo hacia el usuario.
- Probar todos los caminos de las estructuras (bucles) de control para asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez.
- Realizar pruebas de ingreso al sistema para verificar que sólo se permitían usuarios ingresados a la base.
- Si la interacción de formularios fue clara para el usuario.
- Si el usuario consultó el manual del usuario para poder operar la aplicación.

Adicionalmente las pruebas se encaminan a encontrar errores en inicializaciones de datos erróneas, inconsistencias en los tipos de datos, nombres de variables erróneos, comparaciones incorrectas y finalización inesperada de bucles.

Una vez realizadas las pruebas de unidad se efectúan las pruebas de integración, las cuales permiten integrar los módulos probados en unidad y probar todo el sistema en conjunto con el objetivo de detectar errores asociados con la interacción entre los módulos del programa.

En este plan de pruebas se ha planeado utilizar la técnica de la integración descendente, en donde se va probando desde el módulo de ingreso al sistema, donde se crea la sesión de usuario, y luego se sigue la secuencia incorporando los módulos de ingresos, modificaciones, eliminaciones, pagos, consultas, permisos y manejo de formularios.

Después de que las pruebas de integración estuviesen finalizadas se continúa con las pruebas de validación, en las cuales se comprueba que el sistema cumple con los requerimientos funcionales, que se logran los requisitos de seguridad, confiabilidad, facilidad de mantenimiento y que la documentación del código sea correcta y entendible. Es importante destacar que este tipo de pruebas habitualmente deben ser realizadas por el usuario

final, pero en este caso se ha obviado la existencia de los mismos, por lo que las pruebas de validación han sido realizadas por nosotros mismos.

En los casos en que se encontraron errores en el sistema como producto de la ejecución de un recorrido dentro del plan de pruebas, se debe llevar a cabo un proceso de depuración. La depuración del código comienza luego de que un caso de prueba encuentra un error, se evalúan los resultados obtenidos con los esperados, se corrige el error y se vuelve a probar.

5.11. Resultados de las pruebas

Durante el proceso de pruebas de la aplicación se encontraron errores relacionados con inicializaciones de datos erróneas o inexistentes, nombres de variables mal escritos o inexistentes, comparaciones incorrectas y finalización inesperada de bucles, las mismas que fueron solucionadas con una estructura fija de identificación de variables.

Con lo que respecta a la presentación del flujo de materia al estudiante, se modificó la interacción para que simplemente el usuario pueda manipular el gráfico de una manera sencilla, ya sea arrastrando lo seleccionado hacia la parte de interés.

Con respecto a las pruebas de integración, se hizo un chequeo exhaustivo del sistema haciendo un recorrido de cada uno de los escenarios posibles. Estas pruebas finalizaron una vez que ya no se detectaron errores en el sistema ya integrado.

Las pruebas estadísticas del análisis discriminante fueron realizadas en el software Minitab el cual proporciona la función del análisis discriminante con el cual podemos comprobar que los valores obtenidos en nuestro proyecto de tesis son reales y cumplen con el análisis estadístico.

Finalmente, se comprobó que el sistema cumpla con los requerimientos bajo los cuales fue desarrollado, cubriendo de esta manera las pruebas de validación del sistema.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES.

Como resultado de haber desarrollado este proyecto de tesis tenemos un sistema administrador de predicciones; que combinado con un esquema para establecer preferencias de estudiantes se ha logrado satisfacer las necesidades de los mismos.

1.- Nos hemos enfocado en la optimización del uso de la información para establecer perfiles de los estudiantes.

2.- Se implementó una solución para asegurar la independencia de plataforma, esto se lo logró con herramientas que aseguraron la postabilidad y estabilidad del sistema.

3.- El código de la aplicación ha sido diseñado y estructurado de tal manera que facilita la implementación de nuevos módulos al sistema de forma rápida y sencilla.

4.- La interfaz del sistema ha sido diseñada con el fin de brindar al usuario facilidad en la navegación del sitio y en el entendimiento de las tareas que puede realizar.

5.- La base de datos fue desarrollada de tal forma que se logró mantener la consistencia de datos y la integridad relacional de las tablas.

6.- El dividir el proyecto en dos modelos de datos, uno de estudiantes y otro de administrador, permite separar los aspectos relacionados con cualquier usuario. Todo lo relacionado con la presentación de contenidos son aplicados independientemente del usuario.

7.- Los niveles establecidos en la administración de contenidos, aseguraron una división apropiada de las tareas para cada usuario y a la vez proporcionaron la seguridad necesaria para evitar la manipulación de información restringida por parte de usuarios no autorizados.

RECOMENDACIONES.

1.- Es recomendable generar un módulo que recorra todos los registros que se encuentren inactivos e inutilizados durante un determinado período, para eliminarlos físicamente de la base de datos, puesto que en este proyecto se utilizó solo el concepto de eliminación lógica de registros.

2.- Para versiones futuras, se recomienda el uso de procedimientos estándares de seguridad en la transferencia de los registros que están siendo sincronizados, con el fin de que no se transmitan como texto plano, debido a que esto podría ser vulnerable a ataques externos.

ANEXOS

ANEXO 1

GUIA DE USO DEL SISTEMA

Aquí se detallan las guías para el uso del sistema discriminante politécnico. Para poder ingresar a la página principal del sitio se debe utilizar un navegador de Internet y acceder a la dirección:

http://nombre del dominio Servidor SADPOL



Figura A. Pantalla principal del sistema.

Esta pantalla presenta las opciones básicas para un invitado es decir sin haber ingresado un nombre de usuario y una contraseña, El menú para los

navegadores del sitio que no hayan accedido al sistema tienen las opciones de “Inicio”, “ACERCA DE” y “Contáctenos”.

En el menú “INICIO” se tienen las opciones de ingreso al sistema a través del sistema login el cual presentará una ventana modal donde el usuario ingresará su usuario y contraseña para poder acceder dependiendo del menú de estudiante y al menú de administrador.

Opción Acerca de

En esta ventana se muestra información sobre el objetivo que se desea alcanzar al desarrollar el sistema SADPOL⁽²⁾ así como la información sobre los desarrolladores del sistema

Opción Contáctenos:

En esta ventana se muestra un pequeño formulario que permitirá al invitado enviar alguna sugerencia o información que haya encontrado en el sistema. Esta sugerencia será enviada vía e-mail al administrador del sistema y a los desarrolladores del mismo

Opción Acceso autorizado al sistema

Para poder ingresar al sistema existen dos roles el rol estudiante y el rol administrador, cada uno de estos actores tienen diferentes opciones dentro

del sistema. El usuario y contraseña para el rol de estudiante será el mismo que los estudiantes poseen para el sistema de correo de la ESPOL, en cambio para el usuario administrador el usuario y contraseña será creado por los desarrolladores del sistema con la preferencia que desee el administrador.



Figura A1. Acceso autorizado

Si el nombre de usuario o la contraseña son incorrectos, se presentará un mensaje de error (Figura A2), sino se dará paso a la página de bienvenida de los usuarios registrados (Figura A3), la cual indica el nombre del usuario y describe al grupo al cual pertenece, además presenta el menú de opciones según su grupo de usuario.



Figura A2. Mensaje de error.

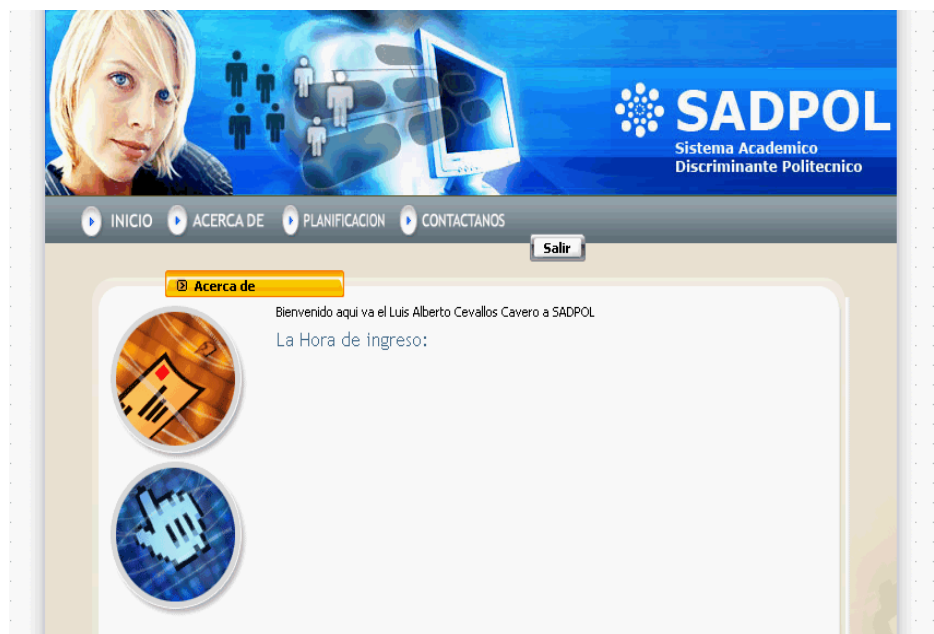


Figura A3. Muestra nombre de usuario.

ANEXO 2

MANUAL DEL USUARIO 1

ADMINISTRADOR

MANUAL DEL ADMINISTRADOR

En este apéndice se detallará la guía para la administración, instalación y configuración del sistema.

Opciones de administradores del sistema

Los usuarios del grupo de administradores tienen las opciones de:

Efectuar un análisis discriminante

Efectuar un análisis predictivo

Consultar las transacciones efectuadas

Ingresar y modificar estudiantes

Menú de análisis discriminante.-

Esta página le permitirá al administrador seleccionar el día la materia dependiendo de su carrera, los datos que han sido conector a través del modulo de estudiantes de las preferencias por grupos.

Los grupos serán:

Grupo de la mañana

Grupo de la tarde

Grupo de la noche

Al haber elegido el día, la unidad y la carrera correspondiente con el botón de calcular comenzará la minería de datos con respecto a la preferencias elegidas en el modulo del estudiante.

El botón calcular comenzará a llenar 3 grillas de los tres grupos mencionados anteriormente; se procederá a extraer de la base de datos el factor P, el nivel y la edad.

Al lado de cada grilla existirá un botón para obtener la función lineal que es nuestra función discriminante; conociendo de esta manera los valores a los cuales denominaremos centroides.

Con estos centroides podremos llegar a determinar a que grupo pertenece un nuevo estudiante.

Análisis predictivo.-

Para realizar esta opción en nuestra ventana disponemos de 4 secciones importantes las cuales vamos a mencionar:

- Una tabla donde se encontrarán los nuevos individuos ingresados (a los cuales se le efectuará el análisis predictivo para averiguar a que grupo pertenecen), se dispondrá de cada uno de estos individuos las variables con las cuales estamos trabajando (Factor P, edad, nivel).
- Una tabla donde constará el último análisis discriminante realizado por el usuario administrador. Esta tabla constará de 3 filas y 3 columnas en las cuales se mostrará el valor de la función discriminante para cada grupo.
- Una tabla en la cual se mostrará el centroide calculado para cada grupo del último análisis discriminante realizado por el administrador.
- La sección de botones donde podremos: calcular el predictivo para el grupo para el cual estamos trabajando y asignar a que grupo pertenece cada individuo, el siguiente botón resetear limpiará la tabla para individuos nuevos.

Se mostrará una ventana modal donde el usuario administrador podrá observar las respuestas de su análisis predictivo realizado (no podrá realizar cambios a menos que efectué otro análisis).

Log de Transacciones

Esta opción mostrará todos los análisis realizados por el administrador durante su sesión. Se mostrará los siguientes campos en una tabla (Estos valores no podrán modificarse, serán solo de consulta):

- Nombre de la Materia
- Día del horario
- Código de la Materia
- Facultad
- Grupo
- Factor 1
- Factor 2
- Factor 3
- Centroide
- Acción
- Hora y día de Ejecución



The screenshot displays the SADPOL (Sistema Académico Discriminante Politécnico) interface. At the top, there is a header with a woman's face on the left and the SADPOL logo on the right. Below the header is a navigation menu with buttons for INICIO, ANALISIS DISCRIMINANTE, ANALISIS PREDICTIVO, ESTUDIANTE, and REGISTRO. A 'Salir' button is also present. The main content area is titled 'Historial de Transacciones' and contains a table with the following columns: Día, Facultad, Nombre Materia, Código Materia, Factor 1, and Factor 2. The table is currently empty, and a scrollbar is visible at the bottom.

Día	Facultad	Nombre Materia	Código Materia	Factor 1	Factor 2

Figura A2.1. Log de transacciones

Ingresar y modificar estudiantes

El administrador podrá ingresar un nuevo estudiante, así como buscar uno ya existente así y modificar de la información que dispone cargada previamente.



The screenshot displays the SADPOL (Sistema Académico Discriminante Politécnico) web application interface. At the top, there is a header with a blue background featuring a woman's face and icons of people. The header includes the SADPOL logo and the text "Sistema Académico Discriminante Politécnico". Below the header is a navigation menu with the following options: INICIO, ANALISIS DISCRIMINANTE, ANALISIS PREDICTIVO, ESTUDIANTE, and REGISTRO. A "Salir" button is located below the navigation menu.

The main content area is divided into two sections:

- Datos del Estudiante:** This section contains a form with the following fields:
 - Matrícula:
 - Nombre:
 - Apellido:
 - Correo Electrónico:
 - Unidad:
 - Carrera:
 - Especialización:
 - Nivel:
 - Dirección:
 - Foto:
 - Teléfono:
 - Factor P:
 - Fecha de Nacimiento:
 - Promedio:
- Datos del Usuario:** This section contains a form with the following fields:
 - Usuario:
 - Contraseña:

At the bottom of the "Datos del Usuario" section, there are two buttons: "Guardar" and "Limpiar".

Figura A2.2. Ingresar o modificar estudiantes

ANEXO 3
MANUAL DEL USUARIO 2
ESTUDIANTE

Opción del usuario estudiante

Los usuarios del grupo de estudiantes tienen las opciones de:

Ingresar preferencia.

Modificar preferencia.

Consultar preferencia.

Eliminar preferencia.

El menú para el usuario estudiante es el siguiente

Inicio

Planificación

Acerca de

Contacto

En la página de inicio del estudiante se mostrará el nombre del estudiante ingresado y la hora del ingreso al sistema.

Ingreso de una planificación.- Al estudiante logoneado en el sistema se le mostrará el flujo correspondiente a la carrera en la que se encuentra registrado. En este flujo se le permitirá arrastrar y soltar, cierta área las materias que desea tomar el siguiente semestre.

En esta área y más debajo de esta sección se encuentra un botón de preferencia al darle clic a este botón se presentará a través de un efecto greybox páginas dinámicas de un horario por cada materia seleccionada así como también los profesores que dictan la materia que ha seleccionado.

En el horario se presentará la opción de elegir en cada celda las horas clases que desea asistir, con las debidas restricciones que posee cada materia según el pensum es decir las horas que se deberían dictar.

Abajo del horario hay un botón para guardar el horario preferido para esa materia a la base de datos.



Figura A3.1. Ingreso o selección de un registro para modificar sus datos

Modificación preferencia.- El usuario tendrá la opción de modificar el horario seleccionado con anterioridad solamente con volver a ingresar la opción de materia seleccionada, y realizando un nuevo registro.

Eliminación preferencia.- Un estudiante tiene permitido la eliminación de registros de materias seleccionadas.

Opción Acerca de

En esta ventana se muestra información sobre el objetivo que se desea alcanzar al desarrollar el sistema SADPOL así como la información sobre los desarrolladores del sistema

Opción Contáctenos:

En esta ventana se muestra un pequeño formulario que permitirá al invitado enviar alguna sugerencia o información que haya encontrado en el sistema. Esta sugerencia será enviada vía email al administrador del sistema y a los desarrolladores del mismo

DÍA\HORA	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
07:00-07:30		ICM01966		ICM01966		
07:30-08:00	ICM01966	ICM01966		ICM01966	ICM00604	
08:00-08:30	ICM01966		ICM01966	ICM00604	ICM00604	
08:30-09:00				ICM01966	ICM00604	
09:00-09:30		ICM00604			ICM00604	
09:30-10:00		ICM00604			ICM00604	
10:00-10:30						
10:30-11:00						

Figura A3.2. Modificación y Eliminación

BIBLIOGRAFÍA

- [1] Piatetski-Shapiro et al., 1991; Chen et al., 1996; Mannila, 1997.
- [2] Richard Creeth, Nigel Pendse, Winkipedia, Enciclopedia Libre, http://es.wikipedia.org/wiki/Data_mining.C-27-IN-6012-010 Documento básico de minería de datos.
- [3] T. Hastie, R. Tibshirani, J. Friedman "The elements of Statistical Learning: Data Mining, Inference, and Prediction".
- [4] Michalski, R. Bratko, I. Kubat, M eds.1998. Machine Learning and Data Mining, Methods and Applications, John Wiley & Sons Ltd, West Sussex, Inglaterra.
- [5] Reportes Técnicos en Ingeniería del Software. 7(1): 26-29, ISSN 1667-5002. © CAPIS-EPG-ITBA,(<http://www.itba.edu.ar/capis/rtis>) 28.
- [6] Luis R. Rivera Fernández, Inteligencia de negocio y bodegas de datos,
- [7] Hand, David; Mannila, Heikki; Smyth, Padric (2001). Principles of data mining. Cambridge: Massachussetts Institute of Technology.
- [8] <http://www.daedalus.es>, C-27-IN-6012-010 - Noviembre de 2002, DAEDALUS – Data, Decisions and Language, S.A.
- [9] Michalski, R.S., Bratko, I., Kubat M. 1998. Machine Learning and Data Mining.

- [10] Methods and Applications. Wiley & Sons Ltd., EE.UU

- [11] Jhon Wiley Alan Simon and Sons. Data Warehouse, Data Mining and OLAP. USA, 1997.

- [12] Stephen Haag et al.. Management Information Systems for the information age, págs. pp 28. ISBN 0-07-095569-7.

- [13] Xingquan Zhu, Ian Davidson (2007). Knowledge Discovery and Data Mining: Challenges and Realities, págs. pp 18. Hershey, New Your. ISBN 978-1-59904-252-7.

- [14] <http://users.dsic.upv.es/~flip/LibroMD/>, Información asociada al libro "Introducción a la Minería de Datos". (C) José Hernández Orallo, M.José Ramírez Quintana, César Ferri Ramírez. Editorial Pearson, 2004. ISBN: 84 205 4091 9.