



INSTITUTO TECNOLÓGICO DE APIZACO

DIRECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

**“APLICACIÓN DE ÁRBOLES DE DECISIÓN EN UN SISTEMA TUTOR PARA LA
CONFORMACIÓN DE HORARIOS BAJO DEMANDA DEL ESTUDIANTE EN EL
INSTITUTO TECNOLÓGICO SUPERIOR DE XALAPA”**

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA: RODRIGO RODRÍGUEZ FRANCO

ASESORES

M.C. BLANCA ESTELA PEDROZA MÉNDEZ

M.C. MARIA GUADALUPE MEDINA BARRERA

APIZACO, TLAXCALA 2013

Índice

Índice de Figuras.....	iv
Índice de Tablas.....	vi
Introducción.....	vii
Capítulo 1 Planteamiento del Problema	10
1.1 Introducción.....	10
1.2 Origen del Proyecto	10
1.3 Planteamiento del Problema	12
1.4 Pregunta de Investigación.....	14
1.5 Justificación.....	15
1.6 Objetivo General	16
1.7 Objetivos Específicos	16
1.8 Contenido y Alcance	17
1.9 Áreas que participan	18
1.10 Procedimiento de la Investigación.....	19
1.11 Limitaciones del Proyecto.....	20
1.12 Estado del Arte.....	20
Capítulo 2 Marco Teórico.....	23
2.1 Introducción.....	23
2.2 Ciclo de Vida de Desarrollo de Sistemas de Información.....	23
2.2.1 Ingeniería de software	23
2.3 Metodología Tradicional o Estructurada	28
2.4 Aprendizaje Automático.....	28
2.4.1 Esquema general de un sistema de Aprendizaje Automático.....	29
2.4.2 Clasificación del aprendizaje	30
2.4.3 Aprendizaje supervisado o clasificación.....	31
2.5 UML (Lenguaje Unificado de Modelado)	43
2.5.1 Diagramas de Casos de Uso.....	44
2.5.2 Diagramas de Clases	46
2.5.3 Bases de Datos	48

2.6	PHP HyperText Preprocessor	50
2.7	HTML HyperText Markup Language.....	51
2.7.1	Nociones Básicas de HTML	51
2.8	JavaScript	53
2.9	XML Lenguaje de Marcas	54
2.10	Hojas Estilo en cascada	54
2.11	AJAX Asynchronous JavaScript And XML	55
2.11.1	Cómo funciona AJAX.....	56
3	Análisis	58
3.1	Introducción.....	58
3.2	Encuesta	58
3.2.1	Estadísticos de la Encuesta	60
3.3	Módulos del Sistema	64
3.4	Lista de Procesos.....	65
3.5	Diagramas de Caso de Uso	67
3.5.1	Generación de carga	67
3.5.2	Entrada al Sistema	68
3.5.3	Generar Horario	68
3.5.4	Conexión BD	69
3.6	Diagramas de Estados	70
3.7	Requerimientos para el Usuario.....	73
3.7.1	Windows.....	73
3.7.2	Mac.....	73
3.7.3	Linux.....	74
3.8	Requerimientos para el desarrollo	75
3.8.1	Requerimientos de software.....	75
3.8.2	Requerimientos de hardware	75
4	Diseño	77
4.1	Introducción.....	77
4.2	Arquitectura del sistema	77
4.2.1	Modelo de diseño	78

4.2.2	Modelo de despliegue	79
4.3	Diagrama General.....	80
4.4	Diseño de la Base de Datos.....	81
4.4.1	Diagrama Entidad-Relación	81
4.4.2	Diccionario de datos	81
4.5	Diagramas de Flujo	84
4.5.1	Registro de Usuarios.....	85
4.5.2	Diagrama de flujo Creación de Carga	86
4.6	Diseño de la Interfaz	87
5	Programación	91
5.1	Introducción.....	91
5.2	Módulos del Sistema	91
5.2.1	Generación de Arreglo de Cargas Académicas.....	93
5.2.2	Selección de Carga Académica.....	96
5.2.3	Restricciones del Usuario.....	98
5.2.4	Aceptación de carga.....	100
5.3	Manual de Usuario	101
6	Pruebas	110
6.1	Introducción.....	110
6.2	Resultados de las pruebas.....	110
6.2.1	Inicio de sesión.....	111
6.2.2	Selección de materias	111
6.2.3	Selección de carga académica.....	112
6.2.4	Finalizar carga.....	112
6.3	Comprobación de la hipótesis	113
6.4	Conclusiones.....	113
	Glosario	ix
	Glosario.....	ix
	Bibliografía.....	xv

Índice de Figuras

Figura 2-1 Esquema General de sistema de aprendizaje	29
Figura 2-2 Modelo del sistema de clasificación	31
Figura 2-3 Comparación tecnología síncrona - asíncrona.....	57
Figura 3-1 Turno Preferido	60
Figura 3-2 Preferencia de horario.....	61
Figura 3-3 Factores a la hora de elegir horario	62
Figura 3-4 Actividades que influyen en la creación del horario.....	62
Figura No. 3-5 Aceptación del sistema de horarios	63
Figura 3-6 Generación de carga.....	67
Figura 3-7 Entrada al sistema	68
Figura 3-8 Generar Horario	69
Figura 3-9 Conexión BD	70
Figura 3-10 Inicio de sesión	71
Figura 3-11 Proceso de selección de materias.....	71
Figura 3-12 Proceso de selección e impresión de carga.....	72
Figura 3-13 Proceso de ayuda	72
Figura 3-14 Proceso de Acerca de	72
Figura 3-15 Proceso de Salir.....	72
Figura 4-1 Modelo de diseño.....	78
Figura 4-2 Modelo de despliegue	79
Figura 4-3 Diagrama General del Sistema	80
Figura 4-4 Diagrama Entidad-Relación.	81
Figura 4-5 Diagrama de flujo para el Registro de Usuarios.....	85
Figura 4-6 Diagrama de flujo de Creación de carga.....	86
Figura 4-7 Página principal del sistema (lado izquierdo).....	87
Figura 4-8 Página principal del sistema (lado derecho)	87
Figura 4-9 Materias a elegir.....	88
Figura 4-10 Desplegar horarios.....	89
Figura 4-11 Movilidad entre los registros, personalización e impresión	¡Error!

Marcador no definido.

Figura 4-12 Personalización materias y catedráticos	90
Figura 4-13 Vista de impresión.....	90
Figura 6-1 Inicio de Sesión.....	111
Figura 6-2 Selección de Materias.....	111
Figura 6-3 Selección de carga académica	112
Figura 6-4 Finalizar Carga.....	113

Índice de Tablas

Tabla 1-1 Aportaciones de las diferentes disciplinas al proyecto.....	18
Tabla 4-1 Estructura de la tabla Catalogo	82
Tabla 4-2 Estructura de la tabla Conocimiento	82
Tabla 4-3 Estructura de la tabla Dalumn	82
Tabla 4-4 Estructura de la tabla Horario	83

Resumen

La presente tesis, muestra el trabajo realizado dentro del Instituto Tecnológico Superior de Xalapa referente a la elaboración de un sistema de elaboración de horarios bajo demanda utilizando árboles de decisión, buscando con ello reducir los costos generados durante el proceso de inscripción, así como los tiempos que toman los alumnos en la elaboración de los mismos, del mismo modo se le proporciona al estudiante la conformación de las cargas académicas que propicien un avance constante y controlado, que ayuden al instituto a tener mejores índices de reprobación, deserción y eficiencia terminal. Para ello se creó un sistema desarrollado con PHP, Ajax y Java Script que se conecta a la base de datos del Sistema de Integración Escolar (SIE) que actualmente maneja la Institución para el control de los avances académicos de los estudiantes.

Introducción

Anteriormente la forma de llevar a cabo el proceso de reinscripciones en el Instituto Tecnológico Superior de Xalapa resultaba lento para todas las personas que intervienen en este proceso y costoso para la Institución, ya que se tiene que llegar a una hora en específico y esperar a aquellos que pasaron con anterioridad a que el horario elegido no tenga fallos, esto suele tomar mucho tiempo aun cuando los administradores dedican de uno a dos días para cada una de las nueve carreras, aun así, año tras año van llegando más alumnos a formar parte del ITSX y este paso se hacía aún más lento.

Para atacar el problema se desarrolló el sistema tutor para la generación de horarios bajo demanda, el cual es capaz de ayudar a los alumnos en la elección de su horario y también ahorrarle tiempo a la escuela para que no dedique tantos días para la reinscripción.

En el capítulo 1 de este trabajo se abordarán los temas del por qué surgió la idea de este proyecto, los objetivos, tanto generales como específicos, la pregunta de investigación, las limitantes del proyecto, la metodología que siguió el proceso de la misma, como las herramientas a utilizar en el desarrollo del sistema.

En el capítulo 2 se dará una breve explicación de las herramientas utilizadas, así como las técnicas de Inteligencia Artificial que se tomaron en cuenta para la creación de este sistema.

El capítulo 3 está conformado por los datos recabados de una encuesta que se aplicó con anterioridad para saber cuáles son los factores que intervienen en la decisión del alumno para la creación de su horario. También se hablará acerca de los módulos, procesos y los diagramas del sistema. Del mismo modo se abordarán temas referentes al diseño de la Base de Datos, y se propondrá una

interfaz tentativa para el sistema.

En el capítulo 4 se muestra el diagrama general del sistema, el diseño de la base de datos y como está estructurada, los diagramas de flujo que presentan un panorama general de lo que será el sistema, así como el diseño de la interfaz.

En el último capítulo se trata todo lo relacionado con la programación del sistema dividido por módulos y el manual de usuario que muestra cómo se deberá usar el sistema.

Por último se presentan las pruebas realizadas para verificar el impacto y funcionamiento del sistema y las conclusiones a las que se llegaron en el desarrollo del mismo.

Capítulo 1 Planteamiento del Problema

1.1 Introducción

Este apartado presenta el origen que dio lugar a la implementación del proyecto “Sistema Inteligente de Creación de Carga Académica” (Sección 1.1) y las problemáticas que se están abordando con la implementación de dicho sistema, que básicamente son: desde el punto de vista institucional, lo referente a la calidad del servicio que la institución presta a sus alumnos, y, desde el punto de vista técnico, el costo computacional que se tiene en la generación de soluciones aproximadas de cargas académicas (Sección 1.2).

Se presenta la justificación del desarrollo de la aplicación en función de la problemática señalada (Sección 1.3) ; así como, los objetivos generales y particulares que se persiguen en la elaboración del proyecto basados en la generación de cargas equilibradas (Sección 1.5 y 1.6).

Para acotar el proyecto, en el apartado de contenido y alcance se describen los requisitos funcionales y no funcionales de la aplicación (Sección 1.7), mencionándose las diferentes disciplinas que se utilizaron en el desarrollo de la misma, y el modelo de ciclo de vida en cascada, a través de la cual se desarrolló la ingeniería del software (Sección 1.9), y las limitantes que no cubre la aplicación (Sección 1.10).

1.2 Origen del Proyecto

El proyecto surge como producto de la implantación del Sistema de Gestión de Calidad y Mejora Continua que tiene el Instituto Tecnológico Superior de Xalapa (ITSX), con el objetivo de otorgar a sus alumnos un servicio de calidad en todas sus áreas y procesos, del mismo modo con el fin de coadyuvar a reducir los índices de reprobación de los estudiantes, que estaban muy por arriba de la media nacional y estatal. Es por ello que se han realizado diferentes encuestas de

servicios, practicadas tanto por la Jefatura de Carrera de Sistemas Computacionales, como por el Comité de Calidad, en busca de identificar los requerimientos del alumno y detectar áreas de oportunidad que sean susceptibles de mejora. Es así, que como resultado de la aplicación de encuestas de servicio, se pudo observar que el proceso de inscripción es de gran relevancia para el alumno.

Es importante señalar que actualmente el ITSX utiliza el Sistema de Integración Escolar (SIE) en el proceso de inscripción de los alumnos, el cual no contempla en su estructura, un módulo que permita al alumno la generación y selección de horarios tomando en cuenta sus requerimientos y avance académico, ya que el sistema únicamente se limita a procesar la carga académica seleccionada por el alumno de manera no automatizada. Es por ello, que antes del establecimiento de horario dentro del SIE, el alumno adquiere en forma impresa la oferta educativa que pone a su disposición el ITSX con el objetivo de seleccionar las materias, grupo y maestros para la configuración de su horario, proceso que realiza de manera manual, la mayoría de las veces sin tomar en cuenta su desempeño académico, lo cual limita sus posibilidades de balancear una carga académica que le permita un avance en el programa de estudios sin riesgo. Del mismo modo durante el proceso de inscripción la Institución proporciona asesoría a los alumnos a través de sus maestros en el proceso de configuración de sus horarios, sin embargo al no contar los asesores con información del desempeño académico mostrado por el alumno durante su formación y de las materias que no fueron acreditadas en el semestre anterior, la asesoría muchas veces se limita sólo al llenado del formato, el cumplimiento del límite de créditos y que las materias no tengan el mismo horario.

Como resultado de la aplicación de las encuestas de servicio se ha observado que una de las quejas que se tiene, es el tiempo tan largo que el alumno invierte para lograr configurar y seleccionar su horario de acuerdo a sus requerimientos y su avance académico, del mismo modo se pudo constatar en las encuestas, que otro

de los principales problemas que genera pérdida de tiempo significativa para el alumno, es el hecho de que una vez que ha configurado su horario y toca el turno de capturarlo en el SIE, los grupos que ha seleccionado se encuentran cerrados en cupo, motivo por el cual tiene que volver a realizar de manera manual todo el proceso de configuración de su horario, teniendo otra pérdida significativa de tiempo.

Estas circunstancias se reflejan en la necesidad de proporcionar a los alumnos un conjunto de opciones finitas de cargas académicas, tomando en cuenta tanto sus necesidades de horarios (matutino, vespertino, mixto), y otras preferencias (por ejemplo maestros), así como restricciones de tipo académico (como lo son las materias reprobadas), lo anterior con el objetivo de balancear sus cargas académicas de manera más ágil y expedita.

1.3 Planteamiento del Problema

Como hemos visto, el interés por el proyecto surge a partir de dos problemáticas, la primera de ellas relacionada con la calidad del servicio que presta el ITSX a sus alumnos en el proceso de inscripción, concretamente en la selección de cargas académicas; y la segunda, que tiene que ver con el costo computacional de la diversidad de opciones generadas en la selección de horarios de los alumnos de acuerdo a las expectativas de cada uno de ellos; lo que da origen a la idea del desarrollo de un sistema automatizado, que ofrezca un conjunto finito de opciones de cargas académicas en tiempo relativamente corto, es así que una de las disciplinas que pueda dar soluciones a esta problemática es la inteligencia artificial de ahí la necesidad de aplicar sus técnicas en la solución del problema. Es por ello que se utilizó la técnica de Árboles de Decisión para generar opciones de cargas académicas, tomando en cuenta el historial académico del alumno, así como también las restricciones propias de la normatividad vigente de los Tecnológicos.

En relación a la primera problemática, es importante señalar que actualmente, el ITSX como cualquier organización prestadora de servicios, busca a través del

sistema de calidad que sus actividades sean coordinadas, dirigidas y controladas en relación a la calidad, con el objetivo de obtener y mantener la satisfacción de las necesidades de los alumnos, es por ello que el proceso de inscripción durante las diferentes revisiones para la obtención de la certificación de la escuela, ha sido identificado como un área de oportunidad y mejora que necesita ser agilizada y sistematizada no sólo para satisfacer las necesidades del cliente, sino para brindarle al alumno un conjunto de opciones de cargas académicas que coadyuven a la consecución de su formación.

Actualmente en la configuración y selección de horarios, no solamente el alumno invierten una cantidad de tiempo en el límite de lo razonable (en los casos más críticos), sino que muchas veces la conformación de las cargas académicas no propicia un avance constante y controlado, pues el sistema no conlleva a una selección racional de la carga, que mejore el uso de las opciones que le ofrece el instituto.

En relación a los tiempos promedios que se tienen en la selección de horarios, actualmente de la población estudiantil que se tiene, el 55 % los alumnos presentan problemas en su situación académica (alumnos irregulares), es por ello que en el proceso de elaboración y configuración de sus cargas dedican un tiempo estimado de los 40 a los 55 minutos para la elaboración de las mismas, aunado a lo anterior, muchas veces no pueden seleccionar las materias que les permiten tener un mejor avance, ya que los grupos se encuentran cerrados por el cupo que se programó, lo que origina problemas con la terminación de su carrera, que está estipulada normativamente para ser cursada en un máximo de 12 semestres.

Del mismo modo durante el proceso de inscripción se presentan otros problemas de orden institucional, que se enuncian a continuación:

- Oferta académica presentada en forma impresa lo que hace difícil su valoración tanto por parte de los alumnos como de los asesores reticulares,

lo que origina mayores tiempos promedios de configuración de los horarios.

- Tiempo excesivo requerido para la conformación del horario por alumnos, sobre todo en los grupos de inscripción con situación académica complicada por reprobación, ubicados en los turnos con menos disponibilidad de cupo en los grupos.
- Cuellos de botella en ciertos lapsos del proceso de inscripción, lo que ocasiona que los alumnos no tengan un correcto asesoramiento en la conformación de sus horarios.
- Selección de materias que no propician un avance académico sin riesgo, lo que repercute en un elevado índice de reprobación.
- Horarios no consecutivos que exigen la permanencia del alumno en la escuela por largos periodos de tiempo, lo que repercute en el incumplimiento de tareas y proyectos.

Por otro lado, la problemática de orden computacional, está relacionada con la diversidad de posibilidades generadas para la configuración del horario de un estudiante, ya que en ella intervienen aspectos que van desde la hora que tiene asignada para inscribirse, las materias que debe llevar de acuerdo a su avance académico, falta de cupo de algunos grupos y necesidades de horarios consecutivos (matutino , vespertino o mixto), por lo cual se requiere de una valoración de criterios la cual consumirá un costo computacional.

1.4 Pregunta de Investigación

El manejo de Arboles de Decisión en la implementación de un sistema para la creación automática de la carga horaria de alumnos del ITSX, considerando diversas restricciones, ¿ayudarán a disminuir el tiempo de generación de horarios en al menos un 60 %?

1.5 Justificación

El manejo automatizado de la oferta académica (grupos que incluyen horario y docente), permitirá valorar con base a reglas de corte heurístico (criterio de expertos), las materias ya acreditadas por el alumno, asignaturas que tiene en estado de repetición, exámenes especiales pendientes de aplicar, rendimiento académico) promedio obtenido en cursos anteriores, límite de semestres y tiempo que le queda para concluir la carrera, toda esta información almacenada y disponible en el SIE, para estimar de manera automática sin requerir esfuerzo humano, mayor número de posibilidades propiciando una selección mejor y racional del horario por parte del alumno.

Al automatizar la configuración y selección de horarios se permitirá agilizar el proceso de inscripción, ya que la aplicación reducirá el tiempo de respuesta lo que le permitirá al Instituto atender de una mejor manera y en el tiempo estipulado a los alumnos propiciando su avance académico.

En relación al tiempo promedio que los alumnos invertirán en la configuración de sus horarios, se verá reducido significativamente, debido a la automatización de generación de soluciones mediante la implementación de árboles de decisión en el proceso de construcción de horarios propuestos.

Con la automatización de la generación de horarios bajo demanda el ITSX podrá asesorar en un 100% de los alumnos en el proceso de configuración de los horarios, ya que se realizó de manera automática tomando en cuenta tanto los antecedentes académicos del alumno como sus expectativas personales.

La aplicación es justificable desde el punto de vista técnico, ya que el problema de la combinación de soluciones es exponencial. Este tipo de problemas han encontrado soluciones mejoradas por medio de la aplicación de técnicas de inteligencia artificial, por lo que la propuesta, tiene ventajas por el tipo de circunstancias de restricción que se presentan y por los resultados que se han

observado en los proyectos de software que incorporan éste tipo de técnicas, reducción en muchos casos, hasta en un 75 % del tiempo de respuesta.

Se puede reducir el tiempo promedio individual de la selección de horario en el proceso de inscripción del ITSX, por medio del desarrollo de un sistema automatizado que aplique técnicas de inteligencia artificial, específicamente árboles de decisión, dentro del proceso de selección de cargas académicas permitiendo generar un número finito de soluciones, bajo un conjunto de situaciones deseables proporcionada por el alumno, respetando las restricciones de corte operativo y académico dadas las condiciones del instituto y el modelo educativo.

1.6 Objetivo General

Desarrollar un sistema de software que automatice el proceso de configuración y selección de horarios por parte de los alumnos, proporcionándoles un número finito de soluciones considerando sus necesidades y expectativas, mejorando con ello el tiempo promedio individual invertido en esta tarea, aplicando para ello una técnica de inteligencia artificial.

1.7 Objetivos Específicos

- Mejorar el proceso de configuración y selección de horarios reduciendo el tiempo, así como también la exploración de toda la oferta académica que ofrece la Institución a los alumnos.
- Facilitar propuestas de cargas académicas equilibradas al alumno, que le permitan tener un avance académico sin riesgo, tomando en cuenta sus necesidades y expectativas.
- Contar con la evaluación de los resultados obtenidos de la aplicación de una técnica de inteligencia artificial en procesos de automatización.

1.8 Contenido y Alcance

Con el objeto de delimitar el proyecto de “Sistema Inteligente de Creación de Carga Académica Bajo Demanda”, a continuación se identifican los alcances y circunstancias que forman la frontera del mismo:

- El desarrollo del software incluido en este proyecto, considera el análisis, diseño, implementación, evaluación de la aplicación, así como la discusión de resultados del uso de árboles de decisión para la configuración de horarios bajo demanda, para lo cual se aplicarán técnicas y herramientas propias de la Ingeniería del Software en cada una de las etapas.
- Los alcances funcionales esperados del proyecto se pueden señalar en la siguiente narrativa: la aplicación deberá ser desarrollada con una interfaz accesible, a través de objetos que le permitan al alumno una interacción sencilla en el proceso de configuración de cargas académicas.
- La aplicación estará desarrollada con herramientas de programación y sistemas de base de datos libres, debido a las carencias económicas que existen en la escuela.
- Una vez hecho el proceso de selección de las materias el sistema revisará la oferta académica de la institución mostrándole en una primera etapa soluciones finitas de su probable horario únicamente en función del tiempo de permanencia en la escuela.
- Terminada la primera generación de probables cargas académicas, el alumno podrá restringir la configuración con base al establecimiento de sus expectativas, como son: el turno en el cual desea asistir a clases y preferencia por algunos maestros, sin descuidar el aspecto académico que le permita un avance controlado y sin riesgo.
- La arquitectura por cuestiones de seguridad, únicamente proporcionará a los alumnos diferentes vistas del SIE, por lo que la selección y configuración de sus cargas académicas serán almacenadas en el servidor donde se ubicará la aplicación.

1.9 Áreas que participan

Dentro de las diferentes áreas del conocimiento que se requieren para el desarrollo de la aplicación, tenemos las mostradas en la Tabla 1-1.

Tabla 1-1 Aportaciones de las diferentes disciplinas al proyecto.

NOMBRE	TIPO DE PARTICIPACIÓN	ETAPA DEL PROYECTO QUE PARTICIPA
Ingeniería del Software	Proporcionará la metodología por medio de la cual se desarrollará la aplicación, garantizando un producto de calidad y de acuerdo a las especificaciones del cliente, para ello se utilizarán las diversas técnicas que permitirán la consecución del objetivo general.	Análisis, Diseño y Pruebas
Base de Datos	Esta área proporcionará la estructura del recipiente de la información, el cual contendrá las cargas académicas. Para ello se utilizarán los modelos entidad- relación y relacional para su diseño, así mismo proporcionará el proceso de normalización para garantizar la integridad, consistencia y no redundancia de la información.	Análisis, Diseño y Programación
Sistemas Distribuidos	Aportando las base de diseño para la arquitectura de la aplicación, que será cliente-servidor, con el objeto de atender el mayor número de peticiones por parte de los alumnos.	Diseño y Pruebas
Inteligencia Artificial	Aportará la base teórica para la formulación del proyecto, con base a las técnicas de clasificación, y algunas formas de optimización, que serán utilizadas en la generación de combinaciones, controlada por heurísticas.	Diseño y Programación

1.10 Procedimiento de la Investigación

La aplicación se desarrolló a través del Modelo en Cascada, el cual consiste en cada una de las fases que se describen a continuación:

- Análisis de los distintos requerimientos y restricciones más comunes por parte los alumnos en el proceso de selección de horarios, así como del proceso de inscripción por parte del ITSX y de los elementos que intervienen en él.

Para determinar los requerimientos en esta etapa se hará uso de herramientas de recolección de información, como son las entrevistas, encuestas, descripción a través de casos de uso, diagramas de flujo y diagramas de estados.

Así mismo, se analiza la forma en que se construyó el árbol de decisión para generación de soluciones, y de las técnicas de optimización para la generación y recorrido del árbol.

Con relación a las herramientas sobre las cuales se desarrolló la aplicación, se analizan las ventajas de cada una ellas en un entorno de una arquitectura cliente servidor.

- **Diseño** de la aplicación con base a los condiciones y restricciones del alumno. En esta etapa se utilizaron diagramas de clases, así como la utilización de diagramas de entidad- relación para describir la estructura de la base de datos y la aplicación del proceso de normalización de la misma, para la optimización de la información. Se diseñó la base del conocimiento con el objetivo de optimizar las búsquedas de horarios con patrones similares.
- **Programación** del sistema, es la fase de programación donde se implementa el código fuente a través de las descripciones de las interfaces mediante las cuales el estudiante puede configurar y seleccionar su carga académica, así mismo se hace una descripción de los algoritmos de mayor

relevancia implementados en la generación de un número finito de soluciones aproximadas para el estudiante.

- **Pruebas, Implantación y Mantenimiento** de la aplicación. Estas etapas se llevan a cabo durante la implementación del proyecto y posteriormente, en la etapa de mantenimiento.

1.11 Limitaciones del Proyecto

La aplicación cuenta con algunas limitantes que se describen a continuación:

- Para el funcionamiento de la aplicación se requiere que el alumno ingrese previamente al sistema “Agente Reticular” (desarrollado simultáneamente) para registrar las materias que debe cursar y sean almacenadas en la base de datos para su uso en el “Sistema Inteligente de Creación de Carga Académica Bajo Demanda”. La aplicación no permite realizar una selección de materias que no estén contenidas en las previamente seleccionadas en el sistema “Agente Reticular”.
- La aplicación no balancea el número de alumnos por grupo durante el proceso de inscripción, sin embargo existe otro proyecto en desarrollo que permitirá determinar la oferta académica, así como la cantidad de alumnos por grupo.

El presente proyecto no es un proceso de maximización o minimización, de hecho sólo es un proceso de generación de aproximaciones de cargas académicas bajo demanda.

1.12 Estado del Arte

Aplicaciones con árboles de decisión

En el trabajo de (Franco-Arcega 2013) desarrollaron varios algoritmos de generación de árboles de decisión a partir de grandes conjuntos de datos, los cuales resuelven algunas de las limitaciones de los algoritmos más recientes del

estado del arte. Tres de estos algoritmos permiten procesar conjuntos de datos descritos exclusivamente por atributos numéricos; y otro puede procesar conjuntos de datos mezclados. Los algoritmos propuestos procesan todos los objetos del conjunto de entrenamiento sin necesidad de almacenarlo completo en memoria. Además, los algoritmos desarrollados son más rápidos que los algoritmos más recientes para la generación de árboles de decisión para grandes conjuntos de datos, obteniendo resultados de clasificación competitivos.

FET es un programa de confección de horarios basado en Arboles de Decisión, el sistema crea un posible horario, este horario se somete a muchos cambios con los que se van seleccionando los mejores horarios, para obtener finalmente un horario razonablemente bueno, éste sistema de generación de horarios está enfocado únicamente a la parte de generación del horario sin tomar en cuenta las necesidades de los alumnos sino que únicamente se adapta a las necesidades de la escuela en cuanto a las aulas, maestros y las materias que están relacionadas a cada uno de ellos. (Sabrina Sestito 2001)

Otro de los proyectos que se tienen es el de Takekas, en el cual se abordó el problema de asignación de carga académica en una institución de educación superior, en él se observó la complejidad del problema debido a la cantidad de restricciones que se presentan y el criterio con el que se aplican. Este trabajo presenta un modelo computacional capaz de encontrar la asignación óptima de clases, maestros y horarios utilizando un algoritmo genético. Se realizaron pruebas al modelo, tomando como base los requerimientos en un período determinado de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Nuevo Laredo y una lista de los maestros disponibles y su horario, encontrando mejores resultados que los obtenidos en forma manual. (Elías-Gutiérrez 2012)

Es importante destacar que se tienen muchos proyectos en cuanto a la generación de horarios, otros de los algoritmos que se utilizan mucho en ésta implementación es la búsqueda tabú de que se define la búsqueda tabú como una metaheurística ya que incluye en sus propias reglas algunas técnicas heurísticas, su papel es dirigir y orientar la búsqueda de otro procedimiento más local de búsqueda. Entre

las ventajas de esta estrategia podemos mencionar que se puede implementar como una técnica determinística, no aleatoria, según lo mencionado por. (Glover 1993). Su versatilidad le hace una técnica muy utilizada para resolver el problema. Como ejemplo se tiene el caso de estudio planteado por Kendall [03] donde se compara el resultado de un sistema automatizado contra el modelo manual habitual, (Ardemil 2001) lo utiliza para optimizar calendarios deportivos, una variante del problema de la generación de horarios. (Di Gaspero 2003) en su tesis doctoral propone el framework llamado EasyLocal++, el cual implementa los procedimientos básicos de búsqueda tabú para ser utilizados en el problema de la generación de horarios. (Opents 2005) es una biblioteca que implementa el algoritmo de búsqueda tabú básico para poder ser utilizado en la solución de cualquier problema de optimización. Es notable su aplicación en el problema de generación de horarios, y una de las razones es debido a que se trata de una técnica reciente en comparación a las demás, y no se ha explorado por completo.

Como podemos observar existen infinidad de proyectos encaminados a la generación de horarios, sin embargo son encaminados sólo a las necesidades de la institución en relación a sus aulas, maestros y disponibilidad de horario, en el caso de la aplicación del presente proyecto es que se desarrolló en base a las necesidades y expectativas del alumno.

Capítulo 2 Marco Teórico

2.1 Introducción

Este apartado presenta el modelo de desarrollo de sistemas (Sección 2.2) así como la metodología a utilizar (Sección 2.3). También se presenta cada elemento que se utilizará para el desarrollo del proyecto.

Se presenta el estado actual del Aprendizaje Automático (Sección 2.4), el esquema general de un sistema de Aprendizaje Automático (Sección 2.4.1), en múltiples aspectos, que incluyen la clasificación general de este tipo de sistemas (Sección 2.4.2). Se proporciona el marco teórico de los sistemas de Aprendizaje Automático Supervisado (Sección 2.4.3), dentro de éste apartado tenemos los diferentes métodos de evaluación de los clasificadores que van desde su complejidad hasta el número de soluciones factibles halladas en el proceso de clasificación (Sección 2.4.3.1), del mismo modo se presenta un panorama de los proyectos en los cuales es utilizado éste tipo de aprendizaje (Sección 2.4.3.2); en el último apartado del aprendizaje supervisado se refiere a los principales modelos de clasificación y sus ventajas, así como los proyectos en los cuales han sido aplicados (Sección 2.4.3.3).

En la Sección 2.5 se presenta UML y cada uno de los diagramas que se utilizaron (Casos de uso, Sección 2.5.1; Diagramas de estados, Sección 2.5.2). Posteriormente se hablará de los lenguajes de programación a utilizar (PHP, Sección 2.7; HTML, Sección 2.8; JavaScript, Sección 2.9).

2.2 Ciclo de Vida de Desarrollo de Sistemas de Información

2.2.1 Ingeniería de software

El mundo moderno depende cada día más de los sistemas de software. Los sistemas computacionales están en todas partes, desde los sistemas de freno de

los autos, hasta el control de los hornos de microondas. Hoy en día, las grandes corporaciones conocen la importancia estratégica de sus políticas en relación con las tecnologías de información y el papel fundamental del software como parte de esas tecnologías (Leite 1997).

En los 40's y 50's, ya era común el hardware de propósito general. El software, sin embargo, era diseñado a medida para cada aplicación y tenía una distribución relativamente limitada. El software como producto, es decir, programas desarrollados para ser vendidos a uno o más clientes estaba en su infancia. La mayoría del software era desarrollado y finalmente usado por la misma persona u organización. El diseño era un proceso implícito llevado a cabo en la cabeza de alguien y la documentación era en general inexistente (Pressman 2005). Un programa era considerado exitoso si: a) ejecutaba, b) ejecutaba rápidamente, c) daba una respuesta aceptable, y la calidad dependía altamente de la habilidad del programador. Hacia fines de los 60 comenzaron a producirse problemas recurrentes en el desarrollo de sistemas que se conocen como la "crisis del software", caracterizada por problemas en el desarrollo de software, que provocaban demoras en los plazos de entrega, altos presupuestos, falta de respuesta a los requerimientos de los clientes y/o usuarios, y dificultad para usar, mantener y mejorar los sistemas (Dorfman 2007). A medida que los sistemas se hacían más grandes, su calidad era sospechosa. La gente responsable del desarrollo de software tenía limitados datos históricos para usar como guías y menos control sobre el curso de un proyecto. En 1969, un conjunto de técnicas, llamadas Ingeniería de Software, surgió como respuesta a esa crisis. Estas técnicas ven al software como un producto de ingeniería que requiere planeamiento, análisis, diseño, implementación, testeo y mantenimiento (Pressman 2005). A medida que fueron pasando los años, surgieron diferentes enfoques y metodologías para la elaboración del software una de ellas es el ciclo de vida, que aunque para muchas personas los proyectos de explotación de información poseen características muy distintas a las de los proyectos de desarrollo de software tradicionales. Las clásicas etapas de análisis, diseño, desarrollo, integración y testeo, no encajan con las etapas naturales de los

procesos de desarrollo de proyectos de negocios, sin embargo en investigaciones en el ámbito de la enseñanza y en la elaboración de sistemas para la toma de decisiones son un mecanismo eficaz para la consecución de los objetivos. (García-Martínez 2011)

Para (Mechán 2012) el proceso de análisis de requisitos constituye sin lugar a dudas el mecanismo a partir del cual se establecen las condiciones bajo las cuales se adelantarán las diferentes fases del ciclo de vida de desarrollo de software. Una buena interrelación entre las metas organizacionales, entorno organizacional y los requisitos debe conducir al éxito mediante el logro de mejores indicadores en las restricciones de recursos, tiempo y costo. Se propone y valida un método de análisis de requisitos contemplado bajo la óptica del rediseño de procesos mediante BPM (Business Process Management) el cual en su aplicación para un caso de estudio muestra indicadores de resultado halagadores frente al método tradicional de análisis de unificar las personas, los procesos y la información de una organización así como los productos que del mismo se puedan desprender como: (1) agilización del trabajo de los usuarios, (2) la posibilidad de integrar tanto a empleados, clientes, proveedores y colaboradores, logrando una mayor trazabilidad y un menor número de errores en la información que manipulan y (3) mejor identificación de necesidades y expectativas para el software que los soporta.

El desarrollo de los sistemas de información, comprende un proceso llamado Ciclo de Vida, este permite comprender como un sistema de información puede soportar las necesidades de un proyecto, a través de la secuencia de tareas o fases del proyecto.

Este, aunque es más comúnmente conocido como Modelo en cascada es también llamado "Modelo Clásico", "Modelo Tradicional" o "Modelo Lineal Secuencial".

El Modelo en cascada puro difícilmente se utiliza tal cual, pues esto implicaría un

previo y absoluto conocimiento de los requisitos, la no volatilidad de los mismos (o rigidez) y etapas subsiguientes libres de errores; ello sólo podría ser aplicable a escasos y pequeños desarrollos de sistemas. En estas circunstancias, el paso de una etapa a otra de las mencionadas sería sin retorno, por ejemplo pasar del Diseño a la Codificación implicaría un diseño exacto y sin errores ni probable modificación o evolución: "codifique lo diseñado que no habrán en absoluto variantes ni errores". Esto es utópico; ya que intrínsecamente el software es de carácter evolutivo, cambiante y difícilmente libre de errores, tanto durante su desarrollo como durante su vida operativa. (SENN 1996)

Algún cambio durante la ejecución de una cualquiera de las etapas en este modelo secuencial implicaría reiniciar desde el principio todo el ciclo completo, lo cual redundaría en altos costos de tiempo y desarrollo.

Sin embargo, el modelo cascada en algunas de sus variantes es uno de los actualmente más utilizados, por su eficacia y simplicidad, más que nada en software de pequeño y algunos de mediano porte; pero nunca (o muy rara vez) se lo usa en su forma pura, como se dijo anteriormente. En lugar de ello, siempre se produce alguna realimentación entre etapas, que no es completamente predecible ni rígida; esto da oportunidad al desarrollo de productos software en los cuales hay ciertas incertezas, cambios o evoluciones durante el ciclo de vida. Así por ejemplo, una vez capturados y especificados los requisitos (primera etapa) se puede pasar al diseño del sistema, pero durante esta última fase lo más probable es que se deban realizar ajustes en los requisitos (aunque sean mínimos), ya sea por fallas detectadas, ambigüedades o bien por que los propios requisitos han cambiado o evolucionado; con lo cual se debe retornar a la primera o previa etapa, hacer los pertinentes reajustes y luego continuar nuevamente con el diseño; esto último se conoce como realimentación. Lo normal en el modelo cascada será entonces la aplicación del mismo con sus etapas realimentadas de alguna forma, permitiendo retroceder de una a la anterior (e incluso poder saltar a varias anteriores) si es requerido. (SENN 1996)

Lo dicho es, a grandes rasgos, la forma y utilización de este modelo, uno de los más usados y populares. El modelo Cascada Realimentado resulta muy atractivo, hasta ideal, si el proyecto presenta alta rigidez (pocos o ningún cambio, no evolutivo), los requisitos son muy claros y están correctamente especificados.

Hay más variantes similares al modelo: refino de etapas (más etapas, menores y más específicas) e incluso mostrar menos etapas de las indicadas, aunque en tal caso la faltante estará dentro de alguna otra. El orden de esas fases indicadas en el ítem previo es el lógico y adecuado, pero adviértase, como se dijo, que normalmente habrá realimentación hacia atrás. El modelo lineal o en Cascada es el paradigma más antiguo y extensamente utilizado, sin embargo las críticas a él han puesto en duda su eficacia. Pese a todo tiene un lugar muy importante en la Ingeniería de software y continúa siendo el más utilizado; y siempre es mejor que un enfoque al azar.

Una vez que se han determinado las diferentes fases que comprende el desarrollo de los Sistemas de Información, es importante determinar la metodología de desarrollo. La metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no como hacerlo.

2.3 Metodología Tradicional o Estructurada

Esta metodología generalmente es usada para orientar los desarrollos de sistemas complejos, bajo la conformación de bloques. Cada bloque es una unidad lógica que desempeña una o varias funciones. (SENN 1996)

La descripción de las fases establece:

El Análisis Estructurado, se describe como un método para definir las entradas, procesos y salidas del sistema, generalmente hace uso de herramientas tales como el diagrama de flujo de datos para representar las entradas, los procesos y las salidas.

El Diseño Estructurado comprende las fases de diseño de software, que incluye el diseño de procedimientos, formularios y pantalla, interfaz de usuario, archivos o bases de datos, así como controles y procedimientos.

La Programación Estructurada permite organizar y codificar programas, conformando módulos, los cuales se convierten en una unidad que desempeña una o varias funciones.

2.4 Aprendizaje Automático

El Aprendizaje Automático es un campo de la Informática en el que se estudian y desarrollan algoritmos que implementan los distintos modelos de aprendizaje y su aplicación a la resolución de problemas prácticos (Anaya (2010)). Entre los problemas que son tratados por esta área, tenemos el de inducir conocimientos a partir de datos o ejemplos. Esto resulta una alternativa de solución a problemas que no pueden ser resueltos mediante técnicas o algoritmos tradicionales, entre los cuales podemos mencionar diagnósticos médicos, reconocimiento de patrones, minería de datos en estructuras de información (en particular en bases de datos de gran tamaño), solución de problemas de clasificación en base de datos,

elaboración de horarios, resolución de juegos, entre otros.

El Aprendizaje Automático se enfrenta con el desafío de la construcción de programas computacionales que automáticamente mejoren con la experiencia (Santos 2003). Estos programas computacionales son sistemas de aprendizaje capaces de adquirir conocimientos de alto nivel y/o estrategias para la resolución de problemas mediante ejemplos, en forma análoga a la mente humana. A partir de los ejemplos provistos por un experto o instructor y de los conocimientos de base o conocimientos previos, el sistema de aprendizaje crea descripciones generales de conceptos.

Como podemos observar uno de los aspectos que la mayoría de los autores coinciden es el hecho de que son programas computacionales y que necesitan de la experiencia para su entrenamiento y mejora.

2.4.1 Esquema general de un sistema de Aprendizaje Automático

En general el esquema de un sistema de Aprendizaje Automático se describe a través de la Figura 2-1. El sistema recibe dos grupos de entradas: los ejemplos y los conocimientos previos y genera una descripción de conceptos como salida. Los ejemplos instancian un cierto concepto, lo describen mediante distintos valores de sus atributos. (Sierra 2006)

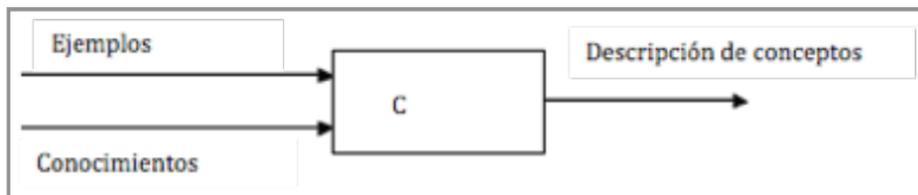


Figura 2-1 Esquema General de sistema de aprendizaje

Los conocimientos previos contienen la información acerca del lenguaje utilizado para describir los ejemplos y los conceptos, es decir, son una parte de metalenguaje. El sistema utiliza entonces, los conocimientos previos para interpretar los ejemplos y para generar descripciones a partir de ellos.

2.4.2 Clasificación del aprendizaje

En el estudio del campo del aprendizaje automático existen infinidad de clasificaciones de diferentes autores pero una de la más conocida y aceptadas es la de (Sierra 2006), en donde existen dos tipos de aprendizaje: el supervisado y el no supervisado. En el aprendizaje supervisado o aprendizaje a partir de ejemplos, el experto define clases y provee ejemplos de cada una. El sistema debe obtener una descripción para cada clase. Cuando el experto define una única clase, provee ejemplos positivos (pertenecen a la clase) y negativos (no pertenecen a la clase). En este caso, los ejemplos importantes son los cercanos al límite, porque proveen información útil sobre los límites de la clase. Cuando el experto define varias clases, el sistema puede optar por realizar descripciones discriminantes o no. Un conjunto de descripciones es discriminante si el total de las descripciones cubren todas las clases, pero una descripción cubre una sola clase en particular.

Cómo podemos observar, una parte importante del aprendizaje supervisado es el experto que define los modelos de entrenamiento y los límites de las clases, es por ello de la trascendencia que tiene el experto dentro del sistema.

En el aprendizaje no supervisado o aprendizaje a partir de observaciones y descubrimientos, el sistema debe agrupar los conceptos sin ayuda alguna de un experto. El sistema recibe los ejemplos, pero no se predefine ninguna clase. Por lo tanto, debe observar los ejemplos y buscar características en común que permitan formar grupos. Como resultado, este tipo de aprendizaje genera un conjunto de descripciones de clases, que juntas cubren todas las clases y en particular describen a una única clase.

2.4.3 Aprendizaje supervisado o clasificación

El aprendizaje supervisado o clasificación es un problema de gran interés para los expertos en Inteligencia Artificial. Los datos de entrada (denominados conjunto de entrenamiento) son instancias de las clases que se desean modelar e incluyen una serie de atributos o características. Como se muestra en la Figura No. 2-2 podemos ver los elementos de un modelo de clasificación.

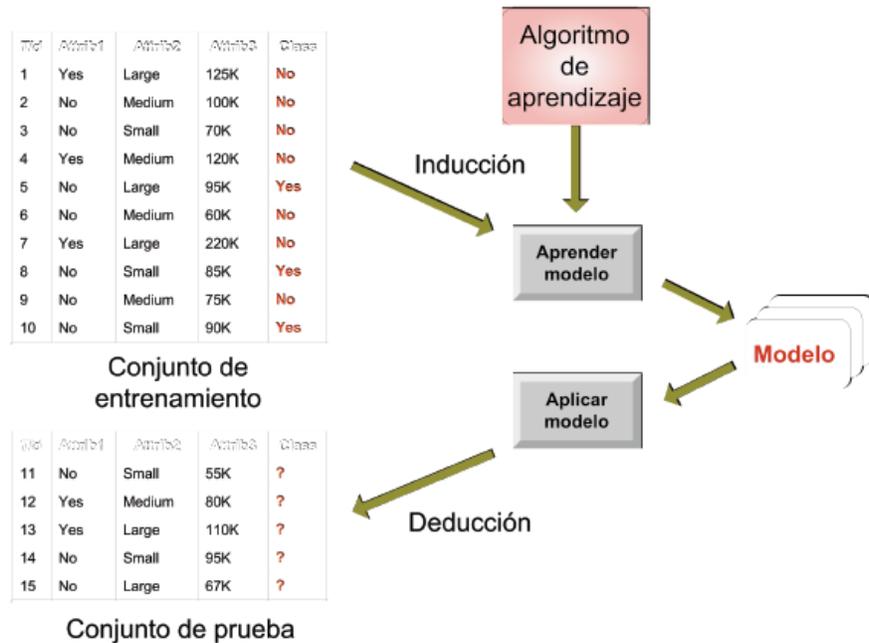


Figura 2-2 Modelo un sistema de clasificación

El objetivo de la clasificación es obtener una descripción precisa para cada clase utilizando los atributos de los datos de entrada. El modelo así obtenido puede servir para clasificar casos cuyas clases se desconozcan o, simplemente, para comprender mejor la información de la que disponemos.

El modelo de clasificación puede construirse entrevistando a expertos que en el caso del proyecto que se va desarrollar es a partir de los asesores reticulares. La mayor parte de los sistemas basados en conocimiento se han construido así a pesar de la dificultad que la extracción manual del conocimiento. No obstante, si

se dispone de suficiente información registrada (como por ejemplo en una base de datos), el modelo de clasificación se puede construir generalizando a partir de ejemplos específicos mediante algún proceso inductivo.

Los casos de entrenamiento utilizados en la construcción del modelo de clasificación suelen expresarse en términos de un conjunto finito de propiedades o atributos con valores discretos o numéricos.

Por su parte, el aprendizaje supervisado permite obtener, a partir de un conjunto de información la cual se puede encontrar en una base de datos, donde se conocen las clases en que se divide dicha información, un conjunto de reglas para poder asociar una nueva observación a una de las clases establecidas. Una de las grandes ventajas de las técnicas de aprendizaje supervisado o clasificadores es que evitan los problemas relacionados con el desarrollo de un modelo explícito de conocimiento o la obtención de algún tipo de modelo, requeridos por otras aproximaciones, así como lo demuestra en su trabajo de “Sistemas de diagnosis en sistemas dinámicos” (D. Borrajo 2005).

2.4.3.1 Evaluación de los modelos de clasificación

La evaluación de un algoritmo de clasificación se puede realizar atendiendo a los distintos aspectos del modelo creado o del proceso utilizado para crearlo y entre las más comunes tenemos las siguientes:

- Precisión. A través de la cual se determinará el porcentaje de casos clasificados correctamente.
- Eficiencia. En donde se medirá el tiempo necesario para construir/usar el clasificador.
- Robustez. En el cual se tomarán aspectos relativos al ruido y valores nulos
- Escalabilidad. Determinada por la utilidad en grandes bases de datos
- Complejidad del modelo de clasificación.

2.4.3.2 Aplicaciones de los modelos de aprendizaje supervisado o clasificadores.

Existen diferentes aplicaciones que han sido implementadas utilizando por ejemplo, redes neuronales o algoritmos de clasificación por vecindad para detectar fraudes en cajeros electrónicos; y en un futuro no muy lejano, se espera un gran crecimiento en la investigación y uso de estas técnicas, entre otros factores debido al auge de tecnologías como Internet y la Web, el crecimiento de los datos corporativos; donde se requiere identificar patrones desconocidos y generar conocimiento que ayude a la toma de decisiones en otras palabras minería de datos, aplicaciones en la generación de horarios para las diferentes instituciones son algunas de las aplicaciones que se están desarrollando utilizando árboles de decisión y algoritmos genéticos. (R.S. Sutton 2000)

2.4.3.3 Modelos de clasificación más comunes

Dentro de los modelos de aprendizaje supervisado más comunes que tenemos y que serán mencionados en el presente trabajo, y que de alguna manera podrían ser parte de la solución son:

- Árboles de Decisión
- Reglas de Producción
- Reglas de Asociación
- Algoritmos Genéticos

2.4.3.4 Árboles de Decisión

La construcción de árboles de decisión, también denominados árboles de clasificación o de identificación, es sin duda el método de aprendizaje automático más utilizado.

Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial, dada una base de datos se construyen estos diagramas de

construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema.

Un árbol de decisión tiene unas entradas las cuales pueden ser un objeto o una situación descrita por medio de un conjunto de atributos y a partir de esto devuelve una respuesta la cual en últimas es una decisión que es tomada a partir de las entradas. Los valores que pueden tomar las entradas y las salidas pueden ser valores discretos o continuos. Se utilizan más los valores discretos por simplicidad, cuando se utilizan valores discretos en las funciones de una aplicación se denomina clasificación y cuando se utilizan los continuos se denomina regresión.

Un árbol de decisión lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. El árbol de decisión suele contener nodos internos, nodos de probabilidad, nodos hojas y arcos. Un nodo interno contiene un test sobre algún valor de una de las propiedades. Un nodo de probabilidad indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema, este tipo de nodos es redondo, los demás son cuadrados. Un nodo hoja representa el valor que devolverá el árbol de decisión. Y finalmente las ramas brindan los posibles caminos que se tienen de acuerdo a la decisión tomada.

El dominio de aplicación de los árboles de decisión no está restringido a un ámbito concreto sino que pueden ser utilizados en diversas áreas (desde aplicaciones de diagnóstico médico hasta juegos como el ajedrez o sistemas de predicción meteorológica) o como sistemas de generación de horarios bajo demanda motivo de la presente investigación.

El conocimiento obtenido en el proceso de aprendizaje se representa mediante un árbol en el cual cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada hoja del árbol se refiere a una decisión (una clasificación). Un árbol de decisión puede usarse para clasificar un caso comenzando desde su raíz y siguiendo el camino determinado por las

respuestas a las preguntas de los nodos internos hasta que encontremos una hoja del árbol.

Dentro de las ventajas que podemos observar con la aplicación de los árboles de decisión tenemos:

- Los resultados que se presentan son fáciles de entender y fáciles de interpretar.
- Durante el proceso de construcción no tienen problemas para trabajar con datos perdidos.
- Se hace automático la selección de variables.
- Es rápido de calcular.

Resulta impactante el hecho de que los tres primeros finalistas del torneo de “Knowledge Discovery in Databases 1999” (KDD-99) utilizaran árboles de decisión como método de clasificación. El ganador del concurso, Bernhard Pfahringer (Elías-Gutiérrez 2012) utilizó como solución una combinación de bagging (cambian la distribución de los ejemplos de entrenamiento) y boosting (no cambian la distribución de los ejemplos de entrenamiento).

El segundo clasificado del concurso fue Levin. Utilizando la herramienta Kernel Miner creó un conjunto de árboles de decisión localmente óptimos (llamados bosque de decisión), de donde se seleccionaban aquellos óptimos para predecir nuevas clases (sub-bosque). Sinclair et al. Utilizaron árboles de decisión ID3 junto con algoritmos genéticos para generar reglas automáticamente con el fin de clasificar conexiones de red y aplicar las reglas mediante un sistema experto. Agarwal and Joshi propusieron un marco de dos fases (general-a-específico) para el aprendizaje de un modelo basado en reglas (PNrule) para aprender modelos de clasificadores en un conjunto de datos con distribuciones de clases muy diferentes en el conjunto de entrenamiento. (Anaya (2010)

2.4.3.5 Reglas de producción

Conforme el tamaño los árboles de decisión aumenta, su inteligibilidad disminuye, cuando el problema de clasificación es complejo, el árbol de decisión generado es tan grande que ni siquiera los expertos pueden comprender el modelo de clasificación construido (ni siquiera simplificándolo al podar el árbol).

(Sabrina Sestito 2001) propuso descomponer un árbol de decisión complejo en una jerarquía de pequeños árboles de decisión para obtener un modelo más comprensible. Sin embargo, es mucho más sencillo expresar el árbol de decisión construido como un conjunto de reglas de producción, una forma de representación del conocimiento más inteligible que los árboles.

Las reglas de producción se pueden derivar de un árbol de decisión con facilidad. El algoritmo que nos permite realizar este cambio de modelo de representación es muy sencillo: de cada camino desde la raíz del árbol hasta un nodo hoja se deriva una regla cuyo antecedente es una conjunción de literales relativos a los valores de los atributos situados en los nodos internos del árbol cuyo consecuente es la decisión a la que hace referencia la hoja del árbol (la clasificación realizada).

También existen otros métodos de clasificación que obtienen reglas de producción directamente, sin necesidad de construir previamente un árbol de decisión. Estas técnicas, más ineficientes, suelen emplear estrategias de búsqueda heurística como la búsqueda dirigida, una variante de la búsqueda primero el mejor.

Michalski y sus colaboradores desarrollaron, bajo el nombre de metodología STAR, un conjunto de técnicas de aprendizaje inductivo incremental basadas en la utilización de expresiones lógicas en forma normal disyuntiva (modelo de representación más expresivo que el empleado por los algoritmos de construcción de árboles de decisión). Estas expresiones lógicas describen conceptos y se pueden utilizar directamente como reglas de clasificación. Entre los algoritmos desarrollados en los años 00 en el entorno de Michalski destacan INDUCE y AQ, del que existen múltiples variantes.

Los sistemas basados en reglas son los más comúnmente utilizados. Su simplicidad y similitud con el razonamiento humano, han contribuido para su popularidad en diferentes dominios. Las reglas son un importante paradigma de representación del conocimiento.

Las reglas representan el conocimiento utilizando un formato SI-ENTONCES (IF-THEN), es decir tienen 2 partes:

- La parte SI (IF), es el antecedente, premisa, condición o situación;
- La parte ENTONCES (THEN), es el consecuente, conclusión, acción o respuesta.

Las reglas pueden ser utilizadas para expresar un amplio rango de asociaciones, por ejemplo:

- Si está manejando un vehículo Y se aproxima una ambulancia, ENTONCES baje la velocidad Y hágase a un lado para permitir el paso de la ambulancia.
- Si su temperatura corporal es de 39 oC, ENTONCES tiene fiebre.
- Si el drenaje del lavabo está tapado Y la llave de agua está abierta, ENTONCES se puede inundar el piso.

Una declaración de que algo es verdadero o es un hecho conocido, es una afirmación (fact). El conjunto de afirmaciones se conoce a menudo con el nombre de memoria de trabajo o base de afirmaciones. De igual forma, al conjunto de reglas se lo denomina base de reglas.

Un sistema basado en reglas utiliza el modus ponens para manipular las afirmaciones y las reglas durante el proceso de inferencia. Mediante técnicas de búsqueda y procesos de unificación, los sistemas basados en reglas automatizan sus métodos de razonamiento y proporcionan una progresión lógica desde los

datos iniciales, hasta las conclusiones deseadas. Esta progresión hace que se vayan conociendo nuevos hechos o descubriendo nuevas afirmaciones, a medida que va guiando hacia la solución del problema.

En consecuencia, el proceso de solución de un problema en los sistemas basados en reglas va realizando una serie de inferencias que crean un sendero entre la definición del problema y su solución. Las inferencias están concatenadas y se las realiza en forma progresiva, por lo que se dice que el proceso de solución origina una cadena de inferencias.

Los sistemas basados en reglas difieren de la representación basada en lógica en las siguientes características principales:

- Son en general no-monotónicos, es decir hechos o afirmaciones derivadas, pueden ser retractados, en el momento en que dejen de ser verdaderos.
- Pueden aceptar incertidumbre en el proceso de razonamiento.

El proceso de razonamiento en un sistema basado en reglas es una progresión desde un conjunto inicial de afirmaciones y reglas hacia una solución, respuesta o conclusión. Como se llega a obtener el resultado, sin embargo, puede variar significativamente:

- Se puede partir considerando todos los datos conocidos y luego ir progresivamente avanzando hacia la solución. Este proceso se lo denomina guiado por los datos o de encadenamiento progresivo (forward chaining).
- Se puede seleccionar una posible solución y tratar de probar su validez buscando evidencia que la apoye. Este proceso se denomina guiado por el objetivo o de encadenamiento regresivo (backward chaining).

A pesar de las desventajas anotadas, los sistemas basados en reglas han permanecido como los esquemas más comúnmente utilizados para la

representación del conocimiento. Como ventajas significativas se pueden mencionar las siguientes: modularidad, uniformidad y naturalidad para expresar el conocimiento.

2.4.3.6 Reglas de asociación

Muchas veces en la vida real no se pueden construir modelos completos que nos permitan una clasificación perfecta de todos los casos con los que uno se pueda encontrar. A veces hay que conformarse con descubrir modelos aproximados, los cuales contemplan algunas características de las distintas clases sin que el modelo abarque todas las clases posibles ni todos los casos particulares de una clase determinada.

La construcción de un modelo de clasificación completo puede no ser factible cuando hemos de tratar con una gran cantidad de atributos, cuando muchos valores son desconocidos, cuando unos atributos deben modelarse en función de otros o cuando el número de casos de entrenamiento es excesivamente elevado. Los árboles de decisión no son muy adecuados para tratar con información incompleta (valores desconocidos en atributos de los casos de entrenamiento) y resultan problemáticos cuando unos atributos son función de otros. Las redes neuronales tampoco son apropiadas cuando tenemos información incompleta y, además, su entrenamiento puede llegar a consumir demasiado tiempo.

Por su parte, un modelo de clasificación parcial intenta descubrir características comunes a los distintos casos de cada clase sin la necesidad de formar un modelo predictivo completo. La extracción de reglas de asociación puede ser útil para resolver problemas de clasificación parcial donde las técnicas de clasificación clásicas no son efectivas.

Las redes asociativas se caracterizan por representar el conocimiento en forma gráfica. Agrupan una porción de conocimiento en dos partes: objetos y relaciones entre objetos. Los objetos se denominan también nodos (elementos del

conocimiento) y las relaciones entre nodos se denominan enlaces o arcos. Cada nodo y cada enlace en una red semántica, deben estar asociados con objetos descriptivos.

Las redes asociativas tienen dos ventajas sobre los sistemas basados en reglas y sobre los basados en lógica:

- Permiten la declaración de importantes asociaciones, en forma explícita y sucinta.
- Debido a que los nodos relacionados están directamente conectados, y no se expresan las relaciones en una gran base de datos, el tiempo que toma el proceso de búsqueda por hechos particulares puede ser significativamente reducido.

Entre las desventajas de las redes asociativas, se pueden mencionar:

- No existe una interpretación normalizada para el conocimiento expresado por la red. La interpretación de la red depende exclusivamente de los programas que manipulan la misma.
- La dificultad de interpretación a menudo puede derivar en inferencias inválidas del conocimiento contenido en la red.
- La exploración de una red asociativa puede derivar en una explosión combinatoria del número de relaciones que deben ser examinadas para comprobar una relación.

Las reglas de asociación han sido ampliamente utilizadas para la detección de intrusos. Su objetivo es el de derivar correlaciones de varias características en una tabla de una base de datos, dada la evidencia de que frecuentemente los datos obtenidos de ejecuciones de programas y actividades de usuarios muestran correlaciones. Mediante las reglas de asociación se pueden encontrar relaciones internas entre datos de una misma conexión. (Stolfo 1998) Lee y Stolfo presentan sus algoritmos de reglas de asociación y episodios frecuentes (para descubrir un

conjunto de eventos que ocurrían en una ventana temporal) utilizados para calcular un conjunto de patrones a partir de datos de auditoría. Daniel Barbará y su grupo también hacen uso de esta técnica en su proyecto ADAM pero como método de detección de anomalías. ADAM utiliza esta técnica de data mining para encontrar eventos anómalos en datos de conexiones de tráfico de red para luego clasificar dichos eventos o reglas de asociación inesperadas en instancias normales o anómalas. MINDS de la Universidad de Minnesota también utiliza reglas de asociación para caracterizar las anomalías detectadas, y ayudar en la creación de nuevas firmas y modelos de ataques emergentes. En IDDM (Intrusion Detection Data Mining) realizado por el departamento de defensa australiano, también incorporan un algoritmo de reglas de asociación a los agentes software [Abr03]. Finalmente, el trabajo realizado por Min Qin y Kai Hwang también está basado en la utilización de reglas de asociación y episodios frecuentes para la detección de anomalías . (Hwang 2004)

2.4.3.7 Algoritmos Genéticos

Los algoritmos genéticos están inspirados en la Naturaleza, en la teoría de la evolución descrita por Charles Darwin en su libro “Sobre el Origen de las Especies por medio de la Selección Natural”, escrito 20 años después del viaje de su autor por las islas Galápagos en el Beagle. La hipótesis de Darwin (y de Wallace, que llegó a las mismas conclusiones independientemente) es que pequeños cambios heredables en los seres vivos y la selección son los dos hechos que provocan el cambio en la Naturaleza y la generación de nuevas especies. Fue Mendel quien descubrió que los caracteres se heredaban de forma discreta, y que se tomaban del padre o de la madre, dependiendo de su carácter dominante o recesivo. A estos caracteres que podían tomar diferentes valores se les llamaron genes, y a los valores que podían tomar, alelos. En los seres vivos, los genes están en los cromosomas. En la evolución natural, los mecanismos de cambio alteran la proporción de alelos de un tipo determinado en una población, y se dividen en dos tipos: los que disminuyen la variabilidad (la selección natural y la deriva genética), y los que la aumentan (la mutación, la poliploidía, la recombinación o cruce y el

flujo genético). A principios de los 60, en la Universidad de Michigan en Ann Arbor, las ideas de John Holland comenzaron a desarrollarse y a dar frutos. Leyendo un libro escrito por un biólogo evolucionista, R.A. Fisher, titulado “La teoría genética de la selección natural”, aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado. Los objetivos de su investigación fueron dos: imitar los procesos adaptativos de los sistemas naturales y diseñar sistemas artificiales (programas) que retengan los mecanismos de los sistemas naturales.

Los algoritmos evolutivos tratan de imitar los mecanismos de la evolución para resolver problemas. La aplicación de un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma y aplicar los métodos de la evolución (selección y reproducción sexual con intercambio de información y alteraciones que generen diversidad). La mayoría de las veces una codificación correcta es la clave de una buena resolución del problema.

Los algoritmos genéticos en sí son métodos de optimización. En el algoritmo genético va implícito el método para resolver el problema. Un algoritmo genético es independiente del problema, lo cual lo hace robusto, por ser útil para cualquier problema, pero a la vez débil, pues no está especializado en ninguno.

Como método general de resolución de problemas que son, los algoritmos genéticos también pueden utilizarse para resolver problemas de clasificación. Al igual que las redes neuronales y los clasificadores basados en medidas de similitud, los clasificadores construidos utilizando algoritmos genéticos suelen destacar porque su rendimiento no se ve excesivamente afectado por la aparición de errores en los casos de entrenamiento (lo que sí ocurre con determinados modelos simbólicos. (Bhavani Thuraisingham 2008)

2.4.3.8 Otros modelos – Representaciones basadas en ejemplos

Los sistemas de aprendizaje basado en ejemplos (Instance-Based Learning algorithms) representan el conocimiento mediante ejemplos representativos, basándose en "similitudes" entre los datos. El aprendizaje consiste en la selección de los ejemplos que mejor representan a los conceptos existentes en la base de datos (se trata de aprendizaje supervisado); estos ejemplos representativos serán los únicos que se almacenen, reduciendo así considerablemente el espacio necesario. El principal problema de estos sistemas es que se necesita una función de "similitud", a veces difícil de definir, para clasificar los nuevos ejemplos según sea su parecido con los ejemplos prototipo. (Stolfo 1998)

Los algoritmos de aprendizaje basado en ejemplos surgieron a partir de los clasificadores por vecindad (nearest-neighbor classifier), y han adquirido importancia más recientemente con los sistemas de razonamiento basado en casos (case-based reasoning), para diagnóstico y resolución de problemas. Además, pueden utilizarse como paso previo a otros sistemas de aprendizaje a partir de ejemplos, para entrenarlos con conjuntos de ejemplos más pequeños y representativos.

2.5 UML (Lenguaje Unificado de Modelado)

El lenguaje unificado de Modelado (UML), es una herramienta que se utiliza para el Análisis y Diseño Orientado a Objetos marca una serie de pasos para construir un sistema de información. (SCHACH 2005)

UML es un lenguaje adaptable, contiene un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML, ofrece

nueve diagramas para modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

Para el análisis de este proyecto solo se utilizarán los diagramas de casos de uso, diagramas de actividad y diagramas de clases.

2.5.1 Diagramas de Casos de Uso

Los diagramas de casos de uso son la principal herramienta para modelar y definir el comportamiento del sistema. Así mismo, describe como los usuarios interactúan para ejecutar alguna actividad. Permiten también a los analistas modelar la interacción de un sistema de información y su ambiente. (SCHACH 2005)

El ambiente de un sistema de información incluye al usuario final (actores) y el sistema de información mismo. Un conjunto de casos de uso, componen un diagrama de casos de uso. El uso principal de los diagramas de casos de uso, es para proporcionar un significado, documentar y entender los requerimientos para el desarrollo de los sistemas de información.

Los casos de uso y los diagramas de casos de uso, se consideran como una de

las herramientas más importantes que son usadas para el análisis y diseño de sistemas.

Los diagramas de clases, tienen como propósito el establecer un vocabulario que es usado entre el analista y el usuario. Generalmente representan pensamientos, ideas o conceptos que están contenidos en la aplicación.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grady Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares. Diagramas de Estados

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

Los diagramas de estado ven a los objetos como máquinas de estado o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito. Por ejemplo, un objeto de tipo NetServer puede tener durante su vida uno de los siguientes estados.

- Listo
- Escuchando
- Trabajando
- Detenido

y los eventos que pueden producir que el objeto cambie de estado son

- Se crea el objeto
- El objeto recibe un mensaje de escucha
- Un cliente solicita una conexión a través de la red
- Un cliente finaliza una solicitud

- La solicitud se ejecuta y ser termina
- El objeto recibe un mensaje de detención

2.5.2 Diagramas de Clases

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

Los diagramas de Clases por definición son estáticos, esto es, representan que partes interactúan entre sí, no lo que ocurre cuando. Estos diagramas se componen de los siguientes elementos.

- Propiedades también llamados atributos o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso.
- Operaciones son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.
- Interfaz es un conjunto de operaciones y/o propiedades que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto.
- Herencia se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre. Por ejemplo: Una persona puede subdividirse en Proveedores, Acreedores, Clientes, Accionistas, Empleados; todos comparten datos básicos como una persona, pero además tendrá información adicional que depende del tipo de persona, como saldo del cliente, total de inversión del accionista, salario del empleado, etc.

2.5.3 Bases de Datos

La base de datos es un elemento esencial en los Sistemas de Información y forma parte del Diseño de los sistemas. Se define como: “Un conjunto de datos organizados sistemáticamente que brindan cierta información”. (Kendall 2006)

Básicamente, la función principal de la base de datos es la de almacenar los datos que son manipulados por el sistema. La base de datos, es una colección de archivos interrelacionados, que son creados con un Sistema Manejador de Base de Datos. El contenido de una base de datos engloba la información de una organización, de tal manera que los datos estén disponibles para los usuarios.

Un Sistema de base de datos, tiene como objetivo el proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente. Existen diferentes niveles de abstracción que simplifican la interacción de los usuarios con el sistema, los cuales se describen a continuación:

- Nivel Físico: Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en cómo de almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).
- Nivel Conceptual: Describe que datos son almacenados en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos. Este nivel incluye los siguientes conceptos:
 - Definición de los datos: Se describen el tipo de datos y la longitud de campo.
 - Relaciones entre datos: Permite definir las relaciones entre datos y

registros. Nivel de visión: Es el nivel más alto de abstracción, es lo que el usuario final puede visualizar el sistema terminado. El sistema es capaz de proporcionar muchas vistas para la misma base de datos.

Los diferentes niveles de abstracción, permiten establecer diferentes modelos para el diseño de la base de datos. Un modelo de base de datos, “se considera como una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, la semántica asociada a los datos y las restricciones de consistencia”. (Korth 2006)

Entre estos modelos, se consideran los siguientes:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.
- Modelos físicos de datos.

Los modelos lógicos basados en objetos, se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo se representan los datos tal y como se captan del mundo real. Existen diferentes modelos de este tipo, entre los más utilizados se establecen: el modelo Entidad-Relación y el orientado a objetos.

El modelo entidad-relación (E-R) se basa en una percepción del mundo compuesta por objetos, llamados entidades, y relaciones entre ellos. Las entidades se diferencian unas de otras a través de atributos.

El orientado a objetos también se basa en objetos, los cuales contienen valores y métodos, entendidos como órdenes que actúan sobre los valores, en niveles de anidamiento. Los objetos se agrupan en clases, relacionándose mediante el envío de mensajes.

Los Modelos lógicos basados en registros se usan para especificar la estructura lógica completa de la base de datos. Proporcionan una descripción de alto nivel de la implantación. Básicamente se estructuran a través de registros de formato fijo de diferentes tipos. En cada tipo de registro se define un número de campos, cada uno con longitud fija.

Los principales modelos de esta categoría son: el modelo relacional, el de red y el jerárquico.

Los Modelos físicos de la base de datos, implican la asignación del diseño lógico en medios físicos y aprovecha las características disponibles de hardware y software que permiten obtener acceso físico a los datos y mantenerlos de la forma más rápida posible.

Hoy en día, el modelo de base de datos relacional, es el más usado y es considerado como la mejor solución dado que permite almacenar y recuperar datos. En congruencia, este modelo incorpora diferentes conceptos que los otros modelos utilizan.

2.6 PHP HyperText Preprocessor

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. (Welling y Thomson 2005)

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP

al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Este mismo sitio web de Wikipedia está desarrollado en PHP. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La más reciente versión principal de PHP fue la versión 5.2.8 del 08 de diciembre de 2008.

2.7 HTML HyperText Markup Language

Se traduce al español como Lenguaje de Marcas Hipertextuales. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para Web. (Pavón 2006)

2.7.1 Nociones Básicas de HTML

El lenguaje HTML puede ser creado y modificado con cualquier editor de textos básico, como puede ser Gedit, el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, Textpad, Vim, entre otros.

Existen además, otros programas para la realización de sitios Web o edición de

código HTML, como por ejemplo Microsoft FrontPage, el cual tiene un formato básico parecido al resto de los programas de Office. También existe el famoso Software de Macromedia (que adquirió la empresa Adobe) llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web. A estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: "lo que ves es lo que obtienes"). Esto significa que son editores en los cuales se ve el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios Web, sino que una forma un tanto más simple, ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML, la cual va generando todo el código a medida que se va trabajando.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo; si se edita todo en HTML y de pronto se olvida algún código o etiqueta, simplemente me dirijo al editor visual o WYSIWYG y se continúa ahí la edición, o viceversa, ya que hay casos en que sale más rápido y fácil escribir directamente el código de alguna característica que queramos adherirle al sitio, que buscar la opción en el programa mismo.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (< >) y algunas tienen atributos que pueden tomar algún valor.

Esta herramienta es la base de nuestra aplicación porque sin ella no se puede decir que el Sistema sea Web, ya que los navegadores de Internet, tales como Internet Explorer, Opera, Firefox, Netscape o Safari, no pueden visualizar

contenido que no sea HTML. Si bien es cierto que pueden interpretar otro tipo de códigos, mediante algún plugin, la base para la cual fueron diseñados es la misma: traducir la notación HTML y mostrar al usuario lo que el programador hizo. Teniendo en cuenta esto, cada página que se visualice en el programa estará codificada mediante HTML para la interpretación del navegador. (Welling y Thomson 2005)

2.8 JavaScript

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

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML. (Zakas 2003)

2.9 XML Lenguaje de Marcas

XML, sigla en inglés de Extensible Markup Language («lenguaje de marcas ampliable»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (Welling y Thomson 2005)

2.10 Hojas Estilo en cascada

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Por ejemplo, el elemento de HTML <H1> indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como <H2>. Versiones más antiguas de HTML permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente). No obstante, cada etiqueta <H1> debía disponer de la información si se deseaba un diseño consistente para una página, y además, una persona que lea esa página con un navegador pierde totalmente el control sobre la visualización del texto.

Cuando se utiliza CSS, la etiqueta <H1> no debería proporcionar información sobre como va a ser visualizado, solamente marca la estructura del documento. La información de estilo separada en una hoja de estilo, especifica cómo se ha de mostrar <H1>: color, fuente, alineación del texto, tamaño, y otras características no visuales como definir el volumen de un sintetizador de voz, por ejemplo. (Pavón 2006)

2.11 AJAX Asynchronous JavaScript And XML

El término AJAX es un acrónimo de Asynchronous JavaScript And XML, que se puede traducir como “JavaScript asíncrono y XML”.

AJAX no es una tecnología en sí mismo. En realidad se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

2.11.1 Cómo funciona AJAX

Una aplicación AJAX elimina la naturaleza arrancar-frenar- arrancar-frenar de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario.

En vez de cargar un página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente sacado en un frame oculto). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo. (Dave Crane 2006)

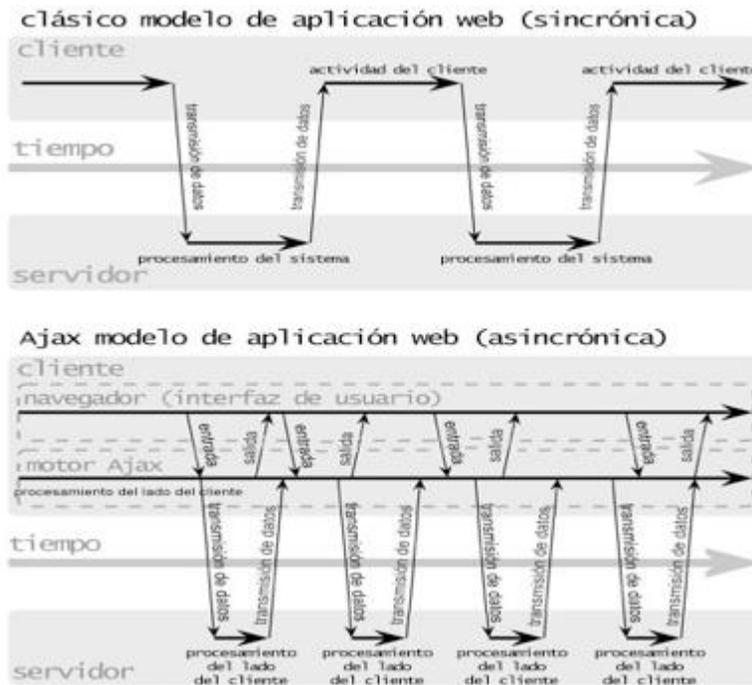


Figura 2-3 Comparación tecnología síncrona - asíncrona

Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera de un viaje de vuelta al servidor (como una simple validación de datos, edición de datos en memoria, incluso algo de navegación) es manejada por su cuenta. Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

El tipo de interacción que ofrece AJAX es la que se integró en el sistema, ya que mediante esta herramienta se pierde menos tiempo en cuestión de validaciones y otro tipo de acciones que no requieran que se establezca una conexión con el servidor como se muestra en la figura 2-3, de esta manera se ahorra tiempo y el usuario sólo debe esperar cuando en realidad tenga que hacer una petición al servidor.

3 Análisis

3.1 Introducción

En este Capítulo se hablará de la encuesta realizada con anterioridad a un determinado grupo de alumnos (Sección 3.2), por otra parte se analizarán las estadísticas resultantes de las encuestas (Sección 3.2.1).

También se desarrollarán los diagramas de Casos de Uso que muestran cómo debería interactuar el usuario con el sistema (Sección 3.5), los diagramas de estados que detallan cada uno de las etapas del sistema y la transición entre sí (Sección 3.6).

Por último se hablará de los requerimientos mínimos y recomendables que el usuario deberá disponer para poder utilizar el sistema (3.7); así como también los requisito para el desarrollo de este proyecto (Sección 3.8).

3.2 Encuesta

Es un estudio de opinión que sirve para conocer el punto de vista o las necesidades de un determinado grupo de personas. (Kendall 2006)

Para conocer las necesidades de los estudiantes del Instituto Tecnológico Superior de Xalapa, se llevó a cabo una encuesta la cual consistió en aplicar un cuestionario a una muestra de 600 alumnos del plantel, para la selección de los estudiantes se tomaron en cuenta a los tres horarios de clases con los que cuenta la institución y a tanto jóvenes con problemas académicos y regulares.

A continuación se muestra el formato con las preguntas y al final los resultados, ya graficados obtenidos en la encuesta.

Instituto Tecnológico Superior de Xalapa

Encuesta para conocer cuáles son algunos de los criterios que un alumno toma en cuenta para la selección de su horario.

Nombre: _____ Fecha: _____

Instrucción. Subrayar la respuesta que más se acerque a los criterios que tomas en cuenta para la elección de horario.

1.- ¿Qué turno te gusta más?

- a) Matutino
- b) Vespertino

2- ¿Prefieres el horario por bloques o intercalado?

- a) Bloques
- b) Intercalado

3.- Eliges el horario de las materias tomando en cuenta alguno de los siguientes factores:

- a) Reputación del maestro.
- b) La hora de la materia
- c) Por influencia de tus compañeros.

4.- ¿Tienes alguna actividad que influya en la creación de tu horario?

- a) Trabajo
- b) Levantarse Tarde
- c) Actividades Deportivas
- d) Clases Externas
- e) Viajar

5.- ¿Qué otros factores son relevantes para ti a la hora de crear tu horario?

- a) Horario corrido / Sin Horas vacías
- b) Disponibilidad de materias / Créditos
- c) Empalmes
- d) Vive lejos
- e) Tiempo para desayunar y/o comer
- f) Actividades diarias
- g) Ociosidad y esparcimiento
- h) Tiempo
- i) Otros. Especificar: _____

6.- ¿Quisieras que al inscribirte un programa eligiera tu horario en base a tus respuestas?

- a) Sí
- b) No

3.2.1 Estadísticos de la Encuesta

En esta parte se mostrarán los resultados obtenidos al aplicar la encuesta anterior a los estudiantes del **ITSX**, con el fin de conocer más a fondo las preferencias de estos y saber qué es lo que toman en cuenta para la creación y/o elección de los horarios. De esta manera se creará un sistema de acuerdo a las necesidades reales del usuario.

Turno Preferido

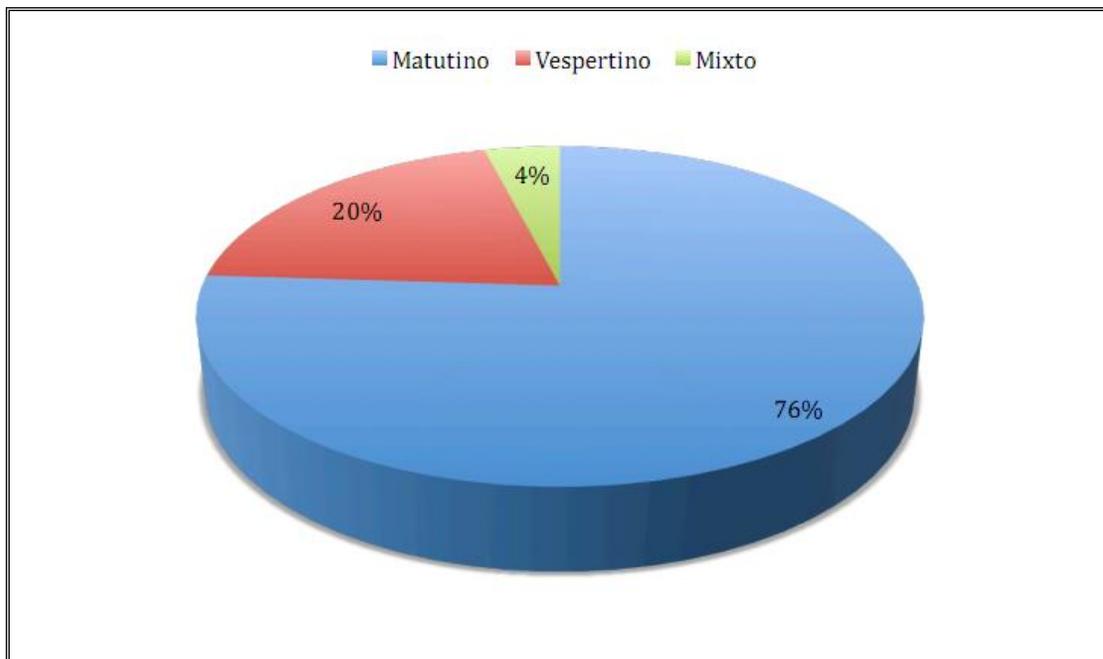


Figura 3-1 Turno Preferido

Los datos revelados en la Figura No. 3-1 nos dicen que el 76% de los estudiantes prefieren estar en el turno de la mañana, el otro 20% en la tarde y sólo el 4% prefieren un turno mixto. De acuerdo a esto, los primeros horarios que se mostrarán en el sistema serán los que estén en la mañana, de esta manera se agilizará el proceso para la mayoría de alumnos.

Preferencia de horario

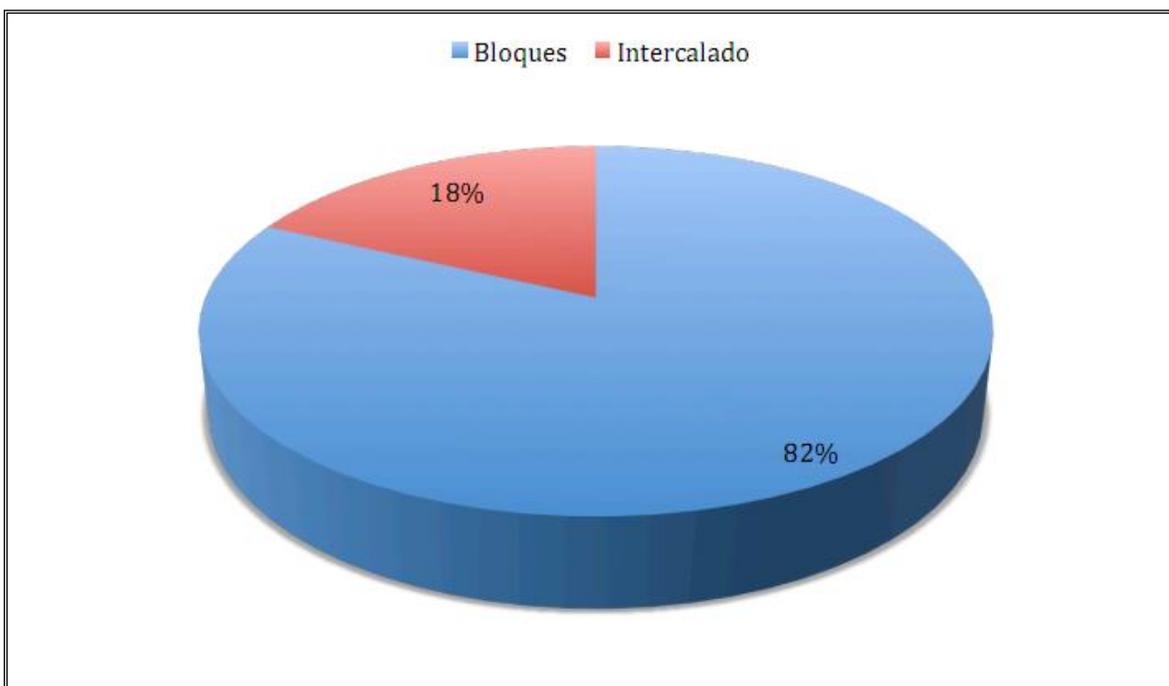


Figura 3-2 Preferencia de horario

En la Figura No. 3-2 nos muestra que el 82% de los alumnos prefiere los horarios por bloque, así que el sistema ordenará los horarios por bloques ya que de esta manera un 18% tardará más tiempo en elegir su horario, ya que lo prefieren mixto.

De acuerdo con la Figura No. 3-3 uno de los factores que toma en cuenta el 45% de los alumnos en el momento de elegir su horario es la hora en que se imparte esa materia, también influye en un 41% quién es el maestro que imparte esa clase, y hay personas que se dejan influenciar por parte de sus amigos, para elegir otro horario. Se crearán filtros para que el usuario pueda personalizar el horario de acuerdo a sus necesidades.

Factores a la hora de elegir horario



Figura 3-3 Factores a la hora de elegir horario

Actividades que influyen en la creación del horario

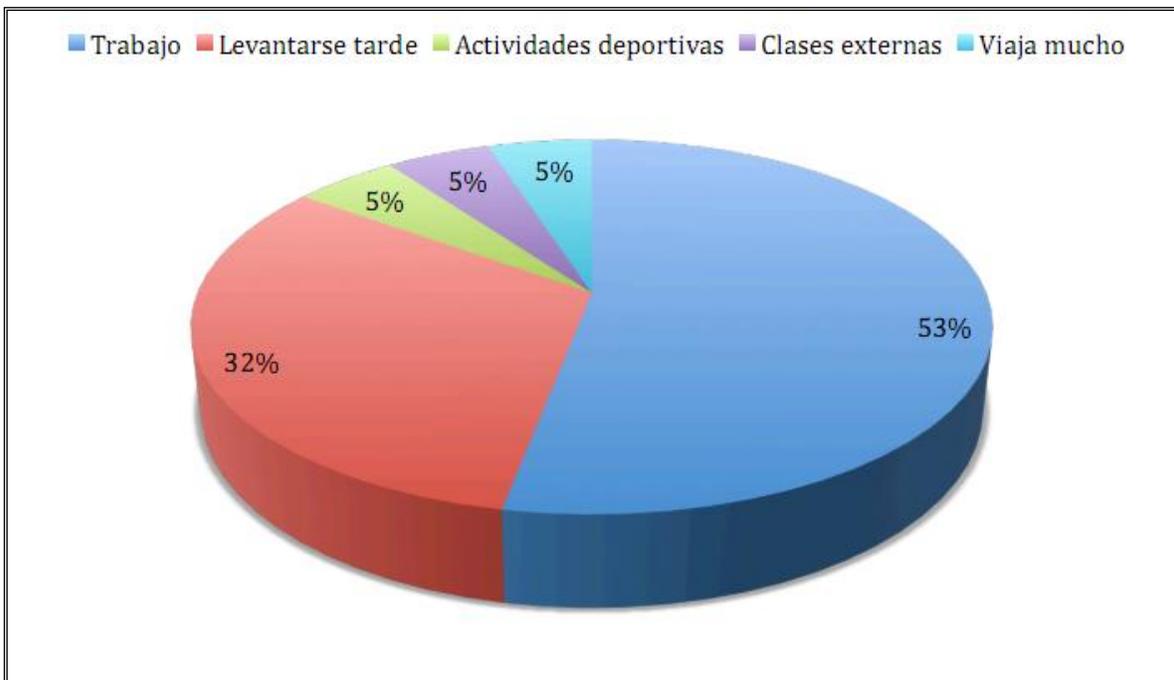


Figura 3-4 Actividades que influyen en la creación del horario

En la Figura No. 3-4 se deja ver que para el 53% de los alumnos el mayor factor que influye en la creación del horario es que trabajan, y un 32% que prefiere no levantarse demasiado temprano.

Aceptación del sistema de horarios

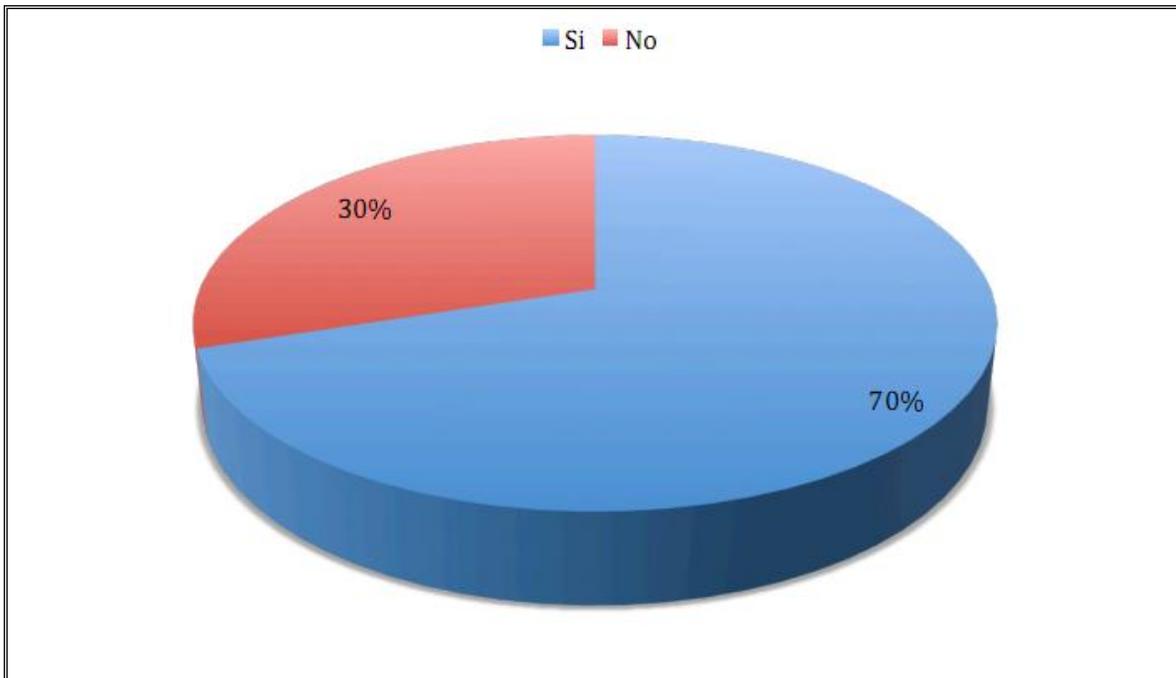


Figura No. 3-5 Aceptación del sistema de horarios

Existe un 5% que no aceptará el sistema de horarios para el alumno, pero el otro 95% están de acuerdo en que se debería de hacer de acuerdo a la Figura 3-5.

3.3 Módulos del Sistema

El proyecto consistió en la búsqueda de una carga académica en base a las materias que se tengan en la base de datos, validando a su vez que no haya empalmes en el horario, esto es, que no haya materias a la misma hora el mismo día dentro del mismo grupo.

Con esto se planea automatizar el sistema actual de inscripciones, el cual, hasta estas fechas se realizan de forma manual, donde cada alumno debe llenar un formato que se le proporciona e ir verificando de forma manual que en su horario no haya algún empalme, para después pasar el horario a un asesor y que dé el visto bueno de este horario. Este proceso requiere de mucho tiempo y esfuerzo por parte de los alumnos y de los maestros, además que está sujeto al punto de vista del maestro. Para la escuela el proceso representaba también un desembolso de recursos económicos, ya que proporcionaba tanto a los maestros como al personal administrativo los alimentos de aproximadamente dos semanas durante el proceso de inscripción.

Este proyecto está enfocado a los alumnos, ya que se creó una herramienta que va a satisfacer las demandas de éstos, recortando el tiempo a la hora de crear su horario, además de brindarle una serie de opciones ordenadas con grupos donde éste pase la menor cantidad de tiempo en la escuela y el menor número de horas libres. Por otra parte, también se creó un espacio para los docentes en el cual puedan consultar su carga académica, así como también la lista de alumnos de cada una de las materias que imparta para que pueda disponer de ella en el momento que necesite. De la misma manera, se le proporcionó al administrador la facilidad de consultar los horarios de los docentes, así como las listas de alumnos de cada materia de la carrera.

Para la realización de todo lo anterior el sistema cuenta con los siguientes módulos:

Módulo que verifica la veracidad de los datos del alumno.

Módulo de creación de una lista de diversas cargas académicas con base a una serie de materias elegidas por el alumno.

Módulo de verificación de cupo en cada grupo de la lista de cargas académicas.

Módulo de creación de una bitácora para cada movimiento del alumno.

Módulo de guardado de los datos elegidos.

Módulo de creación de estadísticas seleccionadas por el docente.

Módulo de Login para docente.

3.4 Lista de Procesos

Aquí se abordó el sistema dividiéndolo en los procesos que se ocupan en él, tales como el funcionamiento y la disponibilidad.

- **Verificación Datos.**- El alumno debe introducir su matrícula, nombre y carrera antes de empezar a generar su carga académica.
- Obtención y creación de un catálogo de materias para que así el usuario pueda escoger de entre **n** números de grupos.
- Con las materias que el usuario elija se verificará que haya registro existente, de esta forma se reduce el tiempo y se carga menos al servidor. Si no hay un registro existente se hace una consulta y se obtiene la información de todos los grupos que hay en la BD con esos nombres de materias. Por cada movimiento que el usuario realiza se genera un historial almacenado en la base de datos.
- Con las materias capturadas se genera el árbol con todas las combinaciones posibles entre ellas.
- Después se puso un filtro, donde los grupos de la materia no contengan empalme, en caso de ser así, se considera ese grupo como aceptado y se le asigna un puntaje a este grupo de materias.
- Para obtener el puntaje de cada grupo de materias lo que se hace es verificar cada día de la semana las horas intermedias que existen entre materias.

- Se reordenan todas las materias, poniendo al principio aquellas materias de menor puntaje, que serían los posibles mejores horarios.
- Se guarda la salida en la base de conocimientos para después reutilizarla.
- **Avanzar hacia delante.**- Posicionarse en el siguiente registro de la respuesta obtenida, haciendo una verificación de horas y catedráticos.
- **Avanzar hacia atrás.**- Posicionarse en el registro anterior de la respuesta obtenida, haciendo una verificación de horas y catedráticos.
- **Inicio.**- Posicionarse al inicio de la respuesta obtenida haciendo una verificación de horas y catedráticos.
- **Final.**- Posicionarse al final de la respuesta obtenida haciendo una verificación de horas y catedráticos.
- **Quitar hora**, oculta al usuario los grupos de materias que contengan la hora eliminada y se posiciona al inicio.
- **Agregar hora**, muestra al usuario los grupos de materias que contengan la hora agregada y se posiciona al inicio.
- **Empalme.**- Verifica que el grupo de materias de la posición actual no contenga horas de las eliminadas y regresa el resultado.
- **Quitar catedrático**, oculta al usuario los grupos de materias que contengan los catedráticos eliminados y se posiciona al inicio.
- **Agregar catedrático**, muestra al usuario los grupos de materias que contengan los catedráticos agregados y se posiciona al inicio.
- Verificar cada materia que no contenga alguno de los catedráticos eliminados y regresa el resultado.
- Seleccionar un grupo de horas predefinido.
- Graficar Horario de la posición actual del grupo de materias en formato de horas.
- Graficar Horario de la posición actual del grupo de materias en formato completo.
- Guardar en el registro del alumno la selección final de horario.

3.5 Diagramas de Caso de Uso

Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre este. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y el programa. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo. (SENN 1996)

3.5.1 Generación de carga

En la Figura 3-6, podemos observar el diagrama de cómo son tratados los datos por el sistema. En este caso el usuario al entrar al sistema genera una lista, la cual contiene las materias a cursar este semestre para seleccionar las materias que puede tomar en este periodo.

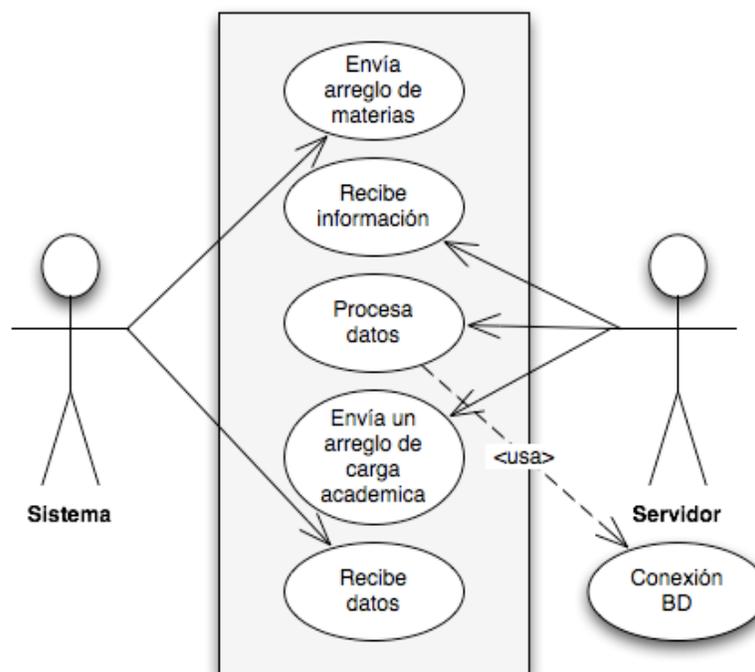


Figura No. 3-6 Generación de carga

3.5.2 Entrada al Sistema

En la Figura 3-7 podemos observar el esquema de la manera en que el usuario entra al sistema. Para esto primero tiene que introducir sus datos y enviarlos al servidor para que éste los procese y valide si es correcta la información.

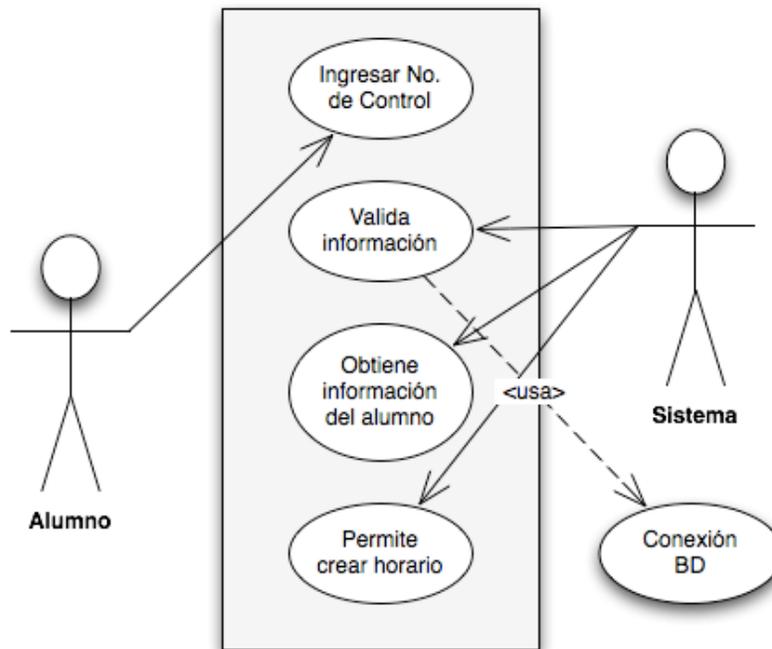


Figura 3-7 Entrada al sistema

3.5.3 Generar Horario

En la Figura 3-8, se muestra el proceso que se debe de seguir para que el usuario genere su horario. Este envía la información de las materias, a cursar este semestre, al servidor. Este último la procesa y cuando termina de generar la lista de horarios la envía de regreso para mostrarle al estudiante el resultado y permitirle personalizarlo para su posterior impresión.

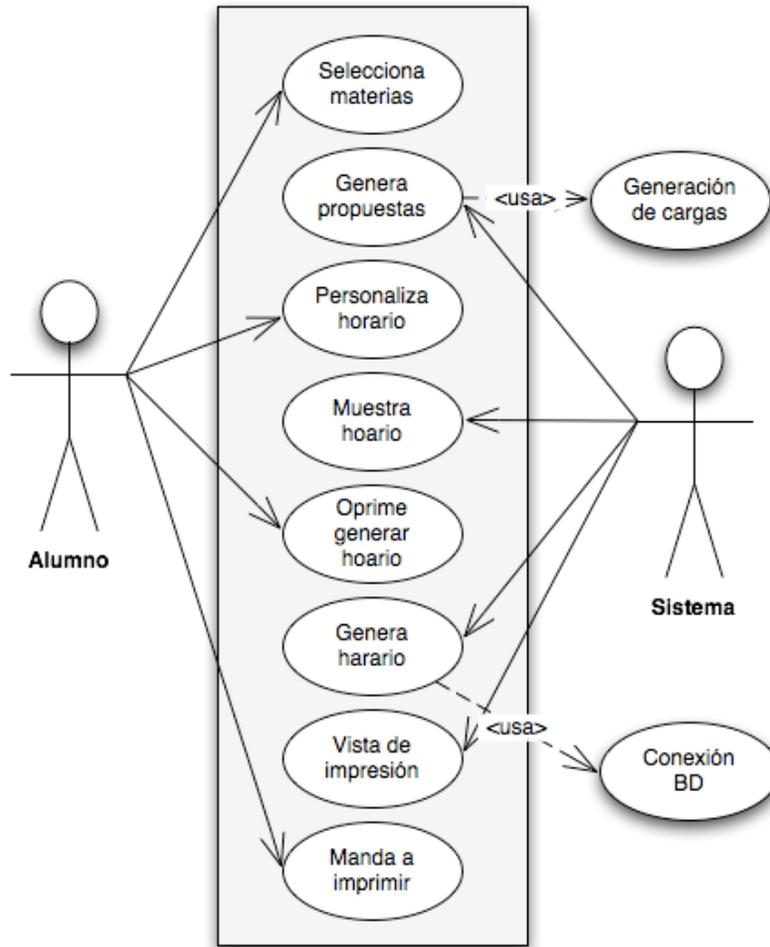


Figura No. 3-8 Generar Horario

3.5.4 Conexión BD

En la Figura 3-9 se muestra el funcionamiento del sistema para acceder a los datos personales del usuario para después mostrárselos y éste pueda modificarlos.

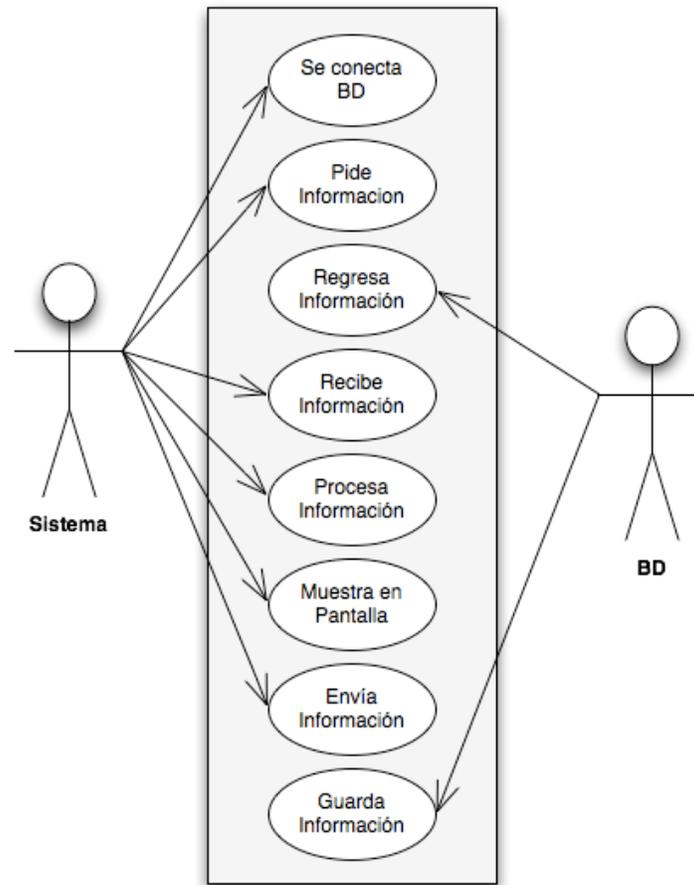


Figura No. 3-9 Conexión BD

3.6 Diagramas de Estados

En este apartado se mostrarán los diferentes estados en los que el usuario puede llegar a encontrarse dentro del sistema.

La Figura 3-10 muestra la manera en la cual el usuario entra al sistema. Para esto primero tendrá que acceder a la ventana de inicio de sesión, para posteriormente introducir su número de control, su contraseña; en caso de no estar registrado se mostrará un mensaje de error, de la misma manera mostrará esto si es que algún dato está mal. Una vez que los datos son válidos, el usuario podrá acceder al sistema.

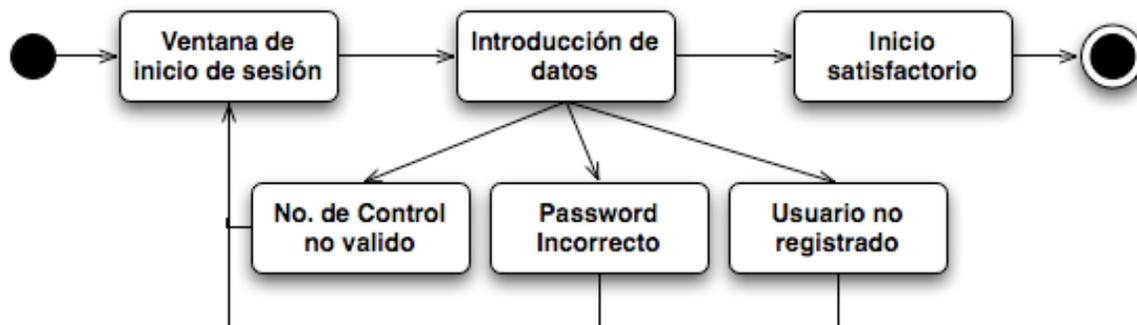


Figura 3-10 Inicio de sesión

El proceso de selección de materias (figura 3-11) inicia en cuanto el usuario accede al sistema y da clic en la opción a inscribirse. Si el alumno no tiene materias por cursar el alumno no podrá inscribirse, en caso contrario se le mostraran las recomendaciones que el sistema le proporciona y éste puede modificar siempre y cuando cumpla con los requerimientos de número de créditos mínimo. Ya que el usuario seleccionó las materias a cursar sólo resta aceptar la carga y se redirecciona a la ventana principal de selección de la misma.



Figura 3-11 Proceso de selección de materias

La Figura 3-12 muestra el proceso de selección de horario, éste empieza en cuanto el usuario acepta las materias que quiere cursar durante el semestre. En esta parte el alumno elegirá de una varias opciones, el horario que más se acople a sus necesidades, una vez que lo haya aceptado pasará a la vista de impresión, ya que el estudiante esté de acuerdo con el horario elegido aceptará la carga y regresará a la ventana principal del sistema.

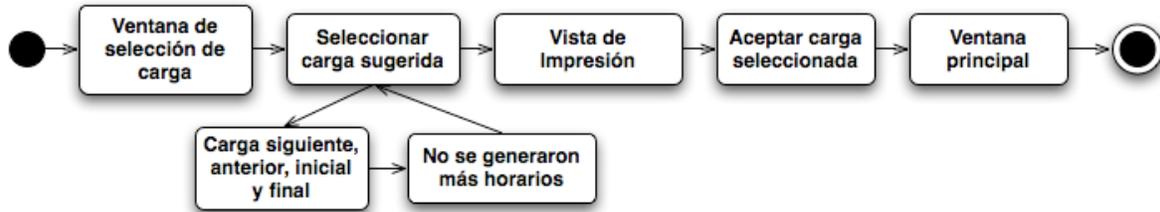


Figura No. 3-12 Proceso de selección e impresión de carga

En la Figura 3-13 se muestra el proceso en el cual el alumno podrá solicitar una guía para entender algún detalle del programa. Para esto tendrá que dar clic en el *link* “Ayuda”.

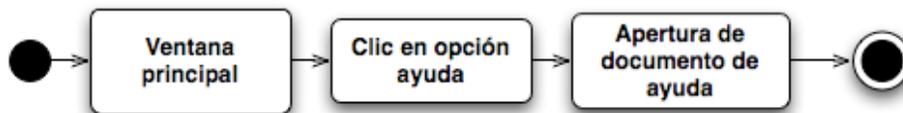


Figura No. 3-13 Proceso de ayuda

Para saber datos acerca del programa el usuario deberá de dar clic en el link “Acerca de Sistema Tutor”. El proceso se muestra en la Figura 3-14.

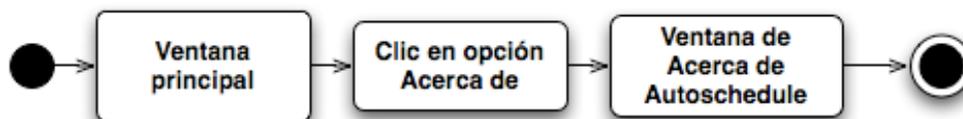


Figura 3-14 Proceso de Acerca de

La Figura 3-15 muestra el proceso de salida. Para realizar esto el usuario tendrá que dar clic en el texto que dice “Salir” ubicado en la parte izquierda de la ventana.



Figura 3-15 Proceso de Salir

3.7 Requerimientos para el Usuario

El sistema requiere de algunos aspectos que habrá que tomarse en cuenta antes de ejecutarlo, estos aspectos son mínimos y hoy en día la mayoría de computadoras cuenta con ellos.

3.7.1 Windows

Requisitos mínimos del sistema:

- Procesador a 233 Mhz
- 64 MB de RAM
- 50 MB de espacio libre en disco
- Microsoft Windows 98

Se recomienda al usuario:

- Procesador a 500 MHz
- 256 MB de RAM
- 100 MB de espacio libre en disco
- Microsoft Windows XP

3.7.2 Mac

Requisitos mínimos del sistema:

- Procesador PowerPC G3
- 128 MB de RAM
- 75 MB de espacio libre en disco
- Mac OS X 10.2.x

Se recomienda al usuario:

- Procesador PowerPC G4 ó Intel
- 512 MB de RAM
- 150 MB de espacio libre en disco

3.7.3 Linux

Requisitos mínimos del sistema:

- Procesador a 233 Mhz
- 64 MB de RAM
- 50 MB de espacio libre en disco
- Kernel Linux 2.2.14 (con glibc 2.3.2, XFree86-3.3.6, gtk+2.0, fontconfig/xft y libstdc++5)

Requisitos mínimos del sistema:

- Procesador a 500 Mhz
- 256 MB de RAM
- 100 MB de espacio libre en disco
- Una distribución de Linux actual

Las especificaciones antes mencionadas son las básicas para que el sistema pueda ser aprovechado por los usuarios. Con menos de esto se puede ocupar, pero el desempeño no sería el mismo, ya que aunque el navegador que se usa como base no requiere mucho procesamiento, al usar una configuración de *hardware* o *Software* podría tener algunas fallas. Esos requerimientos son los que pide *Mozilla Firefox* para que navegador pueda funcionar correctamente en los diferentes tipos de Sistemas Operativos.

En cuanto al uso del Sistema Operativo no hay problema, ya que el ser una aplicación *Web* y el estar desarrollada principalmente hacia el navegador *Mozilla Firefox* tiene sus ventajas, ya que éste programa está disponible para todas las plataformas.

3.8 Requerimientos para el desarrollo

Para su desarrollo se requieren dos equipos con buenas prestaciones.

3.8.1 Requerimientos de software

- Sistema Operativo Debian Etch 4 y Microsoft Windows XP SP2
- Aptana Studio 1.0 ó superior (para la versión Linux).
- Adobe Dreamweaver CS3 ó superior (para la versión Windows).
- Apache 2.0, Mysql 5.0 y PHP 5.0 ó superiores (para la versión Linux).
- WampServer 2.0 ó superior con PHPmyadmin (para la versión Windows).

3.8.2 Requerimientos de hardware

- Procesador Intel 1.8 Ghz ó superior
- Memoria RAM 512 MB ó superior
- Disco Duro de 20 GB ó superior

Todas estas especificaciones son las que requieren los programas antes mencionados para poder trabajar fluidamente, ya que con menos de esto, es posible que se pueda trabajar pero el desempeño no sería el mismo.

Al ser ésta una aplicación *Web*, no es necesario tener una paquetería especial para el desarrollo del sistema, ya que en cualquier plataforma se encuentran las herramientas necesarias, tanto libres como de paga, para la creación de un sistema igual.

Las ventajas de contar con una paquetería comprada es que van a tener la ayuda del soporte técnico y cualquier cosa pueden consultar con ellos. La ventaja de ocupar una libre es que en todos lados encuentras ejemplos de código que te

puede servir. Se recomienda la versión libre ya que no hay que pagar licencias y por qué en cualquier sitio de *Internet* existe información y códigos que pueden servir de ayuda para la realización de este sistema.

4 Diseño

4.1 Introducción

En este capítulo se hablará de cómo está compuesto el sistema, presentando su diagrama general que describe los componentes principales del sistema (Sección 4.3), se muestra el diseño de la base de datos presentando el diagrama Entidad-Relación que expresa las entidades relevantes para el sistema de información así como sus inter-relaciones y propiedades (Sección 4.4.1), también se presenta el diccionario de datos que describe las características lógicas de los datos que se utilizaron en el sistema, incluyendo nombre, descripción, alias, contenido y organización (Sección 4.4.2).

Posteriormente se presentarán los diagramas de flujo del sistema que especifican los detalles algorítmicos de los procesos y constituyen la representación gráfica de un proceso multifactorial (Sección 4.5).

Por último se detallará con el uso de ventanas el diseño de la interfaz que nos muestra un panorama de cómo es el resultado final del programa (Sección 4.6).

4.2 Arquitectura del sistema

Para describir la arquitectura del sistema, se mostrarán los siguientes elementos:

- El modelo de diseño
- El modelo de despliegue, identificando los nodos y sus configuraciones de red, y la explicación correspondiente.

4.2.1 Modelo de diseño

Las capas mostradas en el diagrama y que se muestran a través de la Figura 4-1 son las siguientes:

- La **capa específica de la aplicación**, que contiene todos los paquetes que serán desarrollados para cubrir las funcionalidades expuestas por el sistema particular.
- La **capa general de la aplicación**, que contiene todos aquellos paquetes que pueden ser reutilizados dentro del sistema o que la empresa ya tiene en otros sistemas, de los cuales podemos hacer uso.
- La **capa intermedia**, es el software sobre el cual descansa el desarrollo de todo el sistema, también conocido como middleware. Contiene todas las facilidades proporcionadas por el lenguaje de desarrollo y los manejadores de base de datos.
- La **capa del software del sistema**, que contiene la infraestructura proporcionada por el sistema operativo y el soporte a las diversas tecnologías de comunicación. (Fowler y Scott 1999)

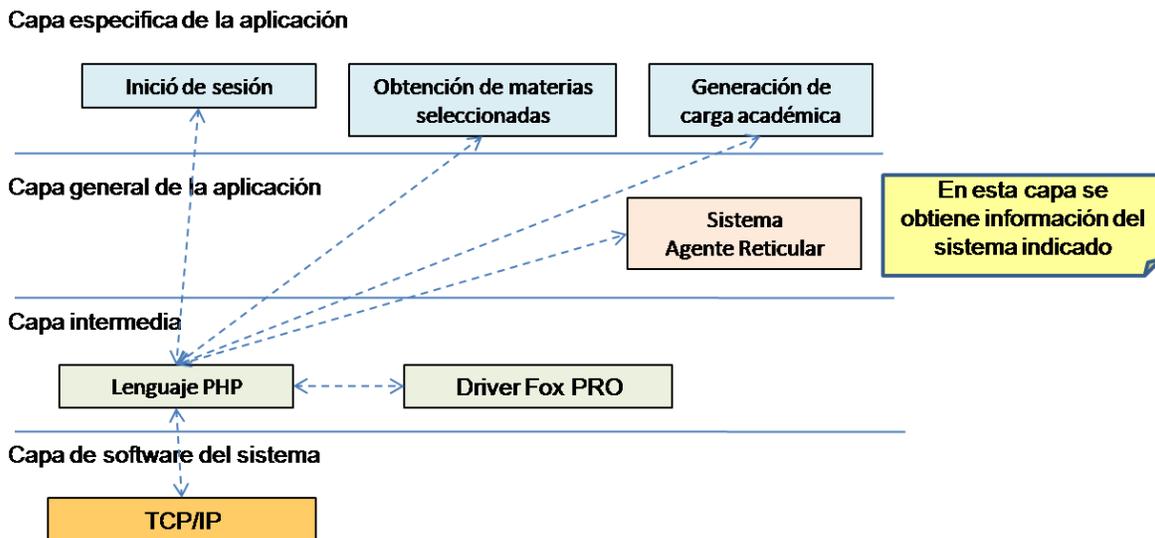


Figura 4-1 Modelo de diseño

Los módulos principales del sistema indicados en la capa específica de la aplicación son los siguientes:

- El módulo de inicio de sesión, que identifica al usuario al entrar al sistema.
- El módulo de obtención de materias seleccionadas, que se encarga de obtener las materias que seleccionó el alumno en el sistema “Agente reticular”.
- El módulo de generación de carga académica, se encarga de realizar la carga del alumno con los parámetros seleccionados.

En la capa general del sistema se muestra el sistema “Agente reticular”, en el cual los alumnos deben ingresar previamente para generar información necesaria para la aplicación.

La capa intermedia está conformada por el lenguaje de programación PHP y los drivers correspondientes para la conexión a la base de datos.

Por último, en la capa de software del sistema, se observa el protocolo sobre el cual descansan las comunicaciones de la red.

4.2.2 Modelo de despliegue

El modelo de despliegue que observamos en la Figura 4-2, muestra los nodos que corresponden a la configuración de la red, en que será ejecutado el sistema.

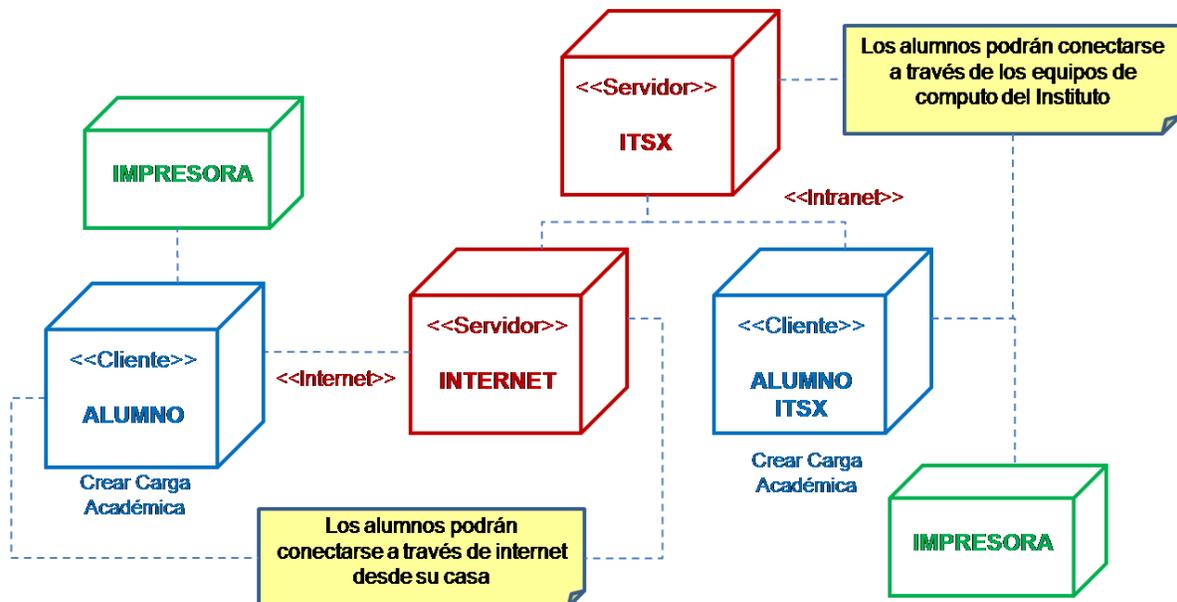


Figura No. 4-2 Modelo de despliegue

Los nodos involucrados son los siguientes:

- Servidor web, donde estará alojada la aplicación.
- Servidor de internet, que servirá para realizar el enlace vía internet.
- El Alumno, el cual podrá acceder a la aplicación ya sea por la intranet o internet.
- Las impresoras que estarán conectadas en los equipos de cómputo del instituto para realizar la impresión de la carga académica.
- La impresora con la que debe de contar el alumno para realizar la impresión de la carga académica desde su casa.

4.3 Diagrama General

En la Figura 4-3 se muestra el diagrama general del sistema, en el cual se presentan todos los componentes del mismo. En el diagrama se representa el servidor de procesos y de la interfaz en forma separada para ilustrarlo de una mejor manera pero teniendo en cuenta que en la práctica estos dos pueden ser uno solo.

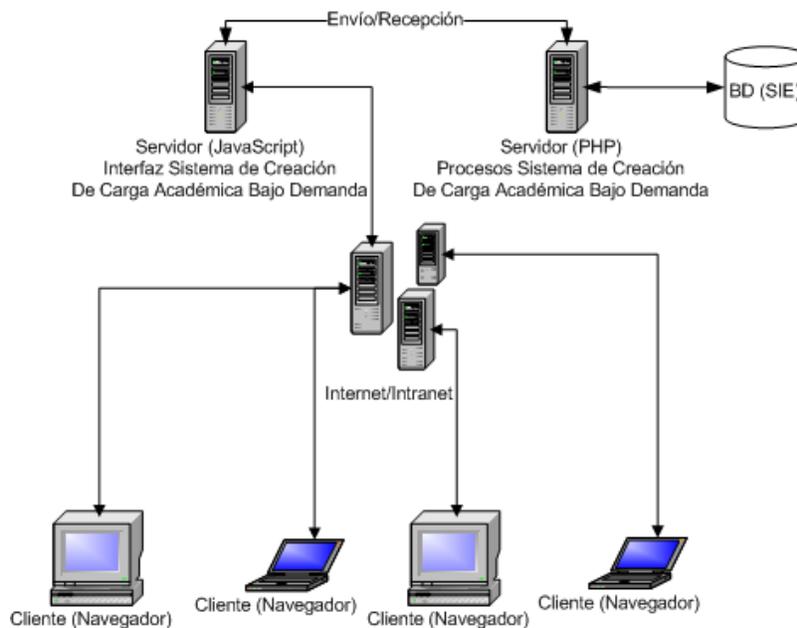


Figura No. 4-3 Diagrama General del Sistema

4.4 Diseño de la Base de Datos

4.4.1 Diagrama Entidad-Relación

En la **Figura 4-4** se muestran las relaciones existentes en la base de datos que utiliza para este sistema.

La tabla “**dalumn**” se ocupa para guardar los datos personales de cada alumno y se relaciona con “**catálogo**”, aquí se va a almacenar el Número de control del estudiante y el nombre de las materias que va a cursar este semestre así mismo “**catálogo**” se relaciona con “**horario**”, ésta contiene los detalles de cada grupo tales como: grupo, materia, créditos, entre otros. Por último tenemos la tabla “**conocimiento**”.

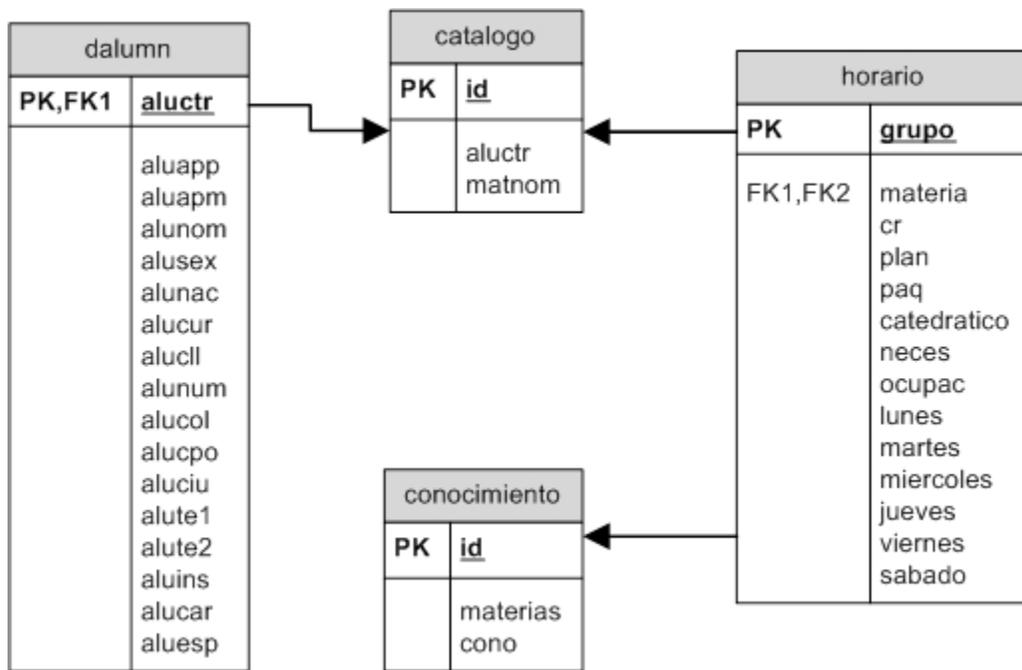


Figura 4-4 Diagrama Entidad-Relación.

4.4.2 Diccionario de datos

A continuación se muestra el diccionario de datos, entendiéndose por éste aquel que contiene las características lógicas de los datos que se van a utilizar en el sistema que estamos programando, incluyendo nombre, descripción, tipos, entre otras cosas.

La Tabla 4-1 muestra la estructura de la tabla “**catálogo**”, junto con los datos que acepta cada campo. En ella se almacenarán el Número de control y el nombre de la materia que el alumno deberá de cursar, en otras palabras aquí se guardan las materias recomendadas para cada alumno.

Tabla 4-1 Estructura de la tabla Catálogo

Campo	Tipo		Rango de Valores	Descripción
<u>id</u>	int(11)	No	0 – 9	Llave primaria
aluctr	varchar(9)	No	A – Z, a – z, 0 - 9	No. de control del alumno
matnom	varchar(25)	No	A – Z, a – z	Nombre de la materia

La tabla “**conocimiento**” se encuentra estructurada como se muestra en Tabla 4-2. Cuando un alumno genere su horario se creará una consulta, la cual contendrá un arreglo de materias, éstas se guardarán aquí para su posterior uso.

Tabla 4-2 Estructura de la tabla Conocimiento

Campo	Tipo	Nulo	Rango de Valores	Descripción
<u>id</u>	int(11)	No	0-9	Llave primaria
materias	varchar(1000)	No	A – Z, a – z	Arreglo de materias que pertenecen a la carga académica
cono	longtext	No	A – Z, a – z	Carga académica con toda la información completa para realizar la

En la tabla “**dalumn**” (Tabla 4-3), se guardarán los datos personales de cada alumno.

Tabla 4-3 Estructura de la tabla Dalumn

Campo	Tipo	Nulo	Rango de Valores	Descripción
ALUCTR	varchar(9)	Sí	A – Z, a – z, 0 – 9	No.de control
ALUAPP	varchar(25)	Sí	A – Z	Apellido paterno
ALUAPM	varchar(25)	Sí	A – Z	Apellido materno
ALUNOM	varchar(35)	Sí	A – Z	Nombre
ALUSEX	varchar(1)	Sí	A – Z	Genero

ALUNAC	Date	Sí	0 – 9/1 – 12/0 – 9	Fecha de nacimiento
ALUCUR	varchar(18)	Sí	A – Z	CURP
ALUCLL	varchar(30)	Sí	A – Z	Calle
ALUNUM	varchar(10)	Sí	A – Z	No. de calle
ALUCOL	varchar(20)	Sí	A – Z	Colonia
ALUCPO	int(11)	Sí	0 – 9	Código postal
ALUCIU	varchar(20)	Sí	A – Z, a – z	Ciudad
ALUTE1	varchar(13)	Sí	0 – 9	Teléfono 1
ALUTE2	varchar(13)	Sí	0 – 9	Teléfono 2
ALUINS	int(1)	No	0 – 9	Inscrito / No inscrito
ALUCAR	int(1)	Sí	0 – 9	Carrera
ALUESP	varchar(2)	Sí	A – Z, a – z	Especialidad

La tabla “**horario**”, la cual se encuentra estructurada como se muestra en Tabla 4-4 contendrá los datos de cada grupo abierto en el semestre en curso.

Tabla 4-4 Estructura de la tabla Horario

Campo	Tipo	Nulo	Rango de Valores	Descripción
<u>grupo</u>	varchar(20)	No	A – Z, a – z, 0 – 9	Identificador del grupo
materia	varchar(20)	No	A – Z, a – z	Nombre de la materia
cr	Float	No	0.0 – 9.9	Número de créditos
plan	int(11)	No	0 – 9	Plan de la materia
paq	varchar(5)	No	A – Z, a – z, 0 – 9	Paquete de la materia
catedratico	varchar(20)	No	A – Z, a – z	Catedrático
neces	varchar(10)	No	0 – 9	Necesidad de la materia
ocupac	varchar(20)	No	0 – 9	Ocupación actual
lunes	varchar(20)	No	0 – 9, -	Horario del día lunes
martes	varchar(20)	No	0 – 9, -	Horario del día martes
miercoles	varchar(20)	No	0 – 9, -	Horario del día miércoles
jueves	varchar(20)	No	0 – 9, -	Horario del día jueves
viernes	varchar(20)	No	0 – 9, -	Horario del día viernes
sabado	varchar(20)	No	0 – 9, -	Horario del día sábado

4.5 Diagramas de Flujo

Los diagramas de flujo son una herramienta gráfica que se emplea para describir y analizar el movimiento de datos de un sistema, incluyendo procesos, lugares para almacenar datos y retardos en el sistema. Los diagramas de flujo forman parte del Análisis Estructurado. (Kendall 2006)

El análisis de flujo de datos examina el empleo de los datos para llevar a cabo procesos específicos de la empresa, donde se estudian las actividades desde el punto de vista de los datos: dónde se originan, cómo se utilizan o cambian, hacia dónde se van, incluyendo las paradas a lo largo del camino que siguen desde su origen hasta su destino.

VENTAJAS

- Favorecen la comprensión del proceso a través de mostrarlo como un dibujo. Un buen diagrama de flujo reemplaza varias páginas de texto.
- Permiten identificar los problemas y las oportunidades de mejora del proceso. Se identifican los pasos redundantes, los cuellos de botella, y los puntos de decisión.
- Muestran las interfaces cliente-proveedor y las transacciones que en ellas se realizan, facilitando a los empleados el análisis de las mismas.

Son una excelente herramienta para capacitar a los nuevos empleados y también a los que desarrollan la tarea, cuando se realizan mejoras en el proceso.

4.5.1 Registro de Usuarios

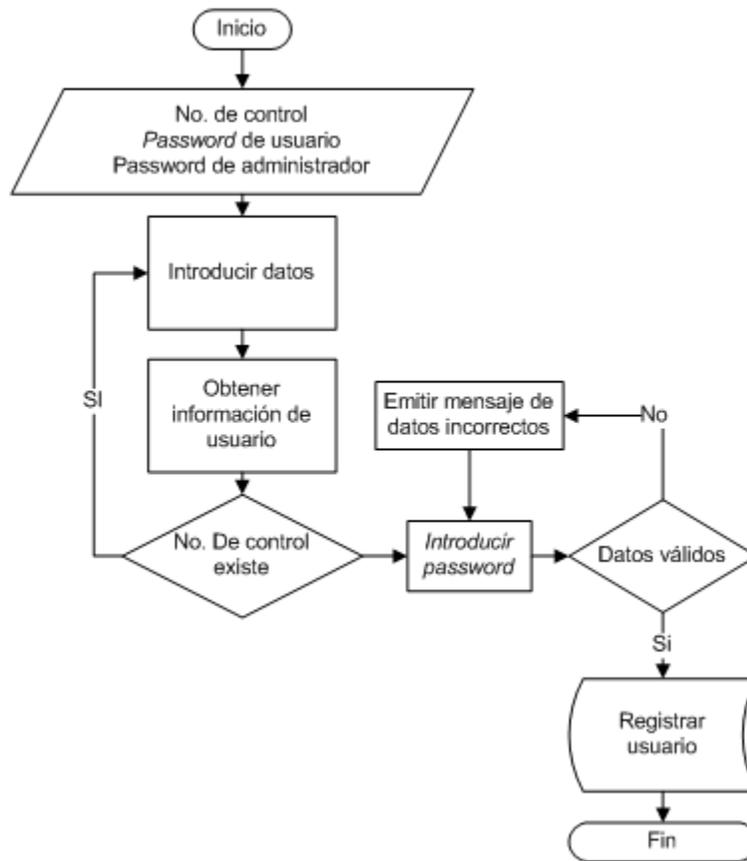


Figura 4-5 Diagrama de flujo para el Registro de Usuarios

El diagrama mostrado en la Figura 4-5, representa los pasos que el usuario tiene que realizar para registrarse en el sistema. Para empezar tiene que ingresar sus datos, la contraseña que va a ocupar en su cuenta y la contraseña del administrador para darse de alta satisfactoriamente.

4.5.2 Diagrama de flujo Creación de Carga

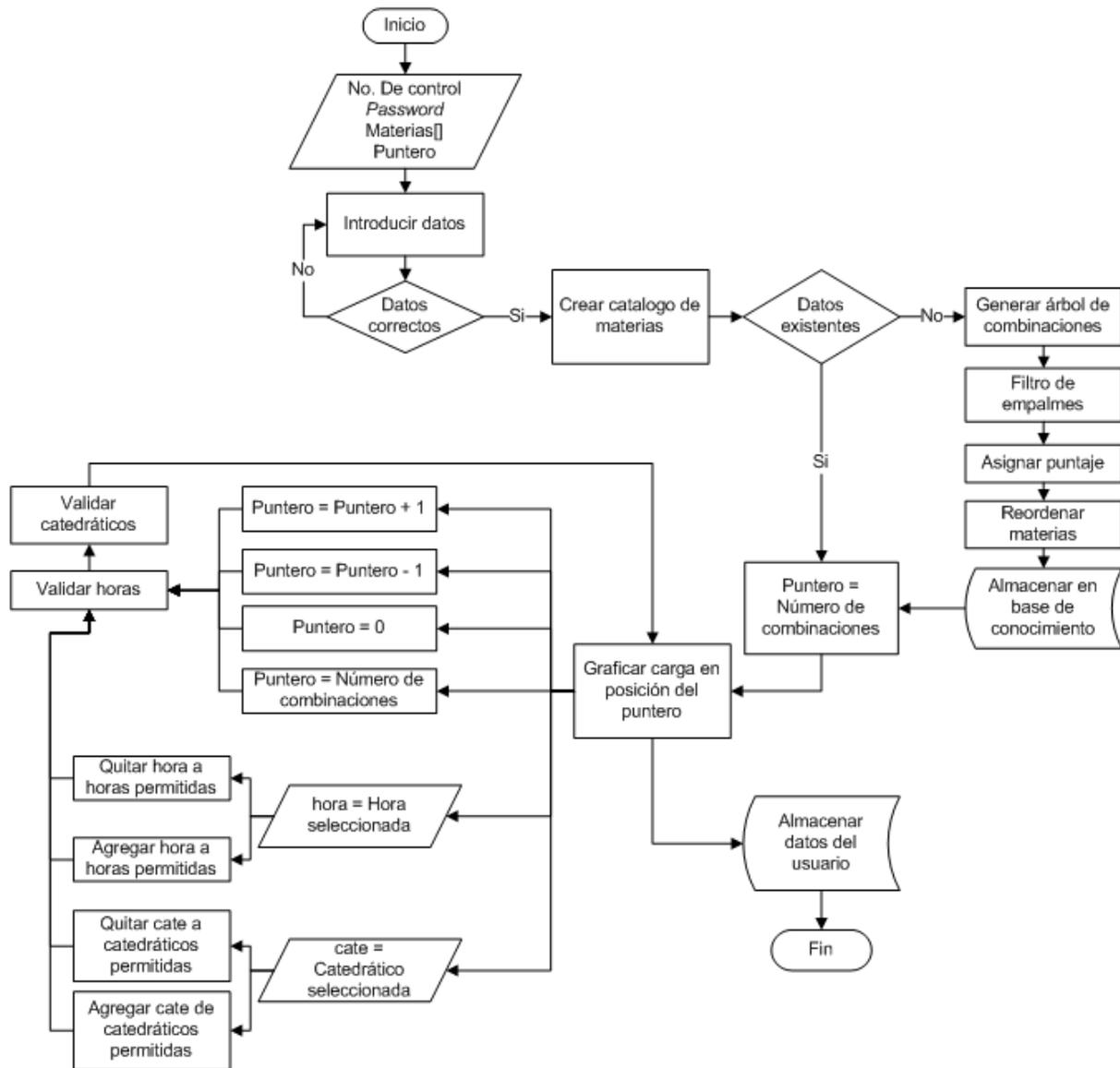


Figura No. 4-6 Diagrama de flujo de Creación de carga

En la Figura 4-6 se muestra la secuencia que se sigue para la creación de los horarios para los alumnos que ocupen este sistema.

4.6 Diseño de la Interfaz

En las figura 4-7 y figura 4-8 se muestra la página principal del generador de horarios y las opciones, respectivamente. Aquí el usuario tiene que dar clic en la opción de inscripción para poder agregar las materias que quiera cursar.



Figura No. 4-7 Página principal del sistema (lado izquierdo)



Figura 4-8 Página principal del sistema (lado derecho)

En la Figura 4-9 se muestra la página en la cual el alumno escoge las materias que quiere cursar este semestre de acuerdo a las que se abrieron en el periodo escolar. Una vez que el alumno ya no quiere agregar más asignaturas debe dar clic en el botón aceptar; de esta manera el sistema generará el árbol de toma de decisiones, aplicará el filtro de empalmes, asignará el puntaje, reordenará y almacenará las materias junto con las opciones de horario en la base de conocimientos. Todo esto se hace de manera transparente para el usuario.

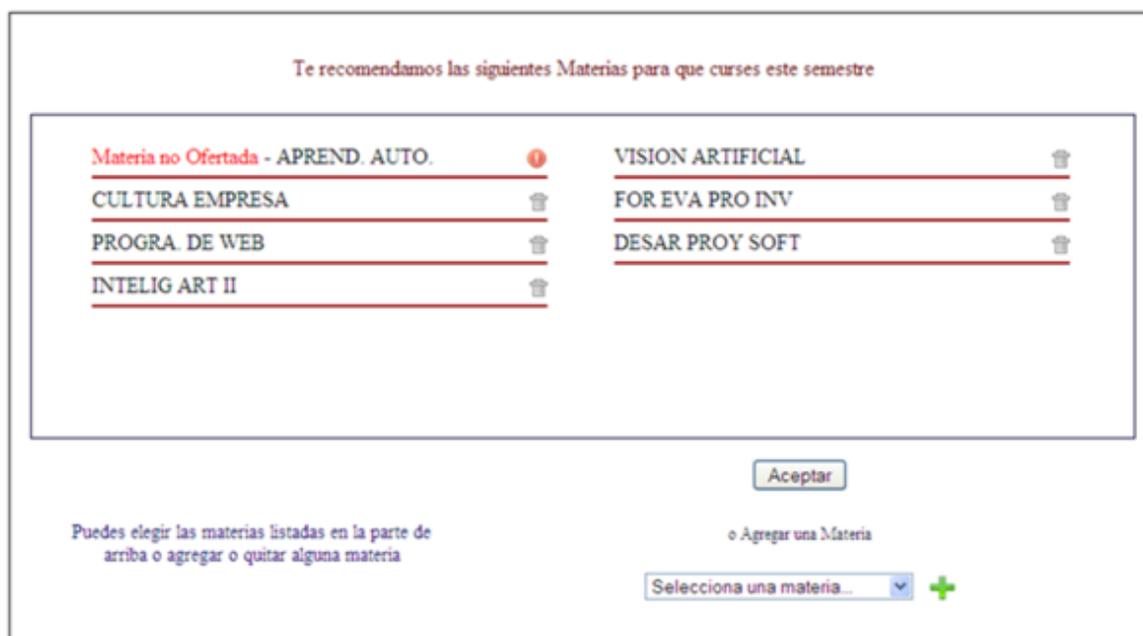


Figura 4-9 Materias a elegir

En la Figura 4-10 se presentan los resultados de ingresar las materias, así mismo, se activan las funciones:

Desplazar Siguiente, Desplazar Anterior, Desplazar Inicio, Desplazar Final, Aceptar Carga Seleccionada, representadas en la Figura 4-12 también se muestran las funciones Verificar Hora, Quitar Hora, Agregar Hora se activan para las dos (figura 4-12 y figura 4-13)

Agregar Catedrático, Verificar Catedrático, Quitar Catedrático, Agregar Materia,

Quitar Materia, las cuales se muestran en la figura 4-12.

Por la naturaleza del programa, la interfaz no va a sufrir cambio alguno y las demás pantallas son iguales a las mencionadas anteriormente

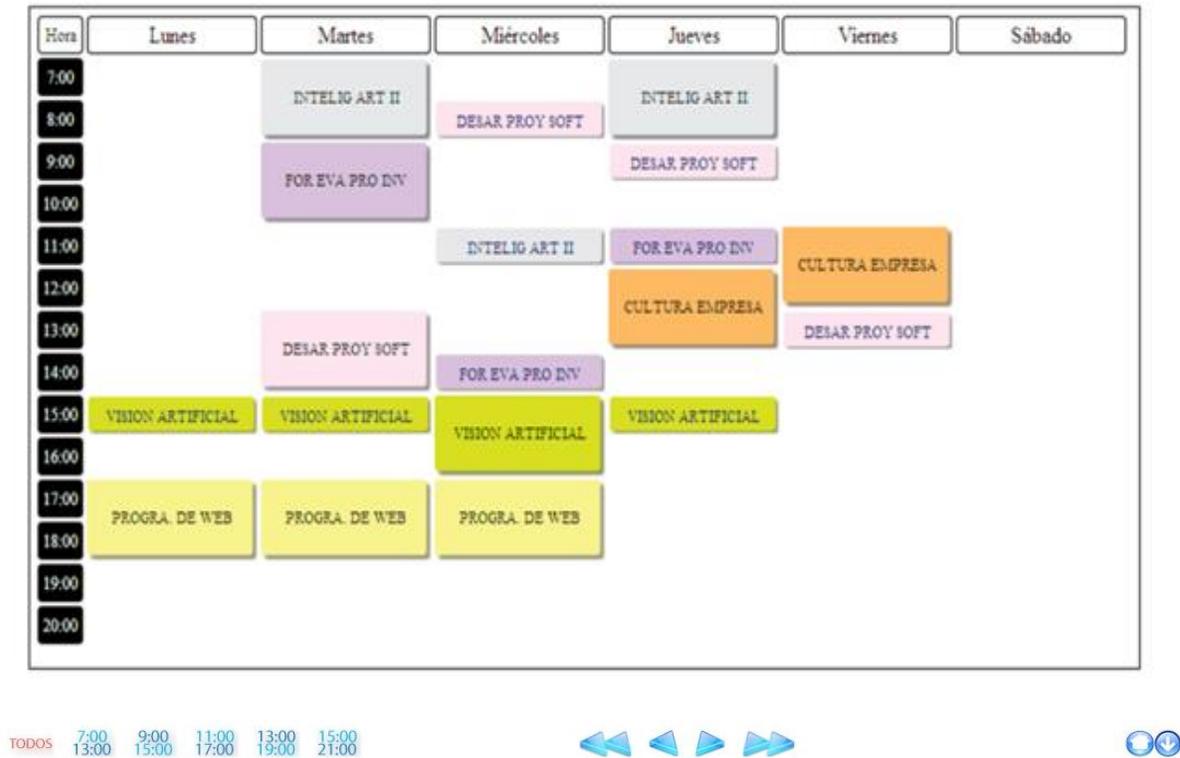


Figura 4-10 Desplegar horarios



Figura No. 4-11 Personalización materias y catedráticos

En la Figura 4-13 se representa la vista de impresión (aceptación de horario), en este momento el usuario solo tendrá que presionar el botón “Aceptar y Finalizar” para, de esta forma, poner a funcionar los métodos Aceptar Carga Seleccionada y Almacenar Carga.

Carga Académica

8/6/2008

NOMBRE:	FELIPE	NUM DE CONTROL:	04700125
CARRERA:	INGENIERIA EN SISTEMAS COMPUTACIONALES	SEMESTRE:	

Grupo	Materia	CR	RC	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
ESCC1048C	VISION ARTIFICIAL	8		15001600.EC01	15001600.EC01	15001700.EC03	15001600.EC03	.	.
ESCC1058C	INTELIG ART II	8		.	07000900.EC03	11001200.EC03	07000900.EC03	.	.
SCE04058A	CULTURA EMPRESA	6		.	.	.	12001400.EC05	11001300.EC05	.
SCE04118C	FOR EVA PRO INV	6		.	09001100.EC03	14001500.EC03	11001200.EC03	.	.
SCF04288D	PROGRA. DE WEB	8		17001900.EEAV	17001900.EC01	17001900.EC01	.	.	.

Firma del Alumno
Firma del Asesor
Firma de Control Escolar

Aceptar y Finalizar

Figura 4-12 Vista de impresión

5 Programación

5.1 Introducción

En este Capítulo se tratará todo lo relacionado con la codificación del sistema, detallando cada uno de los módulos que lo componen utilizando una porción de código y describiendo brevemente algunos aspectos que se tomaron en cuenta en este proceso (Sección 5.2).

Se tratarán temas como la generación de cargas académicas y los procesos que componen este módulo, como son: generar árbol, aplicar filtro de empalmes entre otras. Se mostrará brevemente como se desarrolló el apartado que muestra al usuario las opciones que podrá elegir y como poder realizar un recorrido entre estas, así como también las restricciones que podrá imponer el usuario para la realización de su carga académica. Finalmente se describirán los procesos que forman el módulo de aceptación de carga académica.

Por último se mostrará el Manual de Usuario donde se describirá paso a paso como se debe de utilizar el sistema, utilizando un lenguaje apto para cualquier usuario final, utilizando pequeñas ilustraciones del aspecto final (Sección 5.3).

5.2 Módulos del Sistema

El sistema se compone de 4 módulos, los cuales se componen a su vez de diversas funciones y se describen a continuación.

1. Generación de Arreglo de Cargas Académicas
 - a. Generar Árbol
 - b. Aplicar Filtro Empalmes
 - c. Asignar Puntaje
 - d. Reordenar
 - e. Verificar Existencia En BD

- f. Almacenar En BD
-
- 2. Selección de Carga Académica
 - a. Graficar Horario
 - b. Desplazar Siguiente
 - c. Desplazar Anterior
 - d. Desplazar Inicio
 - e. Desplazar Final
-
- 3. Restricciones del Usuario
 - a. Quitar Hora
 - b. Agregar Hora
 - c. Quitar Catedrático
 - d. Agregar Catedrático
 - e. Verificar Hora
 - f. Verificar Catedrático
 - g. Agregar Materia
 - h. Quitar Materia
-
- 4. Aceptación de Carga
 - a. Aceptar Carga Seleccionada
 - b. Almacenar Carga

1.- El primer módulo se conforma de 6 funciones (Generar Árbol, Aplicar Filtro Empalmes, Asignar Puntaje, Reordenar, Verificar Existencia En BD y Almacenar En BD) relacionadas con el algoritmo de búsqueda de carga académica. Estas funciones están agrupadas en un solo módulo porque son procesos transparentes para el usuario, se les dice así porque el estudiante no trata con ellas directamente y éste desconoce el funcionamiento de ellas, ya que éstas se llevan a cabo del lado del servidor y las salidas de unos son las entradas de otros.

2.- El segundo módulo contiene todas las funciones relacionadas al despliegue en pantalla de las propuestas de cargas (Graficar Horario, Desplazar Siguiente, Desplazar Anterior, Desplazar Inicio y Desplazar Final). Éstas son las más recurridas por el usuario, ya que proporcionan una forma de desplazarse por las diferentes cargas sugeridas de una forma totalmente gráfica y agradable al usuario. Todas ellas se relacionan completamente porque procesan la misma información.

3.- La relación entre las funciones que contiene el tercer módulo (Quitar Hora, Agregar Hora, Quitar Catedrático, Agregar Catedrático, Verificar Hora, Verificar Catedrático, Agregar Materia y Quitar Materia) es todo lo referente a las restricciones del usuario, por ejemplo, impedir que el sistema genere una carga que contenga una hora determinada o un catedrático. Así también poder agregar o quitar materias para que éste tenga un total control de lo que desea.

4.- El último módulo se conforma de las funciones de aceptación (Aceptar Carga Seleccionada y Almacenar Carga), ya que éstas se encargan de verificar los datos seleccionados por el usuario y guardar en la base de datos en el apartado referente a esta persona en particular, la carga académica.

5.2.1 Generación de Arreglo de Cargas Académicas

El siguiente código realiza un proceso de forma recursiva para la creación del árbol de combinaciones de cargas académicas Código 5-1.

```

function generaArbol( $i1t , $i2t , $arregloAyuda ){
    $i1 = $i1t;
    $i2 = $i2t;
    global $value,$arretot;
    for( $ij = 0; $ij < sizeof( $arregloAyuda ) ; $ij++ ){
        $arregloAyuda2[$ij] = $arregloAyuda[$ij];
    }
    $arregloAyuda2[$i1] = $value{$i1}{$i2};

    if( empalme( $arregloAyuda2 )){
        if( $i1 == ( sizeof( $value ) - 1 ) ){
            $arretot{sizeof($arretot)}{'grupos'} = $arregloAyuda2;
            $arretot{sizeof($arretot) - 1}{'puntaje'} = getValorTotal($arregloAyuda2);
        }
        else{
            for( $j = 0 ; $j < sizeof( $value{$i1 + 1} ) ; $j++){
                generaArbol( $i1+1 , $j , $arregloAyuda2 );
            }
        }
    }
}

```

Código 5-1 Generar Árbol de Combinaciones

Para aprobar una combinación es necesario realizar una verificación de cada materia para que no tenga un horario igual. Código 5-2.

```

//Verifica si hay un empalme
function empalme($arreglo){
    for($i = 0; $i < sizeof($arreglo);$i++){
        if(!verifica($arreglo,$i,$i+1,sizeof($arreglo))){
            return false;
        }
    }
    return true;
}
//Verifica grupo indicado con los demas de la derecha
function verifica($arreglo,$var, $varmas,$ultimo){
    if($varmas >= $ultimo ){
        return true;
    }
    for($i = 0; $i < 6; $i++){
        if(!checa($i,$arreglo[$var],$arreglo[$varmas])){
            return false;
        }
    }
    return verifica($arreglo,$var, $varmas + 1,$ultimo);
}

```

Código 5-2 Verificación de Empalme

Si esta combinación es aprobada por la función de empalmes se le asigna un puntaje dependiendo el número de horas entre cada materia. Código 5-3.

```

function getValor( $claves, $dia )
{
    $link = Conectarse();
    $dias = array( "lunes", "martes", "miercoles", "jueves", "viernes", "sabado" );
    $consulta = "select $dias[$dia] from horario".creaConsulta( $claves )." ORDER BY $dias[$dia] ASC " ;
    $result = mysql_query( $consulta, $link );

    $i = 0;

    while( $row = mysql_fetch_array( $result ) )
    {
        if( strlen($row[0]) >=8 )
        {
            $hora_ini[$i]= substr( $row[0], 0, 4 );
            $hora_fin[$i] = substr( $row[0], 4, 4 );
            $i++;
        }
    }
    $valor = 0;
    for($i = 1; $i < sizeof($hora_ini);$i++){
        $valor = $valor + $hora_ini[$i] - $hora_fin[$i-1];
    }
    return $valor/100;
}

```

Código 5-3 Asignar Puntaje a la Combinación

Una vez terminado de realizar las verificaciones de empalmes y asignar el puntaje de cada combinación se reordena el arreglo de combinaciones de forma que tengamos las de menor puntaje al principio (menor puntaje significa menos horas intermedias).Código 5-4.

```

function mejorHorario()
{
    global $arretot;

    for( $i = 0; $i < sizeof( $arretot ); $i++ )
    {
        for( $j = $i+1; $j < sizeof( $arretot ); $j++ )
        {
            if( $arretot{$j}{'puntaje'} < $arretot{$i}{'puntaje'} )
            {
                $temp = $arretot{ $i };
                $arretot{ $i } = $arretot{ $j };
                $arretot{ $j } = $temp;
            }
        }
    }
}

```

Código 5-4 Reordenación de combinaciones

5.2.2 Selección de Carga Académica

Posteriormente a la obtención del arreglo de combinaciones es necesario graficar las cargas académicas para que el alumno pueda elegir la que mas le convenga. Esto se realiza agregando la primera combinación de materias que conforman la carga académica al elemento de la página asignado para ello. (Código 5-5).

```
for(i = 0; i < horGraficar.grupos.length;i++){
    nombreMateria = horGraficar.grupos[i].materia;
    paq = horGraficar.grupos[i].paquete;
    cat = horGraficar.grupos[i].catedratico;
    respuesta.colorProfe(nombreMateria , cat );
    respuesta.colorPaq(nombreMateria , paq );
    for(j = 0; j < 6; j++){
        if(horGraficar.grupos[i][diasSemana[j]].length >= 8){
            tam = tama(horGraficar.grupos[i][diasSemana[j]].substring(0,8));
            top = valo(horGraficar.grupos[i][diasSemana[j]].substring(0,4));
            text+ '<div class="'+tam+'" style="top:'+ top +'px; left:'+leftSemana[j]+
                'px; background:url(img/'+tam+i+'.png)'>'+nombreMateria+'</div>';
        }
    }
}
```

Código 5-5 Graficar horario

Para el recorrido entre las diferentes combinaciones generadas se implementa unas funciones para avanzar al siguiente horario (Código 5-6), retroceder al horario anterior (Código 5-7), desplazarse al final y desplazarse al principio (Código 5-8) para que el alumno pueda observar y estudiar cada una.

```

function siguiente(){
    posicion = respuesta.indice;
    if( horarioGraficar = respuesta.siguiente() ){
        url = servidor + "limite.php?op=verificar&nocache=" + Math.random();
        gruposAc = new Array();
        for(a = 0; a < horarioGraficar.grupos.length; a++){
            gruposAc[a] = horarioGraficar.grupos[a].grupo;
        }
        new Ajax.Request(url, {
            method: 'post',
            parameters: 'grupos=' + Object.toJSON(gruposAc),
            onSuccess: function(transport){

                $res = transport.responseText;
                if( $res == 0){
                    siguiente();
                }
                else{
                    graficaHorario( horarioGraficar );
                }
            }
        });
    }
    return;
}
respuesta.indice = posicion;
noMasHorarios("No hay mas horarios");
}

```

Código 5-6 Graficar siguiente carga.

```

function anterior(){
    posicion = respuesta.indice;
    if( horarioGraficar = respuesta.anterior() ){
        url = servidor + "limite.php?op=verificar&nocache=" + Math.random();
        gruposAc = new Array();
        for(a = 0; a < horarioGraficar.grupos.length; a++){
            gruposAc[a] = horarioGraficar.grupos[a].grupo;
        }
        new Ajax.Request(url, {
            method: 'post',
            parameters: 'grupos=' + Object.toJSON(gruposAc),
            onSuccess: function(transport){
                $res = transport.responseText;
                if( $res == 0){
                    anterior();
                }
                else{
                    graficaHorario( horarioGraficar );
                }
            }
        });
    }
    return;
}
respuesta.indice = posicion;
noMasHorarios("No hay mas horarios");
}

```

Código 5-7 Graficar anterior carga

```
function final(){
    respuesta.fin();
    anterior();
}
function inicio(){
    respuesta.inicio();
    siguiente();
}
```

Código 5-8 Graficar Primer y Última Carga

5.2.3 Restricciones del Usuario

Para personalizar su carga académica el usuario puede imponer algunas restricciones como son: no mostrar cargas con alguna hora en específico (Código 5-9), no permitir que se muestren cargas con algún profesor (Código 5-10), etc.

```

function quitarHora(compar){
  if(respuesta.arreglo[compar]==""){
    respuesta.arreglo[compar]="2";
  }
  else{
    respuesta.arreglo[compar]="";
  }
  posicion = respuesta.indice;
  respuesta.inicio();
  if( horarioGraficar = respuesta.siguiete() ){
    ayuimg = true;
    for(i =0;i<respuesta.arreglo.length;i++){
      if(respuesta.arreglo[i] == "2"){
        ayuimg = false;
      }
    }
    if(ayuimg){
      $("todos").src = "img/todos-1.png";
    }
    else{
      $("todos").src = "img/todos.png";
    }
    $("7a13").src = "img/7a13.png";
    $("9a15").src = "img/9a15.png";
    $("11a17").src = "img/11a17.png";
    $("13a19").src = "img/13a19.png";
    $("15a21").src = "img/15a21.png";
    graficaHorario( horarioGraficar );
    return;
  }
  respuesta.indice = posicion;
  noMasHorarios("Al quitar esa hora no se genera ningun horario");
  if(respuesta.arreglo[compar]=="")
    respuesta.arreglo[compar]="2";
  else
    respuesta.arreglo[compar]="";
}

```

Código 5-9 Quitar Hora

```

function quitaProfe(delmat, delprof){
  respuesta.delProfe(delmat,delprof);
  posicion = respuesta.indice;
  respuesta.inicio();
  if( horarioGraficar = respuesta.siguiete() ){
    graficaHorario( horarioGraficar );
    return;
  }
  noMasHorarios("No puedes quitar este maestro");
  respuesta.indice = posicion;
  respuesta.delProfe(delmat,delprof);
}

```

Código 5-10 Quitar Profesor

5.2.4 Aceptación de carga

Cuando el alumno elija la carga que desee cursar se le muestra en una ventana el horario elegido para su revisión y aceptación final (Código 5-11).

```
for(j = 0; j < horarioActual.grupos.length ; j++ ){
    text1 = text1 + "<tr>";
    text1 = text1 + "<td id ='fgrupo'>" + horarioActual.grupos[j].grupo + "</td>";
    text1 = text1 + "<td id ='fmateria'>" + horarioActual.grupos[j].materia + "</td>";
    text1 = text1 + "<td id ='fcr' class='centrado'>" + horarioActual.grupos[j].creditos + "</td>";
    text1 = text1 + "<td id ='fcr' class='centrado'></td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].lunes + "</td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].martes + "</td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].miercoles + "</td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].jueves + "</td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].viernes + "</td>";
    text1 = text1 + "<td class ='fdia'>" + horarioActual.grupos[j].sabado + "</td>";
    text1 = text1 + "</tr>";
}
```

Código 5-11 Verificar y Aceptar Carga

5.3 Manual de Usuario

1. En la Ilustración 1, se muestra la página en donde debe ingresar el número de control y la contraseña correspondiente para acceder al sistema. En caso que todavía no esté registrado deberá dar clic en el *link* “**Registrarse**” y continuar con el paso 2, en caso contrario salte al paso 9.



Ilustración 1 muestra una pantalla de inicio de sesión con el título "Inicia Sesión". Incluye dos campos de entrada: "No. Control:" y "Password:". Debajo de los campos hay dos botones: "Salir" y "Iniciar". En la parte inferior del formulario, hay un enlace "Registrarse".

Ilustración 1 Inicio de Sesión

2. Una vez que dió clic en el *link* aparecerá una pantalla, Ilustración 2 en la que deberá escribir su número de control en el campo de texto que se encuentra del lado izquierdo del botón “**Buscar**”.



Ilustración 2 muestra una ventana de registro en Mozilla Firefox con el título "Registro - Mozilla Firefox". El contenido de la ventana es "REGISTRO DE ALUMNO". Incluye un campo de texto y un botón "Buscar". Debajo hay un campo de texto grande. Luego, hay campos para "Contraseña:" y "Repetir:". Debajo de eso, hay un campo para "Contraseña de Administrador:". En la parte inferior del formulario, hay un botón "Registrar".

Ilustración 2 Ventana de Registro

- Ya que dió clic en el botón “**Buscar**” aparecerán los datos del alumno en el recuadro de en medio y se activará el botón de “**Registrar**”. Si el número de control es correcto salte al paso 6.



The screenshot shows a web form titled "REGISTRO DE ALUMNO". At the top, there is a text input field containing "047o0057" and a blue "Buscar" button. Below this, a white box with a blue border displays the search results: "Nombre: ALMA ARELY RAMIREZ GONZALEZ" and "Carrera: INGENIERIA EN SISTEMAS COMPUTACIONALES". Underneath, there are three input fields: "Contraseña:" followed by an empty field, "Repetir:" followed by an empty field, and "Contraseña de Administrador:" followed by an empty field. At the bottom, there is a blue "Registrar" button.

Ilustración 3 Datos del Alumno

- Si el alumno ya se registró con anterioridad el mensaje cambiará al de la Ilustración 4.



The screenshot shows the same "REGISTRO DE ALUMNO" form. The search input field now contains "047o0053". The central white box with a blue border now displays a red information icon and the text "Usuario ya registrado". The "Registrar" button at the bottom is now greyed out.

Ilustración 4 Datos del Alumno

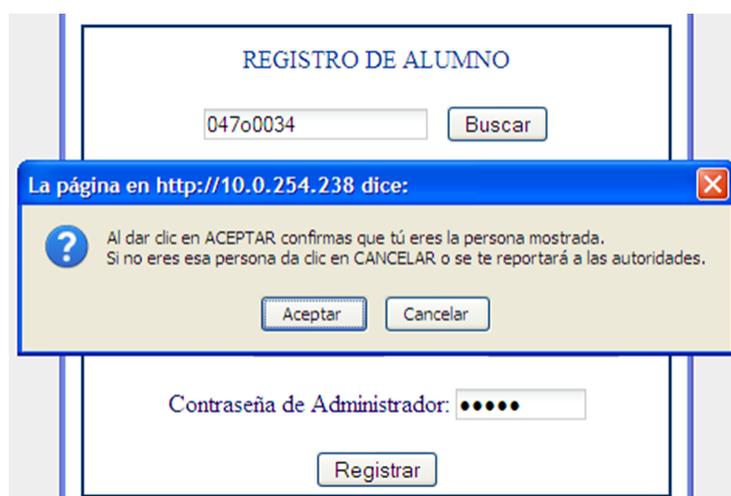
5. En caso que el número de control no esté incorrecto aparecerá un mensaje de error tal y como se muestra en la Ilustración 5. Cuando pase esto vuelva a escribir el número de control, búsquelo y pase al punto 6.



The screenshot shows a web form titled "REGISTRO DE ALUMNO". At the top, there is a text input field containing "047o005" and a "Buscar" button. Below this is a red-bordered box with a red exclamation mark icon and the text "No. Control Incorrecto". Underneath the error box are three input fields: "Contraseña:" (empty), "Repetir:" (empty), and "Contraseña de Administrador:" (empty). At the bottom of the form is a "Registrar" button.

Ilustración 5 Número de Control Incorrecto

6. Una vez que llenó los todos los registros deberá dar clic en el botón "Registrar" y aparecerá un cuadro de diálogo (Ilustración 6), de clic en el botón "Aceptar" y regresará a la ventana de inicio de sesión. Continúa paso 1.



The screenshot shows the same "REGISTRO DE ALUMNO" form as in Illustration 5, but with a confirmation dialog box overlaid. The dialog box has a blue title bar that reads "La página en http://10.0.254.238 dice:". The main content of the dialog box is white with a question mark icon and the text: "Al dar clic en ACEPTAR confirmas que tú eres la persona mostrada. Si no eres esa persona da clic en CANCELAR o se te reportará a las autoridades." Below the text are two buttons: "Aceptar" and "Cancelar". In the background, the form shows the "Contraseña de Administrador:" field filled with five black dots and the "Registrar" button.

Ilustración 6 Registro Exitoso

7. En caso que no introduzca la contraseña correctamente aparecerá el mensaje que se muestra en la Ilustración 7. Deberá volver a escribir la contraseña hasta que esté bien, cuando esté bien vaya al paso 1.



The screenshot shows a web form titled "REGISTRO DE ALUMNO". At the top, there is a text input field containing "047o0055" and a "Buscar" button. Below this is a red error message box with a downward arrow icon, containing the text: "¡Solo puedes usar +, -, *, ., números y letras en la contraseña y una longitud de 6 a 15 caracteres!". Underneath the message are three input fields: "Contraseña:" (empty), "Repetir:" (empty), and "Contraseña de Administrador:" (empty). At the bottom is a "Registrar" button.

Ilustración 7 Error en la Contraseña

8. Si la contraseña del administrador es incorrecta se mostrará un mensaje como el mostrado en la Ilustración 8. Y regrese al paso 1.



The screenshot shows the same "REGISTRO DE ALUMNO" form. The error message box now contains the text: "¡La contraseña de Administrador es Incorrecta!". The "Contraseña:" and "Repetir:" fields are now filled with six black dots. The "Contraseña de Administrador:" field is also filled with a single black dot. The "Registrar" button remains at the bottom.

Ilustración 8 Error en la contraseña de administrador

9. Una vez que ha accedido al sistema se mostrará la pantalla principal del sistema (Ilustración 9). Esta pantalla se muestra dividida en dos partes, del lado izquierdo se muestra el menú (Ilustración 10) y del lado derecho se muestra el área de trabajo, este variará de acuerdo a la opción seleccionada (Ilustración 11).



Ilustración 9 Pantalla principal



Ilustración 10 Menú de Opciones



Ilustración 11 Área de Trabajo

10. La segunda opción del menú es “**Inscripción**” al dar clic aquí el área de trabajo cambiará y mostrará su carga académica recomendada para este semestre, también podrá agregar o quitar materias dependiendo las que se ofertaron en el semestre. Ilustración 12



Ilustración 12 Menú de Inscripción

11. Una vez que se seleccionaron las materias deberá dar clic en el botón **“Aceptar”** y se mostrará la pantalla de selección de horario, Ilustración 13. En la Ilustración 14 es donde se mostrarán los horarios creados. El menú cambiará y se mostrarán las asignaturas y los maestros que impartirán cada una de ellas (Ilustración 15), del lado izquierdo de cada docente hay un botón el cuál permite mostrar horarios sin ese maestro y del lado derecho de cada materia se muestra un signo de menos, éste sirve para quitar la materia del horario. En la parte de abajo estarán los botones para desplazarse por los diferentes horarios así como para escoger intervalos de tiempo en los cuales se desee venir a la escuela y también para imprimir el horario o regresar a la ventana de selección de carga (Ilustración 16).

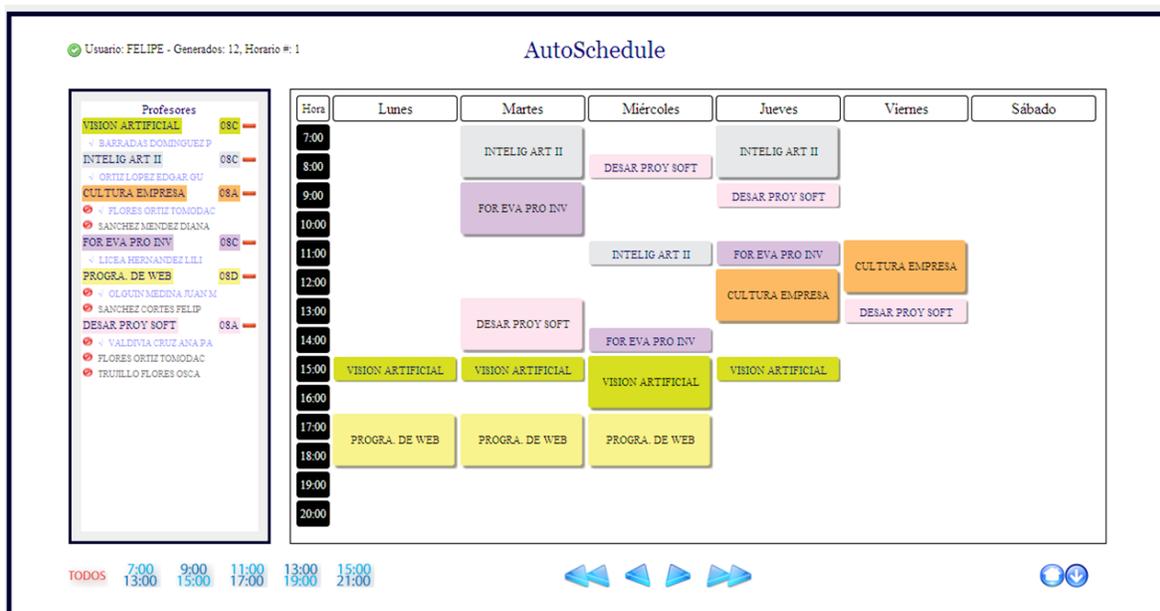


Ilustración 13 Página de selección de horario

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00		INTELG ART II		INTELG ART II		
8:00			DESAR PROJ SOFT			
9:00		FOR EVA PRO INV		DESAR PROJ SOFT		
10:00						
11:00			INTELG ART II	FOR EVA PRO INV	CULTURA EMPRESA	
12:00				CULTURA EMPRESA		
13:00		DESAR PROJ SOFT			DESAR PROJ SOFT	
14:00			FOR EVA PRO INV			
15:00	VISION ARTIFICIAL	VISION ARTIFICIAL	VISION ARTIFICIAL	VISION ARTIFICIAL		
16:00						
17:00	PROGRA. DE WEB	PROGRA. DE WEB	PROGRA. DE WEB			
18:00						
19:00						
20:00						

Ilustración 14 Selección de horarios

Profesores	
VISION ARTIFICIAL	08C
↳ BARRADAS DOMINGUEZ P	
INTELG ART II	08C
↳ ORTIZ LOPEZ EDGAR GU	
CULTURA EMPRESA	08A
↳ FLORES ORTIZ TOMODAC	
↳ SANCHEZ MENDEZ DIANA	
FOR EVA PRO INV	08C
↳ LICEA HERNANDEZ LILI	
PROGRA. DE WEB	08D
↳ OLGUIN MEDINA JUAN M	
↳ SANCHEZ CORTES FELIP	
DESAR PROJ SOFT	08A
↳ VALDIVIA CRUZ ANA PA	
↳ FLORES ORTIZ TOMODAC	
↳ TRUILLO FLORES OSCA	

Ilustración 15 Selección de materias y maestros



Ilustración 16 Selección de horas, movilidad entre registros, regreso e impresión

12. Cuando se presiona el botón de **“Impresión”** se mostrará una vista preliminar (Ilustración 17) de cómo va a quedar el horario. Si es el que se adapta a sus necesidades sólo resta dar clic en el botón **“Aceptar y Finalizar”** para que se guarde y regresar a la página de inicio.

Carga Académica

NOMBRE:	FELIPE	NUM DE CONTROL:	04*00125
CARRERA:	INGENIERIA EN SISTEMAS COMPUTACIONALES	SEMESTRE:	

Grupo	Materia	CR	RC	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
ESCC048C	VISION ARTIFICIAL	8		15001600.EC01	15001600.EC01	15001700.EC03	15001600.EC03	-	-
ESCC059C	INTELIG ART II	8		-	07000900.EC03	11001200.EC03	07000900.EC03	-	-
SCE04058A	CULTURA EMPRESA	6		-	-	-	12001400.EC05	11001300.EC05	-
SCE04118C	FOR.EVA.PRO.INV	6		-	09001100.EC03	14001500.EC03	11001200.EC03	-	-
SCF04288D	PROGRA. DE WEB	8		17001900.EEAV	17001900.EC01	17001900.EC01	-	-	-

Firma del Alumno
Firma del Asesor
Firma de Control Escolar

Ilustración 17 Vista de Impresión

13. La Ilustración 18 muestra el menú **“Acerca de AutoSchedule”**, aquí se desplegarán datos del sistema, tales como ¿quiénes lo crearon?, ¿en qué versión va?, o ¿para qué este sistema?

Acerca de AutoSchedule

AutoSchedule ®

Versión 0.5 Beta
 Copyrigh © 2008
 Javier Fuentes Mora
 Felipe Juárez Murillo

AutoSchedule es una herramienta que te ayudará en el proceso de reinscripción, modificación y consulta de tu carga Académica

Ilustración 18 Acerca de AutoSchedule

6 Pruebas

6.1 Introducción

La elaboración de pruebas permite la verificación del funcionamiento del sistema, en correspondencia con los requerimientos establecidos en el flujo de trabajo de análisis, una prueba exitosa es aquella que permite encontrar el mayor número de defectos posibles y posibilita su corrección.

6.2 Resultados de las pruebas

Para llevar a cabo la etapa de pruebas se tomó una muestra de 400 alumnos de la carrera de Sistemas Computacionales para la generación de su carga académica. Cabe mencionar que el tiempo promedio para finalizar este proceso sin la utilización del “Sistema Inteligente de Creación de Carga Académica Bajo Demanda” es de **30 minutos**.

El sistema se instaló en 5 equipos de cómputo del ITSX para la realización de las pruebas.

Para un mejor análisis del proceso de pruebas se dividió en 4 etapas quedando de la siguiente manera:

1. Inicio de sesión.
2. Selección de materias.
3. Selección de carga académica
4. Finalizar carga.

Para cada etapa se midió el tiempo y se analizó el proceso en busca de posibles defectos que sirvan posteriormente a una actualización del sistema.

6.2.1 Inicio de sesión

La siguiente gráfica describe el tiempo utilizado por los alumnos para esta etapa.

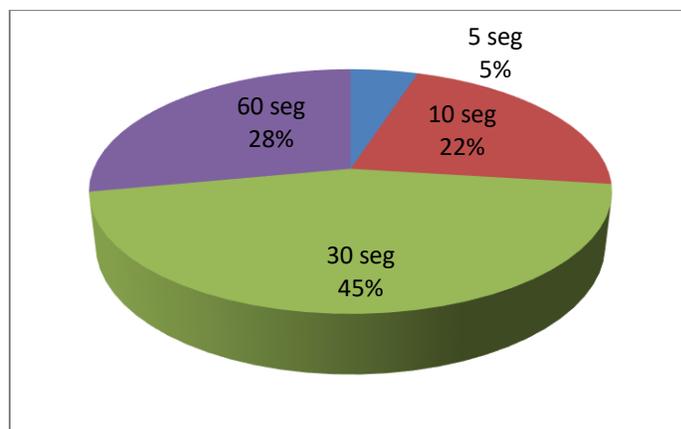


Figura No. 6-1 Inicio de Sesión

En esta gráfica podemos observar que esta etapa no representa un problema de tiempo en la realización de la carga académica de cada alumno ya que se realiza en un periodo de tiempo muy corto.

Este proceso tiene un promedio de **48 segundos**.

6.2.2 Selección de materias

La siguiente gráfica describe el tiempo utilizado por los alumnos para esta etapa.

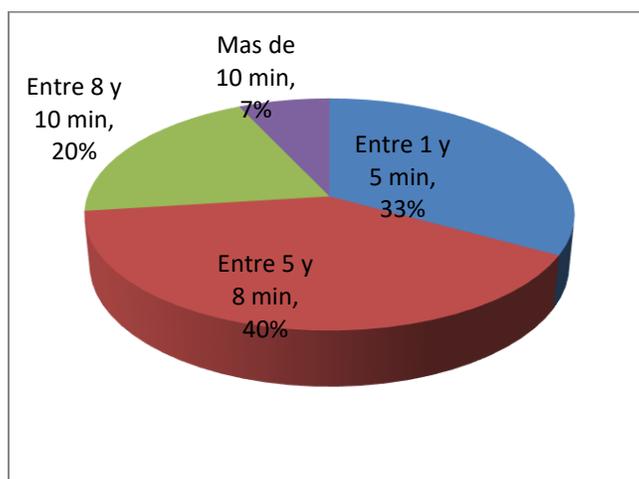


Figura 6-2 Selección de Materias

Durante este proceso se identificó que si los alumnos son regulares (no tienen materias en repite o especial) lo realizan en un periodo de tiempo más cortos que cuando no lo son, tardando en promedio de **4 minutos**.

6.2.3 Selección de carga académica.

La siguiente gráfica describe el tiempo utilizado por los alumnos para esta etapa.

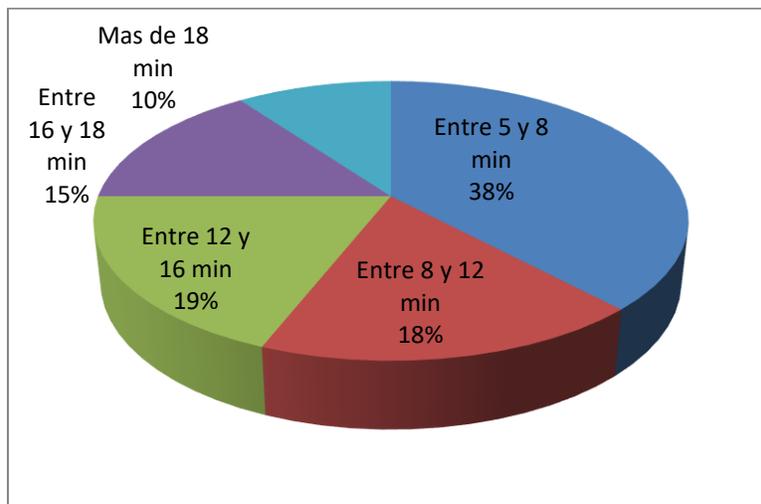


Figura No. 6-3 Selección de carga académica

Esta etapa es la más importante de todo el proceso de generación de carga académica ya que el alumno tiene que seleccionar su carga de las opciones que genera el sistema. El tiempo promedio es de **12 minutos**.

6.2.4 Finalizar carga

Por último, la etapa de finalización de carga académica no representó un problema de tiempo ya que el tiempo promedio para realizar este proceso es de **3 minutos**.

Los resultados se muestran en la siguiente gráfica.

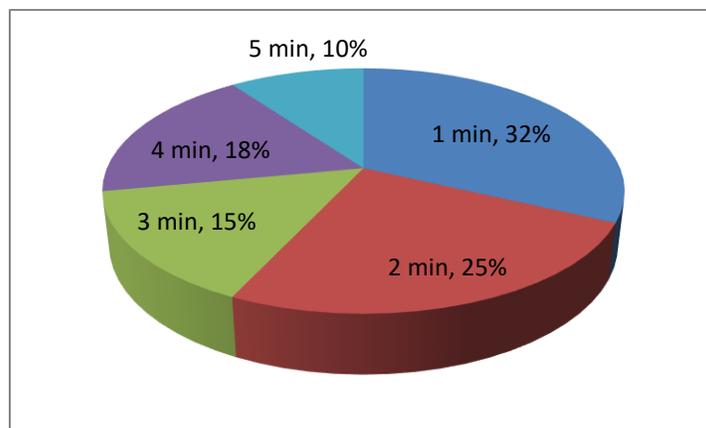


Figura 6-4 Finalizar Carga

6.3 Comprobación de la hipótesis

La hipótesis planteada es aceptada porque con la utilización del “Sistema Inteligente de Creación de Carga Académica Bajo Demanda” el alumno pudo realizar el proceso de creación de su carga académica con un ahorro de por lo menos el 60% de tiempo, produciendo la mayoría de las ocasiones una carga óptima para el alumno.

6.4 Conclusiones

La realización del sistema tutor para la conformación de horarios bajo demanda del estudiante en el Instituto Tecnológico Superior de Xalapa, ha permitido a la Institución reducir significativamente los tiempos del proceso de inscripción de los 5,000 alumnos con los que cuenta la institución, además el proceso requiere de un menor número de personas involucradas tanto docentes como personal administrativo que anteriormente asignaban de dos a tres semanas para llevar a cabo el proceso, lo que conllevaba a menos tiempo de preparación de las cátedras por parte de los docentes o en su defecto menos cursos de actualización para la mejora de la calidad educativa. En el mismo sentido con la implantación del sistema ha permitido la atención del 100 % por ciento de los alumnos en su proceso de asesoría reticular, lo que ha conllevado a durante la presente

administración a tener menores índices de reprobación y deserción escolar, así mismo elevar los niveles de eficiencia terminal del Instituto tal y como se muestra en los informes emitidos a la Junta de Gobierno por parte de la Dirección General que se anexan.

Cabe resaltar que gracias a la elaboración de proyectos como este y otros que existen en la institución (Programas de asesorías y tutorías) de carácter académico se ha logrado nivelar y controlar los avances reticulares de los alumnos teniendo a la fecha mejores resultados en la gestión pública.

No menos importante ha sido el resultado obtenido con el sistema en la reducción de costos de operación de la escuela, ya que al reducir los tiempos, disminuir el número de personas involucradas ha permitido que los ahorros realizados se inviertan en equipamiento de tecnologías de información hacia el interior del plantel.

Glosario

Adobe Dreamweaver.- Es una aplicación en forma de estudio pero con más parecido a un taller destinado para la construcción y edición de sitios y aplicaciones *Web* basados en estándares.

Algoritmo.- Es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

Apache.- Es un *software* (libre) servidor *HTTP* de código abierto para plataformas *Unix* (*BSD*, *GNU/Linux*, etc.), *Windows*, *Macintosh* y otras, que implementa el protocolo *HTTP/1.1* y la noción de sitio virtual.

Arquitectura Cliente-Servidor.- Consiste básicamente en que un programa -el cliente- que realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Asíncrono.- Hace referencia al suceso que no tiene lugar en total correspondencia temporal con otro suceso.

Avance académico.- Número de materias que ha cursado y acreditado en su estadía en el ITSX.

Base de Conocimientos.- Es un tipo especial de base de datos para la gestión del conocimiento. Provee los medios para la recolección, organización y recuperación computarizada de conocimiento.

Biblioteca GTK+.- Es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (*GUI*) para principalmente los entornos gráficos *GNOME*, *XFCE* y *ROX* de sistemas *UNIX*. *GTK+* es la abreviatura de *GIMP toolkit* (conjunto de rutinas para *GIMP*).

Biblioteca Qt.- Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

Carga Académica.- Número de materias que el alumno cursa durante un semestre, así como la descripción de cada una de ellas, su horario, grupo, maestro y número de créditos.

Codificación.- Es el proceso por el cual la información de una fuente es convertida en símbolos para ser comunicada. En otras palabras, es la aplicación de las reglas de un código.

Configuración.- Conjunto de datos que determina el valor de algunas variables de un programa o sistema de software, estas opciones generalmente son cargadas en su inicio y en algunos casos se deberá reiniciar para poder ver los cambios, ya que el programa no podrá cargarlos mientras se está ejecutando, si la configuración aún no ha sido definida por el usuario (personalizada), el programa o sistema cargara la configuración por defecto (predeterminada).

Demanda académica.- Número de alumnos que necesitan cierta(s) materia(s).

Disco duro.- Es un dispositivo de almacenamiento no volátil, que conserva la información aun con la pérdida de energía, emplea un sistema de grabación magnética

digital.

DOM.- El Document Object Model es esencialmente un modelo computacional a través del cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

Especial.- Materia que no acredito en la primera oportunidad ni en la segunda (repite).

Firefox.- Es un navegador de *Internet* libre y de código abierto descendiente de *Mozilla Application Suite*, desarrollado por la *Corporación Mozilla*, la *Fundación Mozilla* y un gran número de voluntarios externos.

fontconfig/xft.- Es una librería para la personalización y configuración de la tipografía.

Frames.- En inteligencia artificial, estructura de datos que contiene una descripción general de un objeto, que se deriva de conceptos básicos y de la experiencia.

Free Software Foundation.- Es una organización creada en Octubre de 1985 por *Richard Matthew Stallman* y otros entusiastas del *Software Libre* con el propósito de difundir este movimiento.

Gedit.- Es un completo editor de textos libre que se distribuye junto al gestor de escritorio *GNOME* para sistemas tipo *Unix*.

Giga Bytes (GB).- Es una unidad de medida informática cuyo símbolo es el *GB*, y puede equivalerse a 23^0 bytes o a 1^09 bytes, según el uso.

glibc 2.3.2.- Es la biblioteca estándar de C *GNU*. Se distribuye bajo los términos de la licencia *GNU LGPL*.

GNU Emacs.- Es parte del proyecto *GNU*, activamente desarrollado. Es la versión más popular de *Emacs*. El manual de *GNU Emacs* lo describe como "un editor extensible, personalizable, auto-documentado y de tiempo real". Es la versión más compatible y portada de las implementaciones de *Emacs*. La última versión, liberada el 5 de Septiembre de 2008, es la 22.3.

Hardware.- Corresponde a todas las partes físicas y tangibles de una computadora, sus componentes eléctricos, electrónicos, electromecánicos y mecánicos, contrariamente al soporte lógico intangible que es llamado *software*.

Ingeniería del Software.- Es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Intel.- Es una empresa multinacional que fabrica microprocesadores, circuitos integrados especializados tales como circuitos integrados auxiliares para placas base de computadora y otros dispositivos electrónicos.

Interfaz.- La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

Internet Explorer.- Generalmente abreviado *IE* es un navegador *web* producido por *Microsoft* para el sistema operativo *Windows* y más tarde para *Sun Solaris* y *Apple Macintosh*, estas dos últimas

discontinuas en el 2002 y 2006 respectivamente.

Internet.- Es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos *TCP/IP*, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Intranet.- Es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos *TCP/IP*, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

JAVA.- Es un lenguaje de programación orientado a objetos desarrollado por Sun *Microsystems* a principios de los años 90.

JSON.- Es un formato ligero para el intercambio de datos. *JSON* es un subconjunto de la notación literal de objetos de *JavaScript* que no requiere el uso de *XML*.

Kardex.- Compendio general de las calificaciones por alumno.

Kernel.- Es el núcleo y es la parte fundamental de un sistema operativo. La primera versión del programa apareció en 1992 y desde entonces *Helios Software Solutions* lo ha seguido desarrollando, de hecho, actualmente este *software* está en su quinta *major* versión (5.2).

Linux.- Es un término genérico para referirse a sistemas operativos similares a *Unix* basados en el núcleo de *Linux*.

Login.- Es el acto de establecimiento o confirmación de algo (o alguien) como

auténtico.

Mac OS X.- Es una línea de sistemas operativos computacionales desarrollado, comercializado y vendido por *Apple Inc.*, es la última versión de toda la línea de computadoras *Macintosh*.

Mega Bytes (MB).- Es una unidad de medida de cantidad de datos informáticos.

Mega Hertz (Mhz).- Se utiliza muy frecuentemente como unidad de medida de la frecuencia de trabajo de un dispositivo de hardware.

Microsoft FrontPage.- Fue un editor *WYSIWYG HTML* y una herramienta para administrar sitios *web* de *Microsoft* para la línea de sistemas operativos *Microsoft Windows*.

Microsoft Office.- Es una *suite* ofimática, compuesta básicamente por aplicaciones de procesamiento de textos, plantilla de cálculo y programa para presentaciones.

Modelo Entidad-Relación.- Herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

Modelo Relacional.- Para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Motor de inferencia.- El motor de inferencia es el "supervisor", un programa que está entre el usuario y la base de conocimientos, y que extrae conclusiones a partir de los

datos simbólicos que están almacenados en las bases de hechos y de conocimiento. Dependen en gran medida de la representación elegida.

Netscape.- Es un navegador *web* y el primer resultado comercial de la compañía *Netscape Communications*, creada por *Marc Andreessen*, uno de los autores de *Mosaic* cuando se encontraba en el *NCSA* (Centro Nacional de Aplicaciones para Supercomputadores) de la Universidad de *Illinois en Urbana Champaign*. *Netscape* fue el primer navegador comercial.

Normalización.- El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

ODBC *Open Database Connectivity*.- Es un estándar de acceso a Bases de Datos desarrollado por *Microsoft Corporation*, el objetivo de *ODBC* es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (*DBMS* por sus siglas en inglés) almacene los datos

Opera.- Es un navegador *web* y suite de *Internet* creado por la empresa noruega *Opera Software*. La aplicación es gratuita desde su versión 8.50, habiendo sido previamente *shareware* o *adware* y, antes de su versión 5.0, únicamente de pago.

Páginas *web*.- También conocida como página de *Internet*, es una fuente de información adaptada para la *World Wide Web (WWW)* y accesible mediante un navegador de *Internet* que normalmente forma parte de un sitio *web*.

PHP License.- Es la licencia bajo la cual el lenguaje de programación es liberado.

Plugin.- Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

PowerPC.- Es el nombre original de la arquitectura de computadoras de tipo *RISC*, fue desarrollada por *IBM*, *Motorola* y *Apple*.

Procesador.- Es un circuito integrado que contiene algunos o todos los elementos necesarios para conformar una (o más) "unidad central de procesamiento" *UCP*, también conocido como *CPU*.

Programa Compilado.- Es un programa escrito en un lenguaje "compilado", se traduce a través de un programa anexo llamado compilador que, a su vez, crea un nuevo archivo independiente que no necesita ningún otro programa para ejecutarse a sí mismo.

Programa interpretado.- Es un programa escrito en un lenguaje interpretado, requiere de un programa auxiliar (el intérprete), que traduce los comandos de los programas según sea necesario.

Random Access Memory (RAM).- Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

Repíte.- Materia que curso y no acreditó en el primer intento.

Reticula.- Manera en la cuál están distribuidas las materias de acuerdo a los conocimientos que debe llevar el alumno antes de tomar dicha asignatura.

Safari.- Es un navegador *web* desarrollado por *Apple Inc.*. Es el navegador incluido en *Mac OS X* y recientemente se han

desarrollado versiones ejecutables para *Microsoft Windows*.

Servidor.- Es una computadora que, formando parte de una red, provee servicios a otras denominadas clientes.

SGML.- Son las siglas de Standard Generalized Markup Language o "Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos.

Sistema de Integración Escolar (SIE).- *Software* propietario adquirido por el ITSX a otra Institución con el objetivo administrar y controlar la información escolar del plantel.

Sistema Manejador de Base de Datos.- El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Sistema Operativo.- Es un *software* de sistema, es decir, un conjunto de programas de computación destinado a muchas tareas entre las que destaca la administración eficaz de sus recursos.

Sistema Orientado a Objetos.- Un sistema de información Orientado a Objetos no es Base de datos + programas, sino que es un conjunto de objetos colaborativos donde los objetos persistente son guardados en una Base de Datos.

Sistema.- Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

Sistemas Distribuidos.- Se define como: una colección de computadores separados físicamente y conectados entre sí por una

red de comunicaciones distribuida.

Software.- La palabra software se refiere al equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Textpad.- Es un popular editor de texto para la familia de sistemas operativos *Microsoft Windows*.

The PHP Group.- Equipo de desarrollo que se encarga de mantener a diario actualizaciones disponibles para el lenguaje *PHP*.

Vim.- Es una versión mejorada del editor de texto *vi*, presente en todos los sistemas *UNIX*.

Wide Web Consortium (W3C).- El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce estándares para la World Wide Web.

Windows.- Es una familia de sistemas operativos desarrollados y comercializados por *Microsoft*. Existen versiones para hogares, empresas, servidores y dispositivos móviles, como computadores de bolsillo y teléfonos inteligentes. Hay variantes para procesadores de 16, 32 y 64 *bits*.

XFree86-3.3.6.- Provee una interfaz cliente/servidor entre el hardware de pantalla y el ambiente de escritorio.

XHTML.- Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

XMLHttpRequest.- Es una interfaz empleada para realizar peticiones *HTTP* y

HTTPS a servidores *WEB*.

XSLT.- Es un estándar de la organización *W3C* que presenta una forma de transformar documentos *XML* en otros e incluso a formatos que no son *XML*.

Bibliografía

- Anaya, A., Boticario, J.G. «Application of machine learning techniques to analyse student .» *Expert Systems with Applications: Special* , (2010).
- Bhavani Thuraisingham, Latifur Khan, Mohammad M. Masud, Kevin W. Hamlen. *ata Mining for Security Applications*, . IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008.
- D. Borrajo, J. G. Boticario, P. Isasi. "Aprendizaje Automático". Editorial Sanz y Torres, 2005.
- Dave Crane, Eric Pascarello, Darren James. *Ajax in Action*. 2006.
- Dorfman, M., Thayer, R. «Software Engineering.» *IEEE Computer* , 2007.
- Elías-Gutiérrez. «Descubrimiento de conocimientos en la base de datos académica de la Universidad Autónoma de Manizales aplicando arboles de decisión.» 2012.
- Fowler, Martin, y Kendall. Scott. *UML Distilled*. Addison-Wesley, 1999.
- Franco-Arcega, A., Carrasco-Ochoa J. A., Sánchez-Díaz, G., y Martínez-Trinidad, J.F. «Decision tree based classifiers for large datasets. Computación y sistemas.» 2013.
- García-Martínez, R., Lelli, R., Merlino, H., Cornachia, L., Rodriguez, D., Pytel, P., y Arboleya. «Ingeniería de proyectos de explotación de información para PyMEs.» *XIII Workshop de investigadores en Ciencias de la Computación. Red de Universidades con Carreras en Informática* , 2011.
- Glover, F. y Laguna, M. «Tabu Search. Modern Heuristic Techniques for Combinatorial Problems.» C. Reeves, ed., *Blackwell Scientific Publishing*, 1993.
- Hwang, Min Qin and Kai. *Anomaly Intrusion Detection by Internet Data mining of Traffic Episodes*. ACM, TISSec, 2004.
- Kendall. *Analisis y diseño de sistemas*. 2006.
- Korth, Silberschatz. *Fundamentos de Bases de Datos*. 2006.
- Leite, J.C.S.P. "Ingeniería de Requisitos". 1997.
- Mechán, L. P., Gómez, D. A., Borja, E. O., Flores, M. E. O., y Torres, P. A. L.

- «Comparativo para la fase de análisis de requisitos entre un método de desarrollo de software tradicional vs. La incorporación de BPM.» 2012.
- Pavón, J. *CREACIÓN DE UN PORTAL CON PHP Y MYSQL, 2ª EDICIÓN*. RAMA, 2006.
- Pressman, R. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, Inc, 2005.
- R.S. Sutton, A.G. Barto. *"Reinforcement Learning: an introduction"*. MIT Press, 2000.
- Sabrina Sestito, S Dillon. *Automated Knowledge Acquisition*. Prentice Hall, 2001.
- Santos, O.C., Rodríguez, A., Gaudioso, E., Boticario, J.G. «Helping the tutor to manage a.» De 2003: Supplementary Proceedings, Universidad de Sidney (AIED). Australia, p.153-162, 2003.
- SCHACH. *Análisis y diseño orientado a objetos UML*. 2005.
- SENN, James A. *Análisis y Diseño de Sistemas de Información*. 1996.
- Sierra, B. *Aprendizaje Automático: conceptos básicos y .* Editorial Pearson Prentice Hall, 2006.
- Stolfo, Wenke Lee and Salvatore J. «Data Mining Approaches for Intrusion Detection.» San Antonio, Texas, 1998.
- Welling, Luke, y Laura Thomson. *DESARROLLO WEB CON PHP Y MYSQL*. ANAYA MULTIMEDIA, 2005.
- Zakas, Nicholas C. *Professional JavaScript for Web Developers*. 2003.
- Campderrich Falgueras Benet, Manspons Ramon.
"Ingeniería del software", Oberta de Catalunya, Editorial UOC, 2003.
- Sommerville, Ian, *Software Engineering (6th Edition)*, Addison Wesley
- Eguíluz, Pérez Javier, *Introducción a AJAX*, Primera Edición
- Eguíluz, Pérez Javier, *Introducción a JavaScript*, Primera Edición