

Reglas de Predicción aplicables
al Diseño de un Curso de Computación

Autor : María Delia Grossi

Directora : Prof. Lic. Laura Lanzarini

Trabajo Integrador para optar por el título de Especialista en
Tecnología Informática Aplicada en Educación

Marzo de 2008

Objetivo

Esta monografía describe la manera de utilizar técnicas de Minería de Datos para la mejora del proceso de enseñanza-aprendizaje de la asignatura Computación que se desarrolla en la Facultad de Ingeniería de la Universidad de Buenos Aires.

El enfoque propuesto busca modelizar la interacción del alumno con el material de estudio utilizando reglas de predicción cuya interpretación permitirá detectar las falencias del proceso educativo así como evaluar la calidad del material de estudio utilizado.

Prefacio

La Minería de Datos es un término que engloba un amplio conjunto de técnicas de análisis de datos y extracción de modelos. Su objetivo principal es la obtención de conocimiento procesable, implícito en las bases de datos.

La aplicación de este tipo de técnicas, capaces de descubrir relaciones interesantes a partir de la información disponible, a sistemas educacionales es un área de investigación novedosa que aun se encuentra poco desarrollada en nuestro país.

A través de esta monografía se describirá la manera de utilizar técnicas de Minería de Datos para la mejora del proceso de enseñanza-aprendizaje de la asignatura Computación que se desarrolla en la Facultad de Ingeniería de la Universidad de Buenos Aires.

La organización de esta presentación es la siguiente:

- En el capítulo 1 se describirá la asignatura Computación que será motivo de estudio a lo largo de todo el trabajo. En este punto se incluirá un breve marco teórico como apoyo a la metodología utilizada actualmente y se indicarán las falencias detectadas en el proceso de aprendizaje.
- Luego, en el capítulo 2 se introducirán los conceptos básicos referidos a las principales aplicaciones de la Inteligencia Artificial en la Educación, como Sistemas Tutoriales Inteligentes y Sistemas Adaptativos. En ambos casos el énfasis estará puesto en su aplicación a los procesos educativos basados en la Web.
- En el capítulo 3 se describirán los conceptos básicos de Minería de Datos y se hará un recorrido por los objetivos de la Minería de Datos aplicada a la Educación, las técnicas más utilizadas en esta área y las principales líneas de investigación.
- En el capítulo 4 se presentará una propuesta de cambio de la estrategia de enseñanza de la materia.
- Para llevar a cabo el cambio de la metodología de enseñanza citado anteriormente, en el capítulo 5 se propone el diseño del curso a través de un Sistema Hipermedia Adaptativo basado en la Web. Este diseño incluye la posterior aplicación de técnicas de Minería de Datos con el objetivo final de hacer modificaciones en el curso para mejorar la enseñanza.
- Finalmente el capítulo 6 resume lo realizado en los puntos anteriores.

Índice general

Objetivo	I
Prefacio	II
1. ESTRATEGIA DE ENSEÑANZA ACTUAL	1
1.1. Caracterización de la Asignatura	1
1.1.1. Sus Objetivos	1
1.1.2. El Contenido y su Organización	2
1.2. Caracterización de los Destinatarios	3
1.3. Caracterización del Curso	3
1.4. Estrategia de Enseñanza	4
1.4.1. Marco Teórico: Modelos de Aprendizaje	4
1.4.2. Comunicación de los Contenidos	5
1.4.3. Resultados Observados	7
2. Sistemas Inteligentes Educativos	9
2.1. Introducción	9
2.2. Sistemas Tutores Inteligentes	10
2.2.1. Arquitectura	10
2.2.2. Funcionamiento y Características	12
2.3. Hipertexto e Hipermedia	13
2.4. Sistemas Hipermedia Adaptativos basados en la Web	15
2.4.1. AHA! Un Ejemplo de Sistema Hipermedia Adaptativo	17

3. Minería de Datos	20
3.1. Introducción	20
3.2. Minería de Datos en Educación	22
3.3. Tareas y Técnicas más utilizadas en Educación	23
3.3.1. Predicción utilizando Reglas de Asociación	23
3.3.2. Clasificación	25
3.3.3. Agrupamiento (clustering)	29
3.3.4. Regresión	30
3.4. Selección de una Técnica de Minería de Datos	30
3.5. La Computación Evolutiva	31
3.5.1. Algoritmos Genéticos	31
3.6. Software de Minería de Datos	33
4. Propuesta de cambio en la Estrategia de Enseñanza	34
4.1. Metodología utilizada actualmente	34
4.2. Modificaciones propuestas	35
5. Propuesta de Diseño de un curso de Computación	39
5.1. Diseño del Modelo Adaptativo	40
5.1.1. Módulo del Dominio	40
5.1.2. Módulo del Alumno	41
5.1.3. Módulo del Tutor o de Adaptación	43
5.2. Integración de los Datos	44
5.3. Información a Descubrir	45
6. Conclusiones y Perspectivas	49
A. Tareas y Técnicas de Minería de Datos	52
B. Herramientas de Software para Minería de Datos	53
B.1. SPSSClementine	53

B.2. WEKA (Waikato Environment for Knowledge Análisis)	53
B.3. Kepler	54
B.4. ODMS (Oracle Data Mining Suite) Darwin	54
B.5. DBMiner	54
B.6. DB2 Intelligent Miner	55
B.7. SASEnterprise Miner	55
B.8. STATISTICA Data Miner	55

Capítulo 1

ESTRATEGIA DE ENSEÑANZA ACTUAL

1.1. Caracterización de la Asignatura

La asignatura Computación que se desarrolla en la Facultad de Ingeniería de la Universidad de Buenos Aires está considerada de formación básica obligatoria para todos los alumnos de las Carreras de Ingeniería, excepto para aquellos que cursan Ingeniería Electrónica e Ingeniería en Informática. Tiene una carga horaria semanal de cuatro horas.

1.1.1. Sus Objetivos

Los Objetivos de la Materia son los siguientes:

- Brindar al alumno una visión global de la Computación de manera que comprenda el aspecto científico de la actual sociedad informatizada y conozca herramientas y vocabulario de profesionales de informática de manera tal de poder interactuar con ellos interdisciplinariamente.
- Lograr que el alumno se involucre con las tecnologías y herramientas fundamentales de la Computación de manera que aprenda a usar la computadora como herramienta de trabajo, conociendo su precisión, capacidad y limitaciones.
- Enseñar al alumno de Carreras de Ingeniería el Análisis, la Sistematización, Programación y Procesamiento de distintos problemas de tipo técnico-científicos a fin de que dichos conocimientos le resulten de utilidad ya sea en el desarrollo de la carrera como así también en su actividad profesional. Concientizar al alumno de la importancia de la Algoritmia como paradigma de resolución de problemas, y de la Programación como práctica y ejercitación en la resolución de problemas.

1.1.2. El Contenido y su Organización

La institución, a través del Programa Sintético, establece el contenido mínimo a ser desarrollado:

- ALCANCE DE LAS CIENCIAS DE LA COMPUTACIÓN. TÉCNICAS PARA REPRESENTAR Y ALMACENAR INFORMACIÓN Y FORMA EN QUE LAS MÁQUINAS DIGITALES MANIPULAN LOS DATOS. ARQUITECTURA DE COMPUTADORAS. SOFTWARE DE SISTEMA, DE APLICACIÓN Y DE TRADUCCIÓN. LENGUAJES DE PROGRAMACIÓN. ALGORITMIA Y PROGRAMACIÓN BÁSICAS.

A partir del mismo, como pauta de organización, se toman ciertas decisiones relativas a cuáles son las estrategias oportunas para organizar globalmente la enseñanza de la asignatura de tal modo que los alumnos puedan ir adquiriendo el conocimiento del modo más eficaz posible, para lo cual es necesario definir y delimitar el contenido.

La elección del contenido se realiza privilegiando los conceptos y técnicas de campos de la disciplina que contribuyan al logro de los objetivos de la materia.

La enseñanza de algoritmia y programación comprende el descubrimiento de algoritmos como proceso de resolución de problemas y su representación para que puedan ser comunicados a una computadora. Se considera que los alumnos están en mejor condición para adquirir nociones referidas a traducción e interpretación de programas, si poseen previamente en su estructura cognoscitiva los conceptos de lenguaje de máquina, variable y constante. Lo mismo ocurre con las nociones de procesamiento de archivos y representación de datos en una computadora habiendo adquirido previamente nociones básicas de arquitectura de computadoras. Estos conocimientos los ilustran acerca de las bases sobre las cuales deben trabajar y los concientiza del potencial de la computadora como herramienta de trabajo y de las limitaciones del software que produzcan. De igual manera, los acerca a conceptos y técnicas de la disciplina que en su futuro profesional los habilitará a interactuar interdisciplinariamente con profesionales en informática sin problemas de comunicación.

La materia está dividida en tres módulos teóricos y en tres módulos prácticos. En los primeros, se procura que los alumnos adquieran una visión global de la computación y comprendan el aspecto científico de la actual sociedad informatizada. En tanto que, en los módulos prácticos, se procura que los alumnos adquieran las habilidades necesarias para desarrollar algoritmos que resuelvan problemas concretos y logren expresarlos mediante un lenguaje de programación.

1.2. Caracterización de los Destinatarios

■ Edad

La edad de los alumnos puede ser variable como ocurre en el sistema universitario en general. La masa crítica está formada por alumnos recientemente egresados del ciclo secundario, cuyas edades están comprendidas entre 18 y 19 años. Como los alumnos pueden cursar esta materia en distintas etapas de avance de los planes de estudio existe un porcentaje menor de alumnos de mayor edad.

■ Conocimientos previos

El nivel de conocimientos previos de los alumnos es variable dado que provienen de establecimientos de distintos puntos del país y de diferentes sistemas educativos. Algunos son egresados del ciclo secundario tradicional y otros del Polimodal en sus diversas orientaciones. Los alumnos que provienen de escuelas técnicas generalmente tienen conocimientos y experiencia previa en programación, en cambio otros poseen experiencia en la operación de computadoras para uso personal.

■ Manejo tecnológico

En términos generales la mayoría de los alumnos poseen conocimientos básicos de la operación de computadoras e Internet, adquiridos en los niveles educativos previos, en cursos realizados individualmente por iniciativa personal o directamente a través del aprendizaje informal en el hogar.

1.3. Caracterización del Curso

■ Cantidad de alumnos

Actualmente la cantidad de alumnos que se inscriben en la materia por cuatrimestre es de aproximadamente quinientos cuarenta en forma regular y ciento veinte en forma condicional. La facultad ofrece nueve cursos de sesenta alumnos de esta materia y los alumnos condicionales deben ser absorbidos en forma proporcional por cada curso.

La facultad establece también que la relación docente/alumno debe ser de un docente auxiliar por cada treinta alumnos. En la práctica, debido a la necesidad de absorber a los alumnos condicionales, esta relación se transforma en un docente auxiliar por cada treinta y cinco alumnos.

■ El aula

Generalmente las aulas destinadas al desarrollo del curso tienen una capacidad de cien alumnos, son angostas y largas y los bancos tienen capacidad para cinco alumnos.

1.4. Estrategia de Enseñanza

1.4.1. Marco Teórico: Modelos de Aprendizaje

En base al escenario planteado y considerando que la relación de enseñanza es una relación ternaria, en la cual intervienen el docente, el alumno y el contenido inmersos en un contexto, se considera apropiado que el aprendizaje sea por recepción pero significativo. Desde la perspectiva de Ausubel, para lograr un aprendizaje significativo, es necesario que el contenido nuevo a aprender, resulte potencialmente significativo, con pertinencia lógica y psicológica, que el alumno posea una estructura cognitiva de anclaje al tema a aprender, y que él muestre una predisposición o actitud significativa hacia el aprendizaje. Se busca la predisposición a aprender significativamente, y la recuperación de aquellos conceptos ya conocidos, implícitos, como pueden ser “la computadora”, el concepto de “algoritmo” o de “lenguaje de programación”.

Mediante ejemplos relacionados con una materia básica de la carrera de Ingeniería, se trata de que los estudiantes reconstruyan significados a partir de procesos de discusión. El estudiante tiene un rol activo pues escucha las explicaciones del docente (mediador), interactúa con el docente en la solución de los distintos enunciados y reflexiona acerca de los errores que pueden tener las distintas propuestas para resolver dichos enunciados [34]. A los efectos de facilitar la organización lógica y estructurada de los contenidos que los estudiantes deben aprender se recurre a mapas conceptuales. Esta técnica permite al docente presentar el material en forma clara y sencilla partiendo de conceptos generales como computadora hasta alcanzar ideas particulares como estructuras de control selectivas. El alumno, a través del mapa conceptual percibe el contenido de la asignatura o del módulo temático como una unidad. Según la percepción de Bruner, *“el conocimiento adquirido, sin estar dotado de la estructura suficiente como para fijarlo, se olvida con facilidad. Un conjunto inconexo de hechos tiene una vida lamentablemente corta en la memoria. Organizar hechos con arreglo a principios e ideas de los que pueden ser inferidos es el único modo que se conoce de reducir la rápida tasa de pérdidas en la memoria humana”* [22]. Una mención especial merece la estrategia utilizada para enseñar a resolver problemas. La resolución de problemas se lleva a cabo en las diferentes etapas del proceso de enseñanza y aprendizaje y es una de las mayores dificultades que presentan los alumnos.

Raymond, Nickerson, Perkins y Smith [28] resumen la visión de la bibliografía psicológica al hablar de resolución de problemas, expresando que *“son procesos de conducta y pensamiento dirigidos hacia la ejecución de determinada tarea intelectualmente exigente”*.

Existen diferentes métodos, principios y reglas que tratan de describir cómo resolver problemas.

En el curso se utiliza el método expuesto por el matemático Polya en 1957 en su tratado de la solución de problemas [13].

Las fases de la metodología de Polya son las siguientes:

1. Comprender el problema.
2. Idear un plan, es decir, diseñar una estrategia que permita resolverlo.

3. Ejecutar el plan.
4. Verificar que los resultados obtenidos se correspondan con el objetivo del problema resuelto.

Estas fases deben completarse en algún momento durante el proceso de resolución, que en general implica repasar una y otra vez las fases en forma evolutiva, retrocediendo algún paso de ser necesario, hasta que la solución sea satisfactoria.

La aplicación de este modelo al ámbito de la creación de programas lleva a las siguientes fases:

1. Análisis: entender el problema y definirlo claramente.
2. Diseño: determinar los recursos a utilizar y cómo utilizarlos
3. Codificación: representar el algoritmo en forma de programa.
4. Evaluación: evaluar el programa en cuanto a su exactitud y a su potencial como herramienta para resolver otros problemas (probar y depurar el programa).

En [28], se menciona que *“muchos autores al escribir sobre la solución de problemas, cuando uno tiene una dificultad mayor de lo normal con un problema, lo mejor que se puede hacer es tratar de hallar un modo radicalmente distinto de representarlo”*. Con el objetivo de concientizar a los alumnos de esta práctica, en la primera clase se hace la presentación de la figura 1.1.

1.4.2. Comunicación de los Contenidos

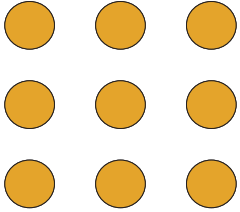
Partiendo del concepto de Transposición Didáctica, proceso que consiste en modificar el contenido epistemológico y transformarlo en contenido didáctico [41], se presentará la estrategia utilizada actualmente por la cátedra para llevar a cabo esta transposición. En la figura 1.2 se presenta en forma esquemática la organización de los módulos teóricos y prácticos.

Cada módulo, teórico o práctico, comienza con una presentación multimedia. De acuerdo a la definición de Guillem Bou Bouzá, *“un sistema multimedia es el que transmite una información mediante imagen, sonido y texto de forma sincronizada, y que hace uso adecuado de la capacidad de usar los diferentes canales de comunicación”* [17].

Daniel Prieto Castillo [6], sostiene que se logra una mayor relación de aprendizaje cuando un material es elaborado aprovechando al máximo las posibilidades comunicacionales del lenguaje en que está armado el mensaje y del medio a través del cual se lo ofrece. Utilizar todas las capacidades expresivas de un lenguaje y comunicativas de un medio, constituye un modo de acercarse a los interlocutores y, por lo tanto, acompañar y promover el aprendizaje [6].

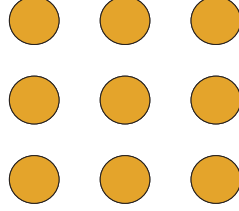
Veamos un problema

Se desean unir entre si los nueve círculos de la siguiente matriz con cuatro líneas rectas continuas. Cuentan con papel y lápiz.



1

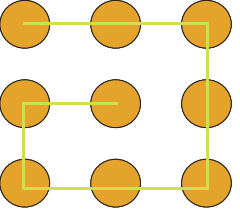
Otra mirada



3

Veamos un problema

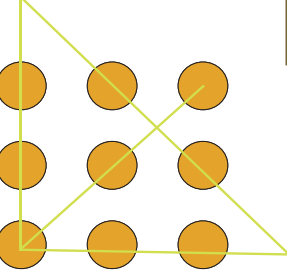
Se desean unir entre si los nueve círculos de la siguiente matriz con cuatro líneas rectas continuas. Cuentan con papel y lápiz.



Se utilizaron 5 líneas.
Repensar la solución

2

Otra mirada



Perfecto!! Mirando el problema desde otro ángulo se pudo resolver.

Recursos	Interlocutor
<ul style="list-style-type: none"> papel lápiz círculos pintados 4 Líneas rectas 	<ul style="list-style-type: none"> Persona que traza líneas
Acciones	
<ul style="list-style-type: none"> Trazado de cada línea 	

4

Figura 1.1: Estrategia a seguir cuando no se logra resolver un problema

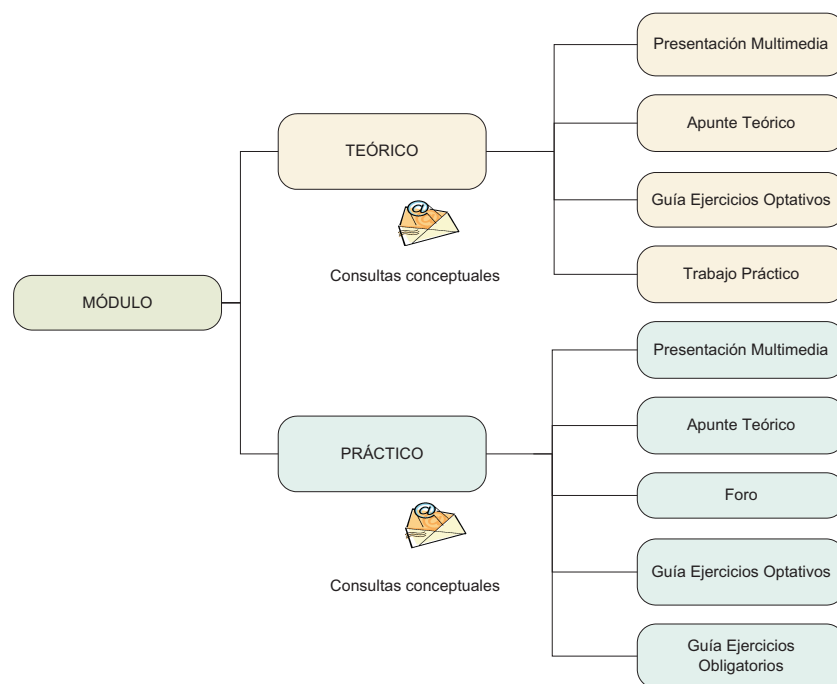


Figura 1.2: Organización del Contenido de la Asignatura Computación

A través de la presentación multimedia se busca considerar la presencia de los estilos cognitivos posibles, y de las capacidades cognitivas a poner en juego procurando desarrollar una organización de la información según colores, títulos, diagramas, imágenes, ítems, listados, redes conceptuales y puesta en contexto de conceptos.

Los conceptos se presentan mediante organizadores previos, que sirven de puente cognitivo entre los conocimientos que el alumno ya posee en su estructura cognitiva y los conceptos a ser aprendidos, durante el desarrollo de la asignatura.

Los Apuntes Teóricos y Prácticos correspondientes a los diferentes módulos se presentan en formato impreso. Abordan en forma concisa y conceptual los temas teóricos y prácticos que se tratan en las clases. Las Guías de Ejercicios Optativos, también en formato impreso, tienen por objeto que los alumnos adquieran las habilidades básicas para poder aplicar los conceptos de cada módulo en forma adecuada.

En la Parte Práctica, luego de la presentación de cada módulo práctico se lleva a cabo un foro de participación optativa que tiene por objetivo ayudar a la comprensión del material desarrollado en la presentación y en el apunte. Los foros constituyen espacios fundamentales de participación, donde los alumnos pueden aportar y tomar ideas y conceptos de los demás.

Los Trabajos Prácticos consisten en actividades generativas. Una actividad generativa busca la capacidad de aplicar principios y conceptos centrales a la resolución de problemas concretos. Corresponden al área de producción. Se considera que la aplicación del conocimiento adquirido por el alumno a la resolución de problemas del área de la Ingeniería aumenta su interés por el tema.

En determinados módulos prácticos cada alumno en forma individual debe cumplimentar con la resolución de un problema sencillo y conceptual de entrega obligatoria a través de la mensajería. Constituye una actividad diagnóstica que le permite al docente detectar errores y reforzar aquellos temas que denoten una mala interpretación.

1.4.3. Resultados Observados

Durante los últimos tres cuatrimestres se han aplicado diversas estrategias didácticas hasta alcanzar el enfoque presentado anteriormente. Los cambios obedecen a la necesidad de modificar la metodología de enseñanza para lograr mejorar el aprendizaje de los alumnos.

La incorporación de un entorno de trabajo informatizado ha sido, sin lugar a dudas, una de las estrategias con mayor impacto en el proceso de enseñanza-aprendizaje de esta materia. La asesoría virtual a través del correo electrónico permitió al docente estar disponible en forma asincrónica y distribuida. El alumno por su parte, en el momento que desee, tiene la posibilidad de definir y plantear sus dudas, no sólo en forma particular al docente sino a los restantes miembros del grupo de estudio, a través de un foro que se convierte en un espacio de aprendizaje colaborativo en el cual docentes y alumnos pueden aportar y tomar ideas y conceptos de los demás.

Esta estrategia permitió incrementar la participación de los alumnos en el proceso educativo corrigiendo de esta forma una de las falencias detectadas en la modalidad presencial.

Actualmente se cuenta con el prototipo del material multimedial de los módulos teóricos y se está trabajando en la elaboración del material necesario para presentar al alumno el contenido de la parte práctica del curso de una forma “amigable” y motivadora. Con esto se espera reducir aun más la brecha que separa a los estudiantes de Ingeniería del uso de las herramientas informáticas básicas.

Para llevar a cabo esta tarea, es preciso definir las diferentes modalidades que permiten la utilización de la computadora en el proceso de enseñanza-aprendizaje. Taylor [38] plantea tres modalidades:

- Tutor: la computadora guía el aprendizaje del alumno mediante el conocimiento aportado por un experto en la disciplina a enseñar.
- Herramienta: la computadora facilita el proceso de aprendizaje del alumno permitiéndole automatizar determinadas tareas. Por ejemplo: los procesadores de texto.
- Aprendiz: la computadora se transforma en el alumno y ejecuta algoritmos diseñados por el estudiante. Por ejemplo, la utilización de lenguajes de programación para el aprendizaje de una disciplina, como el Logo.

Capítulo 2

Sistemas Inteligentes Educativos

2.1. Introducción

La creación de Sistemas Inteligentes Educativos (SIE) se enfoca más como una herramienta complementaria de la enseñanza y del aprendizaje que permite aumentar su calidad más que como una herramienta que los sustituye.

El desarrollo de los primeros sistemas de enseñanza asistida por computadora (CAIs), se remonta a los años 50, época en que aparecen los programas lineales. Estos programas tenían un orden establecido por el programador y ningún factor podía alterarlo. El modelo respondía a un enfoque del alumno como el que se definía en el marco de las teorías conductistas de aprendizaje: *“las personas aprenden una conducta o comportamiento de un mundo externo (real), excluyendo el papel de la mente”*. Aprender se considera como la formación de asociaciones entre estímulos y respuestas, se basa en aplicar recompensas y castigos. Es decir, se entiende por aprendizaje aquel cambio de conducta esperada en respuesta a un estímulo determinado. Este cambio de conducta se produce como resultado de la experiencia, es decir, de la interacción de la persona con su entorno. El contenido se abordaba desde lo más simple a lo más complejo. En los sistemas lineales, en el desarrollo de una sesión de enseñanza no se tiene en cuenta la aptitud del alumno.

Posteriormente, Crowed en 1959 implementó programas ramificados que, a pesar de tener un número fijo de temas, estaban dotados de una cierta capacidad de actuar según la respuesta del alumno repitiendo textos de explicación, volviendo a hacer ejercicios, etc. De todas formas, tampoco pueden ofrecer una enseñanza individual al alumno. El sistema responde ante una acción en forma independiente del alumno.

A finales de los años sesenta y principios de los años setenta (1967-1971) surgieron los sistemas generativos (también llamados sistemas adaptativos). Este enfoque plantea la necesidad de que la enseñanza se adapte a las necesidades del alumno. Considera que *“el alumno aprende mejor enfrentándose a problemas de dificultad adecuada que atendiendo a explicaciones sistemáticas”*. Este tipo de sistemas tiene capacidad para determinar el grado de dificultad del problema que se le presenta al alumno. Una vez que el alumno emite su respuesta, el sistema la compara con su solución y las diferencias entre ambas

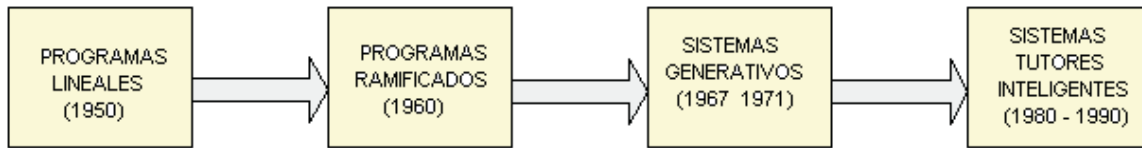


Figura 2.1: Evolución de los Sistemas de Enseñanza

respuestas se consideran errores. Una limitación de estos sistemas es la relación entre la cantidad de soluciones que establecen para un problema (una) y la cantidad real de posibles soluciones que tiene ese problema. Como evolución de los sistemas CAI y con la incorporación de técnicas de Inteligencia Artificial (IA) aparecieron los Sistemas Tutores Inteligentes (STI) [20].

En la figura 2.1 se grafica la evolución de los Sistemas de Enseñanza.

La aplicación de la IA a la enseñanza a través de computadoras tiene como objetivo final construir sistemas de enseñanza inteligentes que simulen a un buen profesor. Los STI se caracterizan por su capacidad para comportarse como expertos tanto en el “*qué enseñar*” como en el “*cómo enseñar*”. El “*qué enseñar*” representa el dominio del conocimiento, en tanto que el “*cómo enseñar*” representa el dominio didáctico. De acuerdo a la situación de aprendizaje en la que se encuentra el estudiante, el dominio didáctico le ofrecerá una acción o solución que le permita progresar en su proceso de aprendizaje. Un principio general que rige la concepción de un STI es que debe ser capaz de adaptar sus instrucciones para poder considerar la presencia de los distintos estilos cognitivos que pueda tener el alumno.

Los STI hacen uso del conocimiento de que disponen para guiar el proceso del tutorial y adaptarse al nivel, preferencias y otras características del alumno.

2.2. Sistemas Tutores Inteligentes

2.2.1. Arquitectura

Un STI permite separar lo que se enseña del cómo se enseña. Está formado por: el modelo del experto o del dominio (Conocimiento Pedagógico), el modelo didáctico, el modelo del alumno y el modelo que representa la interfaz con el alumno.

La figura 2.2 representa la arquitectura de un STI diseñada por Kaplan [15]:

El **Modelo del Alumno** almacena información sobre las características de cada alumno: sus datos personales, los temas y actividades a las cuales accedió, las pantallas visitadas, su comportamiento frente a las diferentes actividades (sus errores, sus aciertos) y las calificaciones obtenidas en las evaluaciones. Mantiene un registro histórico acerca de las dificultades y progresos que experimenta el alumno. Modela “lo que sabe y lo que no sabe”. Infiere su estado cognitivo a partir de la información almacenada. Este modelo se

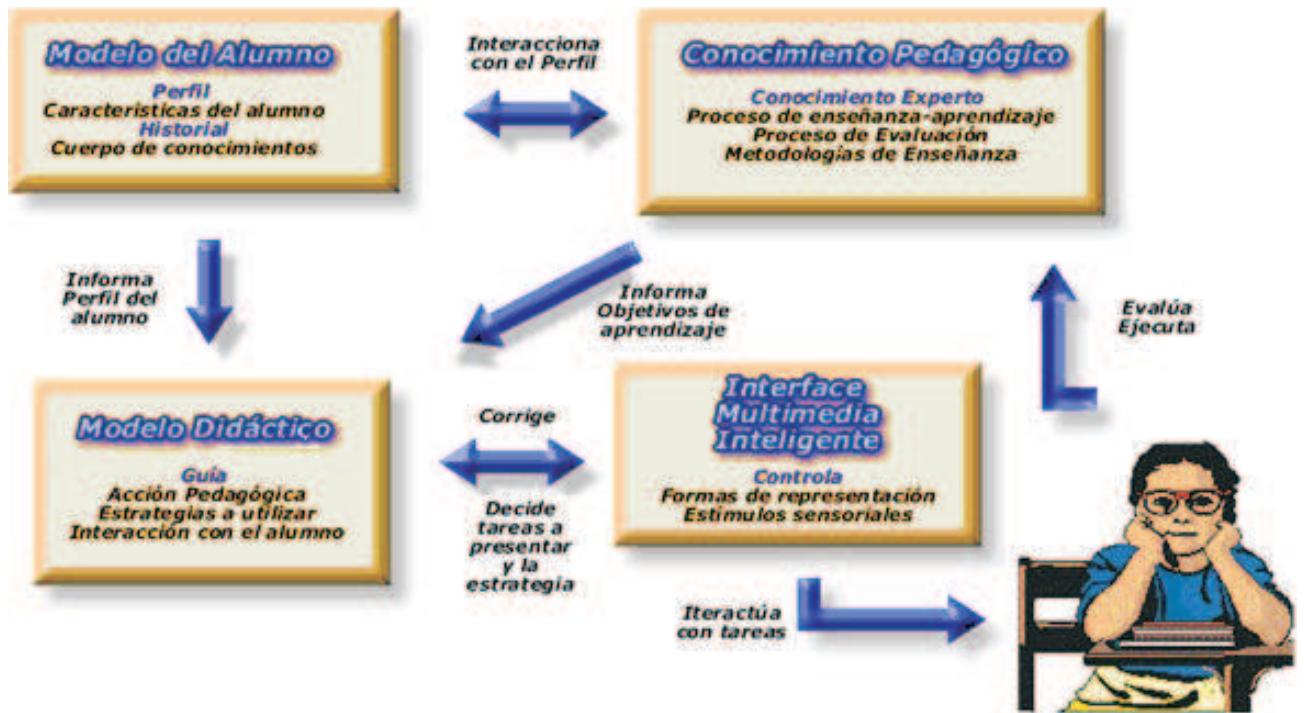


Figura 2.2: Arquitectura de un STI

forma y actualiza con cada interacción del alumno con el sistema.

El **Modelo del Experto o del Dominio** contiene la representación del conocimiento experto. Constituye el “qué enseñar”. El docente, experto en conocimiento, cumple la función importantísima de definir los principios conceptuales y procedimentales de la materia. En la asignatura Computación, algunos ejemplos de principios conceptuales son:

- La Computación es una disciplina que busca establecer una base científica para resolver problemas mediante la computadora[13].
- Una idea fuerza de la computación es la de algoritmo.
- Todo algoritmo implica una secuencia y orden.
- Los algoritmos no deben ser ambiguos.
- En Informática los algoritmos se representan como programas.
- Los programas otorgan inteligencia a las computadoras.
- Los algoritmos se comunican a las computadoras por medio de Lenguajes de Programación.

y como ejemplos de principios procedimentales se pueden citar:

- La resolución de algoritmos deberá realizarse utilizando el método de refinamientos sucesivos.
- Los programas se resolverán utilizando la metodología propuesta por Polya [13].
- Todo programa deberá estar documentado.
- Se diseñarán programas modulares.
- El objetivo de diseñar un programa modular debe ser maximizar la independencia de los módulos.

Un error en la representación del conocimiento llevaría a una mala enseñanza. Este módulo es el responsable de dirigir la ejecución del “Modelo Didáctico” teniendo en cuenta los datos ingresados desde el “Modelo del Alumno”.

El **Modelo Didáctico** ofrece una metodología para el proceso de aprendizaje. Representa al tutor o profesor en su papel de transmisor del conocimiento. Caracteriza el “cómo enseñar”. Almacena información referida a las estrategias de motivación, metodologías de aprendizaje y secuencia de tareas a ofrecer al alumno. En función de su historial de interacciones determinará el tipo de material o de evaluación que se le presentará al alumno. Este módulo permite activar el módulo de la interface.

El **Modelo de Interface** constituye el nexo entre el tutor y el alumno. Debe permitir que el alumno acceda al material didáctico de una forma sencilla y clara. Los tiempos de respuesta del sistema no deben ser extensos. Debe tener la capacidad de mediar el conocimiento a través de diferentes canales con el objetivo final de lograr una enseñanza adaptada, eficiente y que favorezca la motivación del alumno por el aprendizaje.

2.2.2. Funcionamiento y Características

Una sesión en un STI comienza con la identificación del alumno. A continuación, el Modelo Didáctico se comunica con el Modelo del Dominio para conocer cuál es el tema que el alumno va a aprender, y con el Modelo del Alumno para obtener información acerca de su perfil y nivel de conocimiento. De acuerdo a los datos recabados el Modelo Didáctico determinará la estrategia de enseñanza más adecuada para el alumno. La estrategia de enseñanza del tutor podrá consistir en:

- La formulación de preguntas al alumno. El tutor podrá presentar una pregunta o problema junto con respuestas o soluciones alternativas para evaluar cuánto sabe un alumno sobre un tema.

- Orientar las actividades del alumno.
- La interacción con el alumno permitiendo el intercambio de conocimientos.
- La generación de actividades que permitan reforzar el conocimiento del alumno.

El Modelo de Interface será el encargado de llevarla a la práctica.

Algunos de los factores que caracterizan a los STI son:

- El conocimiento del dominio está acotado y debe contener la totalidad del conocimiento que se desea transmitir.
- El sistema, a través del conocimiento del alumno, puede adaptar la enseñanza.
- La secuencia de enseñanza no está predefinida por el docente (a diferencia de los sistemas lineales) y debe ajustarse al perfil del alumno.
- La construcción de los STI se realiza bajo el marco teórico de las teorías cognitivas.

Una de las críticas que numerosos autores le plantean a los STI es que convierten a la tarea educativa en excesivamente restringida a las órdenes del tutorial [33].

2.3. Hipertexto e Hipermedia

Cuando se reflexiona acerca de cómo plantear y desarrollar un curso en un entorno virtual, se debe tener presente que el soporte que se utilizará será básicamente el texto escrito. A este texto se le podrán incorporar diferentes posibilidades tecnológicas como animación, hipertextos, imágenes y sonido.

El hipertexto se caracteriza por la organización no lineal de la información [18]. El contenido textual se vincula de múltiples maneras. El alumno se enfrenta a diferentes estrategias de lectura. El hipertexto concreta la idea de una lectura interactiva donde la interacción permite reforzar el mensaje y que el alumno tenga una función activa en el proceso de enseñanza/aprendizaje.

Bajo todo sistema hipertextual se encuentra la intencionalidad del docente que diseña una “interacción dirigida”. Es imprescindible que haya “un hilo conductor” entre los diferentes fragmentos que conforman el contenido textual. El alumno en todo momento debe reconocer el tema de la página que está desplegando para evitar la navegación al azar y la falta de motivación por aprender [17].

Algunas de las características de los hipertextos son la facilidad y velocidad de acceso a la información. Las uniones interactivas de información representada mediante múltiples canales de comunicación como texto, imagen, sonido y animaciones, se conocen como

hipermedios [18]. Los materiales con formato multimedia son materiales educativos que incluyen textos, imágenes, animación, videos y recursos hipertextuales [41].

En las presentaciones multimediales que permiten introducir los conceptos de cada módulo temático de la asignatura, los hipertextos tienen distintas funciones, según la sección de la aplicación en la que se encuentran:

- dentro de la ejercitación, tienen la función de recuperar claves nodales (definiciones que el usuario debe conocer para poder avanzar en el aprendizaje).
- en la bibliografía tienen función de ampliación, permiten acceder a otras presentaciones del contenido abordado desde otra fuente de consulta, como es Internet; de esta manera, el usuario puede profundizar las unidades temáticas y acceder al desarrollo de las mismas desde distintos enfoques.

Con respecto a las interfaces de las presentaciones, para facilitar la lectura se procura utilizar íconos sencillos, de fácil comprensión para manejar la navegación y acceder a la información. De igual modo se intenta componer el significado de la interface utilizando un lenguaje claro, simple, con textos breves, acompañado de imágenes que grafican, contextualizan y complementan.

Se intenta utilizar formas variadas de procesamiento de datos: visual, verbal y sus combinaciones. Se procura que el alumno participe de la presentación de los temas interactuando con la aplicación.

Se prevé circuitos de navegación que permiten que el alumno acceda a cada unidad desde el índice o desde cualquier punto de la aplicación. De esta forma puede seleccionar el contenido a abordar según su interés y, una vez seleccionado, decidir si lo va a recorrer en forma secuencial o si va a acceder directamente a las actividades o a la bibliografía. Se procura de esta forma darle al usuario los grados de libertad necesarios para “construir su propia aplicación”.

Los botones de navegación permiten al alumno avanzar, retroceder y/o volver a cualquier punto (noción de refugio). También le permiten retornar al inicio (noción de peligro) y salir de la aplicación desde cualquier página. En la figura 2.3 se muestra un ejemplo de interfaces.

Una desventaja que se le plantea a este tipo de sistemas es que el alumno puede encontrarse “muy libre”, es decir, puede llegar a recorrer caminos que no le permitan apropiarse de los principios procedimentales y conceptuales planteados por el docente. Por otro lado, no tienen capacidad para determinar el grado de avance del alumno en el aprendizaje ni para personalizar la enseñanza [32].

Considerando que en los STI el proceso de enseñanza-aprendizaje puede estar excesivamente guiado y que en los Sistemas Hipermedia (SH) puede ser muy libre, se propuso integrar los SH con los STI generando un nuevo modelo llamado “Sistema Hipermedia Adaptativo” (SHA).



Figura 2.3: Ejemplo de Interfaces

2.4. Sistemas Hipermedia Adaptativos basados en la Web

Los ambientes de aprendizaje basados en la WEB permiten conformar Comunidades Virtuales constituídas por estudiantes que comparten la misma meta de aprendizaje. De acuerdo a la definición Pere Marquès Graells [16], se puede definir una Comunidad Virtual como *“un grupo de personas que se comunican a distancia y comparten información, teniendo algún campo de interés común y una cierta conciencia de grupo”*. El entorno de comunicación al menos debe permitir comunicaciones entre dos personas y también de uno a todos.

Los Sistemas Hipermedia Adaptativos basados en la Web tienen por objetivo brindar un aprendizaje individualizado y flexible, que tenga en cuenta el conocimiento alcanzado por el alumno, sus preferencias, sus avances y sus retrocesos.

Brusilovsky en [4] da la siguiente definición de Sistema Hipermedia Adaptativo : *“Por Sistemas Hipermedia Adaptativos queremos decir, todos los sistemas hipertextos e hipermedia que reflejan algunas características del usuario en el modelo del usuario y aplican este modelo en la adaptación de varios aspectos visibles del sistema al usuario”*.

Es decir, el sistema hipermedia o hipertexto se adapta al usuario de acuerdo a la información otorgada por el modelo del usuario como se visualiza en la figura 2.4.

Los Sistemas Tutores Inteligentes y la evolución de los Sistemas Hipermedia Adaptativos han abierto el camino para el desarrollo de sistemas educativos basados en la Web.

El término adaptación se refiere a la capacidad de un sitio WEB de cambiar de es-

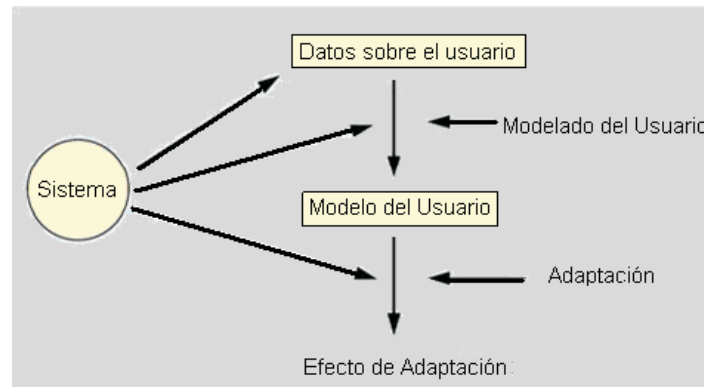


Figura 2.4: Ejemplo de estructura de un Sistema Adaptativo

trategias o de respuestas de acuerdo a las características del usuario. La adaptación podrá aplicarse tanto a un alumno como a un grupo de alumnos con intereses en común. El sistema deberá reconocer a los usuarios, “razonar” de acuerdo al perfil que tengan y planificar las acciones a llevar a cabo [11]. En particular, en el ámbito de la educación, la adaptación permitirá individualizar

- las presentaciones y las actividades,
- la recomendación de enlaces relevantes que le permitirán al alumno (o al grupo) ampliar el conocimiento accediendo a otras presentaciones del contenido abordado desde otra fuente de consulta pero en forma guiada, limitando su espacio de búsqueda y evitando que se pierda en laWEB [32].

Se debe remarcar la importancia de la navegación guiada en el ámbito de la educación pues generalmente los alumnos ingresan a un sitio educativo sin conocimiento previo del tema a abordar y la excesiva cantidad de grados de libertad en la navegación puede retardar el proceso de aprendizaje.

En la figura 2.5 se grafican las distintas tecnologías utilizadas en Sistemas Hipermedia Adaptativos. Estas técnicas difieren en el “qué adaptar”.

Los objetivos de los Sistemas Hipermedia Adaptativos basados en Web se pueden sintetizar en:

- **Personalización**

Constituye el proceso de reflexión y cambio que lleva a cabo el sistema de acuerdo al perfil del usuario. Para lograr este objetivo el sistema necesita información acerca de los conocimientos del alumno. Esta información se obtiene a través del registro histórico del alumno o en tiempo real.

- **Recomendación**

Está representado por los enlaces que se presentan en forma individualizada al alum-



Figura 2.5: Tecnologías de Adaptación utilizadas en Sistemas Hipermedia Adaptativos

no (o grupo) para recomendarle lecturas, lecciones y cursos que favorecen a la construcción del conocimiento.

■ Selección

Corresponde a la conformación del material que se le presentará al alumno (o grupo). De acuerdo a su perfil, el material podrá incluir o suprimir fragmentos, podrá abarcar la totalidad de los temas o exclusivamente los temas que se adapten al nivel de conocimiento del alumno y de esta manera evitar la información redundante y la pérdida de motivación.

2.4.1. AHA! Un Ejemplo de Sistema Hipermedia Adaptativo

En el punto anterior se analizaron los objetivos de los Sistemas Hipermedia Adaptativos como herramienta para alcanzar el objetivo de una enseñanza individualizada y flexible. A continuación se describirá el Sistema AHA! [30], un sistema de adaptación genérico, de código abierto sobre el cual se realizaron mejoras [26] tendientes a lograr una mayor adaptación a la educación. La arquitectura de este sistema se muestra en la figura 2.6

AHA! consiste en servlets Java (aplicaciones Java que se ejecutan dentro de un servidor Web) que toman páginas del sistema local de archivos o de servers externos de http. Estas aplicaciones permiten la personalización del curso a cada alumno. Está formado por el Modelo del Dominio (DM), del Alumno(UD) y del Tutor o de Adaptación (AM).

La metodología que deberá llevar a cabo el docente para crear un curso en AHA! es:

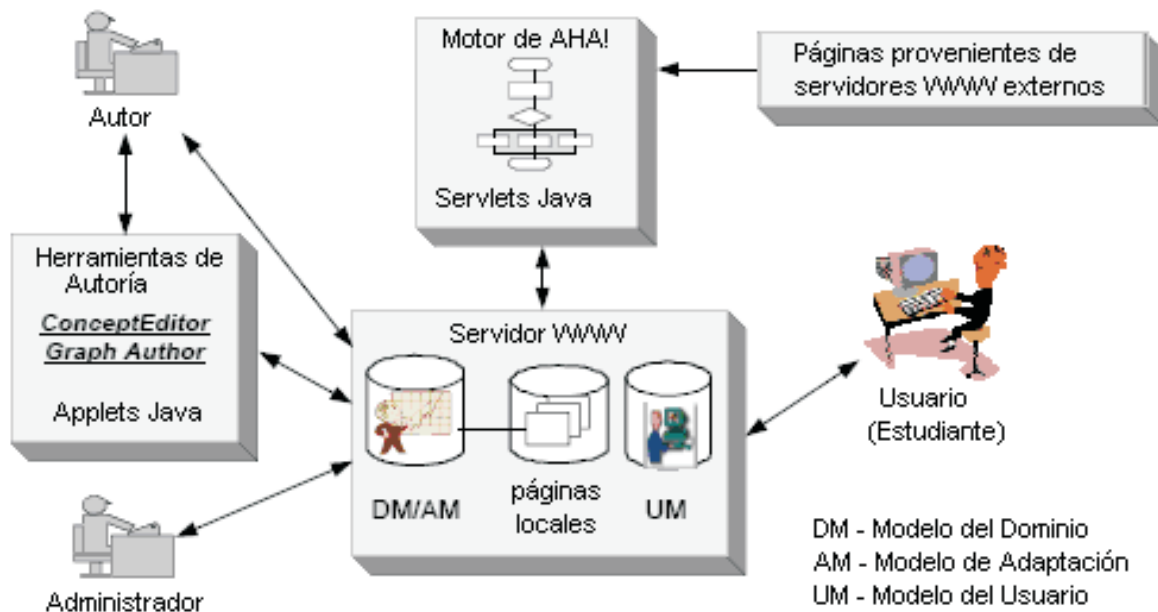


Figura 2.6: Arquitectura del Sistema AHA!

■ Representar el Modelo del Dominio

Deberá diseñar la estructura del curso. Pensar y establecer cuáles son los conceptos que se van a tratar. Un concepto es un componente que representa un ítem abstracto de información que es principal para el sistema. Cada concepto podrá abarcar una o varias páginas WEB relacionadas.

■ Representar el Modelo de Adaptación (AM)

Para ello deberá diseñar un conjunto de reglas denominadas “reglas de adaptación”. Estas reglas permiten relacionar el Modelo del Usuario con el Modelo del Dominio y definir el tipo de adaptación a llevarse a cabo, es decir relacionan los atributos de los conceptos con la respuesta del alumno y definen la estrategia a seguir. Existen herramientas de Autor que permiten escribir las reglas y graficar la estructura del curso.

Para llevar a cabo la adaptación, según se muestra en la figura 2.7, AHA! utiliza técnicas que le permiten incluir u omitir conceptos en una página [30]. Con respecto a los enlaces, utiliza tres tipos de enlaces:

- Buenos: enlaces a páginas aconsejables y que no han sido leídas (en color azul).
- Neutral: enlaces a páginas aconsejables y que ya han sido leídas (en color púrpura).
- Malos: enlaces a páginas no aconsejables (en color negro).

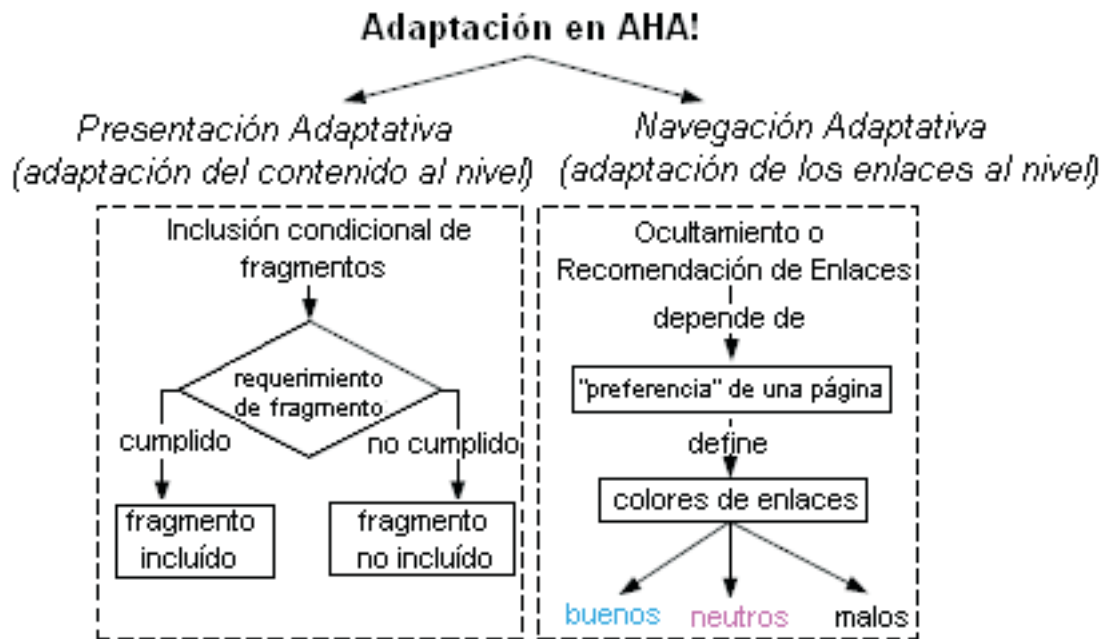


Figura 2.7: Técnicas de Adaptación utilizadas en AHA!

En el Modelo del Alumno para cada usuario se mantiene una especie de tabla en la que se almacena cada concepto con el valor de su atributo correspondiente. Contiene el nivel de conocimiento de cada alumno sobre cada concepto. Este conocimiento se obtiene a través de la respuesta del alumno a evaluaciones y a cuestionarios como así también a través del análisis de las páginas visitadas.

Capítulo 3

Minería de Datos

3.1. Introducción

La Minería de Datos es una de las fases del proceso de descubrimiento de conocimiento en bases de datos o KDD (Knowledge Discovery from Databases).

El KDD tiene por objetivo la identificación no trivial de patrones válidos, novedosos, potencialmente útiles y comprensibles a partir de bases de datos de gran volumen que pueden encontrarse tanto en el dominio científico como en el comercial. Este proceso consta de una secuencia iterativa de fases, una de las cuales es la Minería de Datos (Data Mining), como se observa en la figura 3.1.

La Minería de Datos se refiere a la aplicación de métodos de aprendizaje y estadísticos para la obtención de patrones y modelos. Además se utilizan también métodos presentes en áreas como: sistemas de bases de datos, recuperación de la información, aprendizaje automático, sistemas para la toma de decisiones, visualización de datos, computación paralela y distribuida.

Numerosos especialistas señalan que la Minería de Datos necesita de la integración de enfoques de múltiples disciplinas. Su campo de aplicación comprende los procesos industriales, el ámbito científico (medicina, astronomía, psicología, física, etc.), la actividad financiera, la política, la economía, deportes y la educación para citar algunos ejemplos [29].

Las tareas que pueden realizarse a través de las distintas técnicas pertenecientes a la Minería de Datos pueden dividirse en tres grandes áreas:

- Descripción

Este tipo de técnicas buscan derivar descripciones concisas de características de los datos. De esta manera es posible disponer de un modelo adecuado de la información existente. Por ejemplo, sería posible caracterizar un pozo de petróleo a partir de distintas mediciones realizadas con anterioridad o caracterizar un alumno en base a su situación personal y su rendimiento académico.

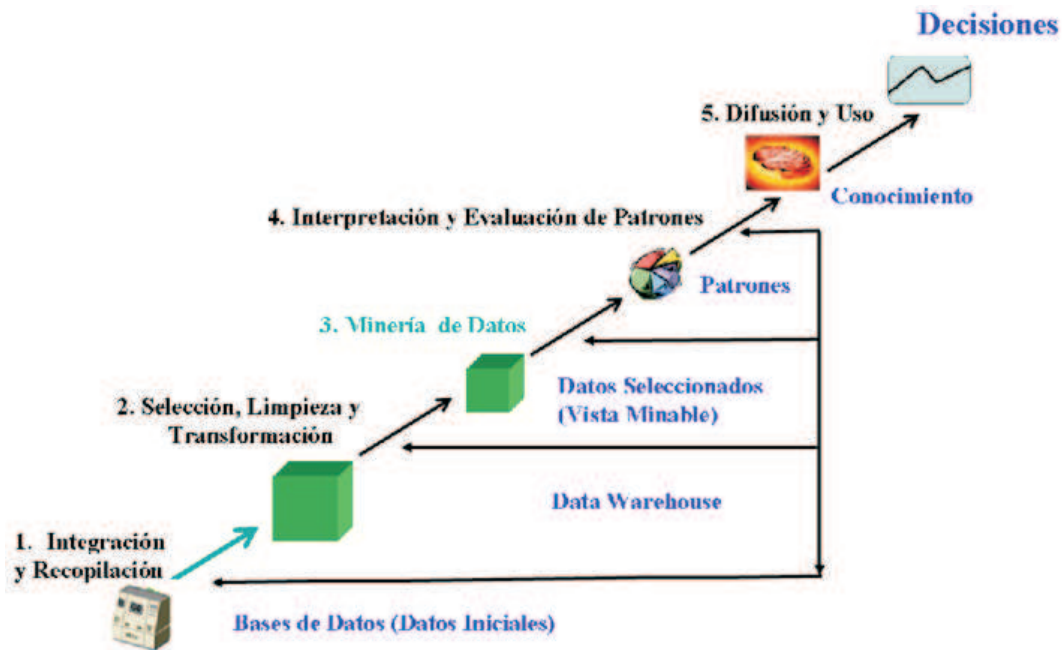


Figura 3.1: Fases del proceso de descubrimiento de conocimiento en bases de datos KDD

- **Predicción**

La meta es inducir un modelo para poder predecir o estimar, dados los valores de los atributos, el valor de salida correspondiente. Por ejemplo, en el caso de los pozos de petróleo, resulta de interés poder predecir, en base a las experiencias anteriores, si una zona dada reúne las características adecuadas para realizar una perforación o no; es decir, si se estima que se hallará petróleo o no. En el caso del alumno, podría predecirse el resultado de sus calificaciones en base a lo observado en otros alumnos con características similares.

- **Segmentación**

Estas técnicas buscan la separación de los datos en subgrupos o clases interesantes. Por ejemplo, pueden ser muy útiles para identificar, a partir de distintas mediciones de establecimientos de todo el país, la existencia de localidades con deficiencias educativas.

Es importante destacar que, independientemente del tipo de tarea a realizar, el resultado a obtener depende de observaciones realizadas previamente. Por tal motivo, resulta de fundamental interés, disponer del relevamiento adecuado a fin de poder generalizar la información de la manera adecuada.

La figura 3.2 representa el proceso ideal de Minería de Datos. Las tareas constituyen los problemas a resolver mediante técnicas de Minería de Datos, por ejemplo, “clasificar el aprendizaje de los alumnos de Computación como óptimo, bueno, regular o malo”. En este ejemplo, el tipo de tarea es la clasificación. La vista minable, por su parte, contiene solamente la información necesaria para realizar la tarea.



Figura 3.2: Proceso ideal de Minería de Datos

Definida la tarea que se desea realizar, las técnicas de Minería de Datos reciben los datos (en forma de vista minable) y producen un resultado que puede representarse en forma de árboles, reglas, etc.

Una técnica o método constituye el enfoque conceptual para extraer la información de los datos. En general es implementada por varios algoritmos. Cada algoritmo representa, en la práctica, la manera de desarrollar una determinada técnica paso a paso, de forma que es preciso un entendimiento de alto nivel de los algoritmos para saber cual es la técnica más apropiada para cada problema. También es necesario entender los parámetros y las características de los algoritmos para preparar los datos a analizar.

A través de la aplicación de las técnicas de Minería de Datos se pueden predecir hechos futuros (modelo predictivo) o mostrar nuevas relaciones entre los datos (modelo descriptivo).

3.2. Minería de Datos en Educación

La aplicación de la Minería de datos a la educación se denomina Minería de Datos Educativa (Educational Data Mining, EDM) y los datos corresponden a la información generada en entornos educativos.

El conocimiento a extraer podrá focalizarse en el alumno, el profesor o la institución y dependerá del tipo de información que se dispone (previa o posterior al curso) y del modelo educativo utilizado (educación presencial, a distancia o mixto) [27]. Si el objetivo es:

- Extraer información acerca del alumno, ésta podría ser:
 - Determinar el tipo de actividades que se ajustan mejor a su perfil.
 - Qué tipo de presentación del material didáctico le permitiría mejorar su proceso de aprendizaje.
- Extraer información útil para el profesor:
 - Detectar abandonos de alumnos a cursos.

- Cuantificar la efectividad del proceso de enseñanza-aprendizaje.
 - Clasificar o agrupar alumnos en base al perfil.
 - Corregir o mejorar la estructura del curso.
 - Evaluar las actividades realizadas en el curso en cuanto a los resultados obtenidos, a la motivación demostrada por los alumnos y al grado de dificultad.
 - Detectar los errores más frecuentes que cometen los alumnos en una actividad
- Extraer información útil para la institución:
 - Estimación del tiempo de permanencia de alumnos y profesores en una institución.
 - Evaluación del sitio web. Mejora del sitio.
 - Cuantificar la demanda educativa para su posterior mejora y adaptación a los intereses de los potenciales alumnos.
 - Demanda horaria.

Las primeras experiencias de utilización de Minería de Datos en el contexto educacional, se remontan al año 1995. Las tareas que se llevan a cabo con mayor frecuencia son: extracción de reglas de asociación, clasificación, agrupamiento o clustering y regresión. A continuación se hará una descripción de estas tareas y de las técnicas más utilizadas para resolverlas.

3.3. Tareas y Técnicas más utilizadas en Educación

3.3.1. Predicción utilizando Reglas de Asociación

Las reglas de asociación son una de las herramientas más utilizadas de la Minería de Datos por su capacidad para determinar correlaciones entre los atributos de una Base de Datos. Su simplicidad de expresión las convierte en un mecanismo ideal para modelizar la información disponible.

Una regla de asociación tiene la forma “Si X ENTONCES Y” y posee ciertos valores que permiten estimar su calidad. Los más utilizados suelen ser el Soporte y la Confianza.

- El **Soporte** o Cobertura es el número de instancias que la regla predice correctamente y se expresa en proporción a la cantidad de ejemplos totales disponibles.
- La **Confianza** o precisión es el porcentaje de veces que la regla se cumple cuando se puede aplicar.

Por ejemplo, si la siguiente regla posee un soporte del 40 % y una confianza del 66 %

Si los estudiantes obtienen ‘‘incorrecto en X’’ ENTONCES
también obtienen ‘‘incorrecto en Y’’

debe entenderse que el 40 % del total de los alumnos verifican esta regla (soporte) y además, el 66 % de los estudiantes que obtienen ‘‘incorrecto en X’’ también obtienen ‘‘incorrecto en Y’’ (confianza).

Dada una base de datos, el proceso habitual que permite el *aprendizaje de las Reglas de Asociación* consiste en extraer todas las reglas de asociación que tengan un soporte y una confianza por encima de umbrales establecidos por el usuario. Este problema de extracción de reglas de asociación se suele descomponer en dos subproblemas:

- La obtención de los conjuntos de ítems cuyo soporte es mayor que el soporte mínimo requerido, también denominados ítems frecuentes.
- La generación de reglas a partir de estos ítems.

Un algoritmo ‘‘clásico’’ para la obtención de reglas de asociación es el algoritmo ‘‘A Priori’’, presentado por primera vez en [3]. Se caracteriza por realizar varias pasadas sobre los datos con el fin de encontrar los ítems frecuentes. A partir de este algoritmo básico se desarrollaron numerosas variantes y actualmente sigue siendo motivo de investigación. La eficiencia en la búsqueda de las reglas de asociación es un campo importante de la Extracción de conocimiento en bases de datos o KDD.

Como ejemplo de aplicación y de línea de investigación en el descubrimiento de reglas de asociación en sistemas de e-learning que utilizan algoritmos basados en el algoritmo A-Priori se pueden mencionar:

- Agathe Mercaron y Kalina Yacef [23] utilizaron reglas de asociación para hallar errores que ocurren al mismo tiempo durante la resolución de ejercicios. De esta manera, a partir del conocimiento obtenido, el docente puede rediseñar el material del curso. En este caso se empleó una variante del algoritmo A priori estándar que tiene en cuenta el orden en el que ocurren los atributos.

Un caso especial de reglas de asociación, que recibe el nombre de reglas de asociación secuenciales, se utiliza para determinar patrones secuenciales en los datos. En este tipo de reglas las relaciones entre los datos se basan en el tiempo. Los algoritmos más populares para el aprendizaje de estas reglas, son el algoritmo AprioriAll [1] y el algoritmo Generalized Sequential Patterns (GSP) [2].

Primer TP	Actividad Virtual	Actividad Presencial	Resultado 1er Parcial
Bien	Alta	Alta	Aprobó
Bien	Baja	ALta	No Aprobó
Muy Bien	Alta	Alta	Aprobó
Regular	Alta	Alta	Aprobó
Regular	Alta	Baja	No Aprobó
Regular	Baja	Baja	No Aprobó
Muy Bien	Baja	Baja	Aprobó
Bien	Baja	Baja	No Aprobó
Bien	Alta	Baja	Aprobó
Regular	Baja	Baja	No Aprobó
Bien	Alta	Baja	Aprobó
Muy Bien	Alta	Alta	Aprobó
Regular	Baja	Baja	Aprobó
Regular	Alta	Alta	Aprobó

Cuadro 3.1: Muestra de Base de Datos de Alumnos

Como ejemplos de aplicaciones y de líneas de investigación en análisis de secuencias se tiene:

- O. Zaine y J. Luo [40], para evaluar las actividades on-line de los estudiantes y personalizar el envío de recursos.
- W. Wang, J. Sun y S. Tseng [36] para ayudar a los profesores a comprender los motivos del buen aprendizaje o no de los alumnos.

3.3.2. Clasificación

Las técnicas de clasificación pueden utilizarse tanto para realizar predicción como para caracterizar o modelizar la información disponible. Su objetivo es identificar y representar las similitudes existentes entre los datos.

Por ejemplo, sea una Base de Datos como la de la tabla 3.1 en la que se recogen los datos relativos a la actuación de los alumnos en un curso de Computación hasta el Primer Parcial Práctico. Cada fila de la Base de Datos representa una instancia.

“Resultado Primer Parcial” es un atributo que se denomina clase de la instancia y representa el resultado obtenido por el alumno en el Primer Parcial. Puede adoptar los valores “Aprobó” o “No Aprobó”. Los atributos “Actividad Virtual” y “Actividad Presencial” pueden tomar los valores “Alta” o “Baja”.

Ejemplos de técnicas que se utilizan para llevar a cabo la tarea de clasificación son:

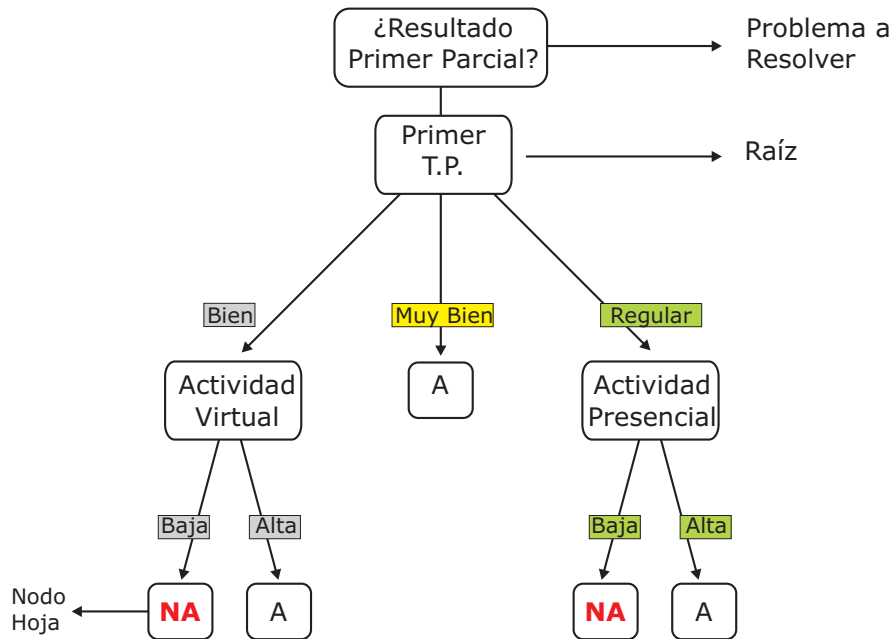


Figura 3.3: Ejemplo de Arbol de Decisión

Arboles de Decisión o de Clasificación

Conforman una serie de decisiones o condiciones organizadas en forma jerárquica a modo de árbol [29] mediante la cual es posible englobar condiciones. Esta técnica se caracteriza por su facilidad de uso y de interpretación pero presenta cierta inestabilidad ante variaciones de la muestra. Utilizando un algoritmo de aprendizaje de árboles de decisión, se podría obtener el árbol que se muestra en la figura 3.3. Esta estructura permite clasificar los ejemplos existentes según el resultado del primer parcial, donde A representa Aprobado y NA representa No Aprobado. Nótese que el árbol, se construye a partir de los ejemplos existentes convirtiéndose en una herramienta de modelización de la información y de predicción, ya que puede presentársele nuevos ejemplos para los cuales podrá indicarse el resultado “esperado” del primer parcial.

Los árboles de decisión se pueden considerar como una forma de aprendizaje de reglas del tipo IF-THEN, ya que cada rama del árbol puede interpretarse como una regla, donde los nodos internos en el camino desde la raíz a las hojas definen las condiciones del antecedente de la regla, y la clase asignada en el nodo hoja es el consecuente.

Ejemplo:

```

IF PrimerT.P. = Muy Bien THEN ResultadoPrimer Parcial = Aprobado
IF (PrimerT.P. = Bien) AND (ActividadVirtual = Baja)
    THEN ResultadoPrimerParcial= No Aprobado
IF (PrimerT.P.= Regular) AND (ActividadPresencial = Alta)

```


THEN ResultadoPrimerParcial=Aprobado

Uno de los métodos más utilizados para construir el árbol es ID3 y su variante C4.5. En ambos casos se utiliza el principio de “divide y vencerás” ya que durante la construcción de la estructura se trata de identificar y ubicar en la parte superior del árbol a los atributos que mejor separan los ejemplos o muestras de la Base de Datos. Existen distintas versiones de estos algoritmos que implementan diferentes estrategias de poda sobre el árbol de manera de reducir la estructura preservando una cota de precisión, en la clasificación, aceptable.

Clasificador Bayesiano

Este tipo de clasificador se basa en el teorema de Bayes. Este teorema, permite actualizar la creencia que si tiene de un suceso o conjunto de sucesos en base a nuevos datos u observaciones. Es decir, que permite pasar de la probabilidad a priori $P(\text{suceso})$ a la probabilidad a posteriori $P(\text{suceso}/\text{observaciones})$. La ecuación 3.1 representa el teorema de Bayes:

$$p(H/O) = \frac{p(O/H) \cdot p(H)}{p(O)} \quad (3.1)$$

donde aparecen las probabilidades a priori de la hipótesis H y de las observaciones O , $P(H)$ y $P(O)$ respectivamente y las probabilidades condicionadas $P(H/O)$ y $P(O/H)$.

Si se trata de estimar la probabilidad de que una determinada muestra pertenezca a la clase C en base a un conjunto de variables predictoras o atributos A_1, \dots, A_n , el teorema de Bayes toma la forma de la ecuación 3.2

$$p(C/A_1, \dots, A_n) = \frac{p(A_1, \dots, A_n/C) \cdot p(C)}{p(A_1, \dots, A_n)} \quad (3.2)$$

Evidentemente, si C tiene k valores posibles c_1, \dots, c_k , lo que interesa es encontrar el más plausible y devolverlo resultado de la clasificación. En el marco bayesiano, la hipótesis más plausible es precisamente la que tiene máxima probabilidad a posteriori dados los atributos y es conocida como la hipótesis máxima a posteriori o hipótesis MAP (*maximum a posteriori*). De esta forma, el valor a devolver será:

$$c_{MAP} = \max_{c \in \Omega_C} p(c|A_1, \dots, A_n) = \max_{c \in \Omega_C} \frac{p(A_1, \dots, A_n/c) \cdot p(c)}{p(A_1, \dots, A_n)} = \max_{c \in \Omega_C} p(A_1, \dots, A_n/c) \cdot p(c) \quad (3.3)$$

donde Ω_C representa el conjunto de valores que puede tomar la variable C . Nótese que en el último paso se ha eliminado la división ya que el divisor es el mismo para todas las

categorías o clases.

Si bien el teorema de Bayes brinda un método claro y sencillo para implementar el clasificador, posee un costo computacional importante ya que es preciso trabajar con distribuciones de probabilidad que involucran muchas variables. Esto puede simplificarse suponiendo que dichas variables son independiente, dando lugar a lo que se conoce como Naive Bayes.

El fundamento principal del clasificador Naive Bayes [10] [31] es la suposición de que todos los atributos son independientes conocido el valor de la variable clase. En base a esto, la expresión 3.3 puede escribirse como

$$c_{MAP} = \max_{c \in \Omega_C} p(A_1, \dots, A_n/c) \cdot p(c) = \max_{c \in \Omega_C} p(C) \prod_{i=1} p(A_i/c) \quad (3.4)$$

A continuación se ejemplifica el uso del clasificador Naive Bayes para clasificar un nuevo ejemplo E para la Base de Datos de la tabla 3.1

Primer T.P.	Actividad Virtual	Actividad Presencial	Resultado 1er.Parcial
Regular	Baja	Alta	?

En dicha tabla hay sólo dos clases posibles “Aprobado” o “No Aprobado”. Por lo tanto, deberán calcular los términos de la ecuación 3.4 para cada una de ellas y tomar el mayor valor.

Para la clase “Aprobado” condicionada por los valores de la combinación de atributos de “Primer TP”, “Actividad Virtual” y “Actividad Presencial” del ejemplo E , será:

$$\frac{P(\text{PrimerTP} = \text{Reg/Aprobo}) \cdot P(\text{AV} = \text{Baja/Aprobo}) \cdot P(\text{AP} = \text{Alta/Aprobo})}{P(E)}$$

Reemplazando valores

$$P(\text{Aprobo}/E) \cdot P(E) = \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{5}{9} = 0,0274$$

Siguiendo el mismo procedimiento para la clase “No Aprobó”, se obtiene:

$$P(\text{NoAprobo}/E) \cdot P(E) = \frac{3}{5} \cdot \frac{4}{5} \cdot \frac{1}{5} = 0,096$$

Por lo tanto, como $P(\text{NoAprobo}/E) \cdot P(E)$ es mayor que $P(\text{Aprobo}/E) \cdot P(E)$ se asignará el valor “No Aprobó” al atributo “Resultado Primer Parcial” del dato E .

Expresando el resultado en forma de regla, se obtiene:

```
IF(PrimerT.P.= Regular) AND (ActividadVirtual = Baja) AND
  (ActividadPresencial = Alta)
  THEN ResultadoPrimerParcial= No Aprobado
```

Como ejemplos de aplicaciones y de líneas de investigación en la tarea de Clasificación se pueden mencionar:

- Agathe Mercaron y Kalina Yacef [24], para predecir la actuación que tendrá un alumno en el examen final a partir de los trabajos desarrollados en el curso. Utilizan el algoritmo C4.5 de construcción de árboles de decisión.
- Mihaela Cocea y Setephan Weibelzahl [7], para predecir el nivel de motivación de los alumnos en sistemas de e-learning de acuerdo a las actividades desarrolladas. Utilizan el algoritmo C4.5.
- Chen, Liu y Ou [9], para descubrir grupos potenciales de estudiantes con características similares para definir una determinada estrategia pedagógica. Utilizan los algoritmos C4.5 y C5.0.

3.3.3. Agrupamiento (clustering)

Es una tarea descriptiva que consiste en obtener grupos “naturales” a partir de los datos. La agrupación de los datos se basa en el principio de maximizar la similitud entre los elementos de un grupo minimizando la similitud entre los diferentes grupos. La similitud de los datos es una medida del grado de proximidad. Al agrupamiento también se lo denomina segmentación, ya que parte o segmenta los datos en grupos que pueden ser o no disjuntos.

El clustering se puede utilizar, por ejemplo, en un modelo de educación a distancia, para identificar grupos de estudiantes en base a sus estilos cognitivos. Esto le permitiría al sistema ofrecer una enseñanza más personalizada, procurando ofrecer a cada grupo el material didáctico adaptado a las habilidades cognitivas.

Los algoritmos que se aplican a este tipo de tarea pueden ser Jerárquicos o No Jerárquicos. Los algoritmos jerárquicos consisten en crear grupos que están jerárquicamente relacionados con otros grupos. Algunos ejemplos de estos algoritmos son “single-link” y “complete link”. Los algoritmos no jerárquicos consisten en particionar los datos en un número prefijado de grupos K al tratar de minimizar una determinada función objetivo. Como ejemplos de esta técnica se pueden citar K-medias y expectation maximization.

Como ejemplos de aplicaciones y de líneas de investigación en la tarea de Clustering se tienen:

- Talavera y Gaudioso [39], para analizar los datos obtenidos de cursos basados en sistemas e-learning y detectar patrones que reflejen comportamientos análogos en los

usuarios con el objetivo final de mejorar la actividad en los espacios de colaboración. Utilizan técnicas de clustering similares al modelo de Naive Bayes.

- Julia Miguillón y Enric Mor [25], para agrupar estudiantes para establecer itinerarios de educación personalizados.
- Sebastián Ventura Soto, para agrupar estudiantes de acuerdo a las actividades realizadas en un curso desarrollado sobre la Plataforma Moodle y para determinar la relación existente entre esos grupos y la calificación final. La información recolectada fue procesada utilizando una herramienta de Minería de Datos de acceso libre denominada Weka. La aplicación del algoritmo K-medias sobre el conjunto de datos les permitió obtener tres grupos de características diferentes.

3.3.4. Regresión

Es una tarea predictiva y consiste en aprender una función real que asigna a cada instancia de la base de datos un valor real. Se basa en supuestos estadísticos. El valor a predecir es numérico. El objetivo es minimizar el error entre el valor predictivo y el valor real.

La regresión se puede utilizar, por ejemplo, para predecir las respuestas de alumnos sobre un módulo, en base a la actuación del alumno en evaluaciones previas, a la complejidad de las preguntas y al contenido de su aprendizaje [12].

Como técnica para resolver este tipo de tareas se pueden utilizar los árboles de decisión que, si tienen por objetivo predecir variables continuas se denominan árboles de regresión.

3.4. Selección de una Técnica de Minería de Datos

La correspondencia entre tareas y técnicas es variada. Algunas técnicas permiten resolver distintas tareas y otras permiten resolver solamente un tipo de tarea. Debido a esta variedad, es necesario conocer las características funcionales de cada técnica: capacidad, ámbitos donde suele funcionar mejor, la eficiencia, la robustez, etc.

Cada método permite expresar mejor cierto tipo de patrones. Se define como *expresividad* a la forma en que se indican los patrones aprendidos.

Se dice que un método que construye un modelo es comprensible cuando es sencillo examinar y comprender el conocimiento obtenido.

En el Anexo B se grafican las tareas de Minería de Datos y las técnicas que permiten abordarlas.

3.5. La Computación Evolutiva

Otra técnica que se utiliza en problemas de clasificación, regresión, agrupamiento y reglas de asociación es la basada en Algoritmos Genéticos y Evolutivos. Nos detendremos a hacer una breve descripción de los fundamentos de la Computación Evolutiva (CE).

La CE trata de resolver problemas tomando como referencia el proceso de la evolución natural. Esta es la razón por la cual se utilizan términos como: individuo, cromosoma, mutación, cruce, fenotipo y genotipo entre otros.

Si se piensa en una población de individuos situados en un contexto, el grado de adaptación de cada individuo a su entorno establece la probabilidad de que pueda sobrevivir y reproducirse. Con el transcurso del tiempo, a través de sucesivas generaciones, el grado de adaptación de la población en general y del mejor de los individuos, aumentará.

Si ahora se considera que un individuo es una solución a un problema, que el entorno es el problema, que la población es un conjunto de soluciones candidatas a resolver un problema y que el grado de adaptación es la calidad de una solución, se podría plantear: *Si se piensa en un conjunto de soluciones a un problema, la calidad de cada solución a un determinado problema establece la probabilidad de que pueda guardarse y utilizarse como semilla para construir nuevas soluciones candidatas. Con el transcurso del tiempo, a través de sucesivas iteraciones, la calidad del conjunto de soluciones en general y de la mejor solución, en particular, aumentará [37]*

La figura 3.4 presenta un esquema de las áreas que comprende la Computación Evolutiva.

Los Algoritmos Genéticos consituyen una de las áreas de la Computación Evolutiva más utilizadas para resolver el problema de descubrimiento de reglas.

3.5.1. Algoritmos Genéticos

El uso de Algoritmos Genéticos en el descubrimiento de reglas de predicción se caracteriza por trabajar con una población de soluciones sobre las que se aplican un conjunto de operadores genéticos. Existen dos enfoques para abordar el descubrimiento de reglas utilizando algoritmos genéticos. Su diferencia fundamental radica en la forma en que las reglas se representan en la población:

1. Cada individuo de la población contiene en su cromosoma al conjunto de reglas completo (enfoque Pittsburg)
2. Cada individuo de la población representa a una única regla y la población completa busca dar respuesta al problema (enfoque Michigan)
3. Cada individuo de la población representa a una única regla y el algoritmo genético debe ser aplicado varias veces para obtener la solución del problema. Por cada aplicación, el mejor individuo obtenido será una de las reglas del conjunto de reglas buscado (enfoque IRL – Iterative Rule Learning)

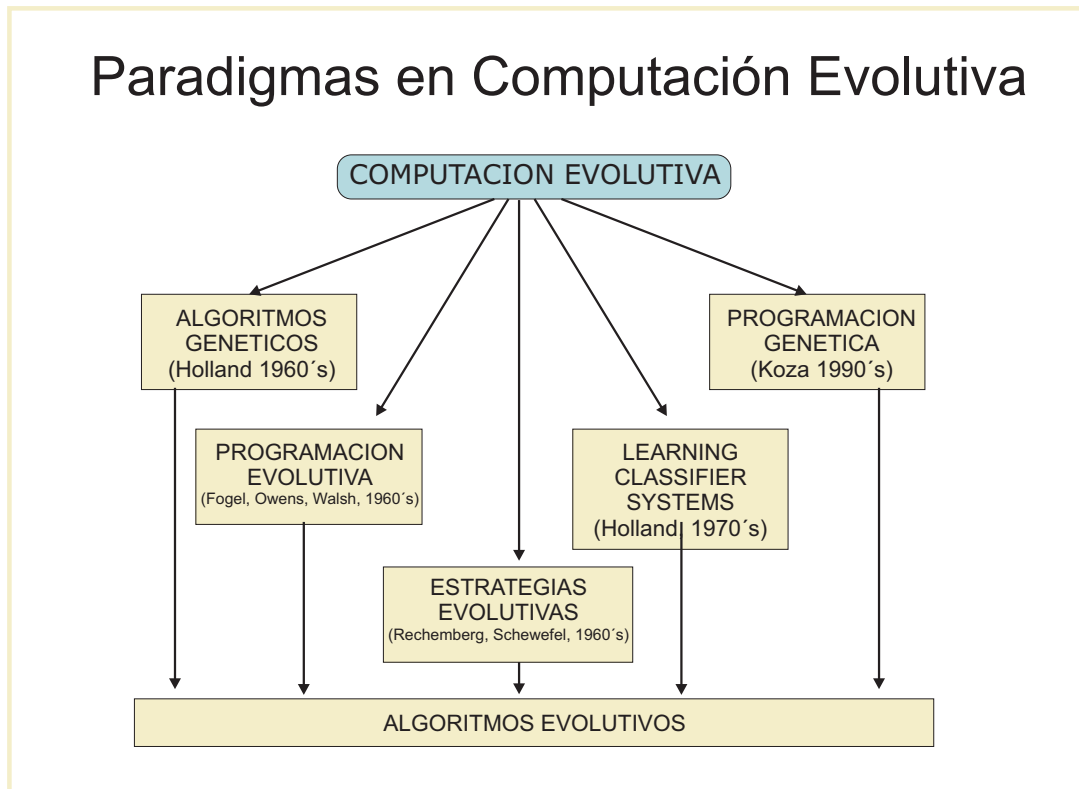


Figura 3.4: Computación Evolutiva

El enfoque Pittsburg presenta la ventaja de poder evaluar el funcionamiento del conjunto de reglas completo ya que el mismo se encuentra contenido en el cromosoma del individuo. Como contrapartida, las reglas que se obtienen con este enfoque son de menor calidad que las que se pueden obtener con los otros dos. Además, los cromosomas de la población presentan mayor longitud incrementando de esta forma el tiempo computacional del algoritmo y requiriendo un mayor esfuerzo a la hora de definir los operadores genéticos.

Por su parte los enfoques Michigan e IRL, que representan una única regla en cada cromosoma, poseen individuos más cortos con operadores genéticos más simples. Como contrapartida, resulta más difícil evaluar la aptitud de cada regla en la resolución del problema y suelen perder la diversidad de sus individuos con facilidad lo que generalmente obliga el uso de nichos.

Como puede deducirse, la elección del método de representación de las reglas dependerá del tipo de problema a resolver. En el caso de la obtención de reglas de clasificación uno de las estrategias más sencillas es el enfoque IRL.

Si se trata de obtener reglas de clasificación, otra decisión a tomar con respecto a la representación es si el consecuente de la regla va a ser codificado o no dentro del cromosoma. En caso de no hacerlo, puede ejecutarse el algoritmo genético tantas veces como clases haya a fin de obtener las reglas que mejor cubren los distintos grupos de ejemplos.

Una vez representadas las reglas, es preciso establecer las estrategias de selección y reemplazo de individuos dentro de la población. La obtención del conjunto de reglas será un pro-

ceso iterativo que permitirá establecerlas luego de analizar reiteradamente su desempeño sobre el conjunto de ejemplos disponibles.

3.6. Software de Minería de Datos

Se pueden encontrar en ámbitos comerciales y académicos entornos de software diseñados para dar soporte a la Minería de datos.

Los paquetes integrados contienen en un mismo entorno herramientas para el preprocesado de datos, diferentes modelos de análisis, diseño de experimentos y soporte gráfico para la visualización de resultados.

Son ejemplos de paquetes integrados: SPSS Clementine, WEKA, Kepler, DB2 Intelligent Miner, ODMS, DBMiner, YALE, SAS Enterprise Miner, STATISTICA y Data Miner.

Cada una de estas herramientas requiere determinados tipos de archivos de entrada.

En el Anexo A se detallan las características de cada una de ellas.

Capítulo 4

Propuesta de cambio en la Estrategia de Enseñanza

Este capítulo se describen las experiencias obtenidas a partir de la metodología actual y se detalla la propuesta de cambio en la estrategia de enseñanza en la asignatura Computación.

4.1. Metodología utilizada actualmente

Pimenta Garrido [14] afirma que “la construcción de una teoría didáctica desde la práctica exige considerar el triángulo didáctico en situación, es decir, los contextos socioteóricos históricos en donde se lleva a cabo dicha práctica”; dicho Triángulo Didáctico se encuentra formado por los procesos de Enseñanza-Aprendizaje [8] como lo representa la figura 4.1.

Si a este modelo se le incorpora el concepto de Transposición Didáctica (proceso que realiza el docente para transformar el conocimiento académico en conocimiento didáctico), el conocimiento/contenido se desdobra en “conocimiento epistemológico” y “conocimiento a transmitir o conocimiento didáctico” como lo representa la figura 4.2.

El proceso que vehiculiza los contenidos a través de la tecnología, y por lo tanto necesita relacionarlos con los atributos de cada medio en particular y con el uso pedagógico de estos medios, se denomina mediación [41].

Los medios que se utilizan actualmente en la asignatura Computación son de tipo informático e impreso.

En la facultad de Ingeniería durante el segundo cuatrimestre del año 2006 y el primer cuatrimestre del año 2007, se realizaron pruebas piloto con asignaturas del Departamento de Matemática en las que se utilizó una plataforma de enseñanza basada en web (plataforma

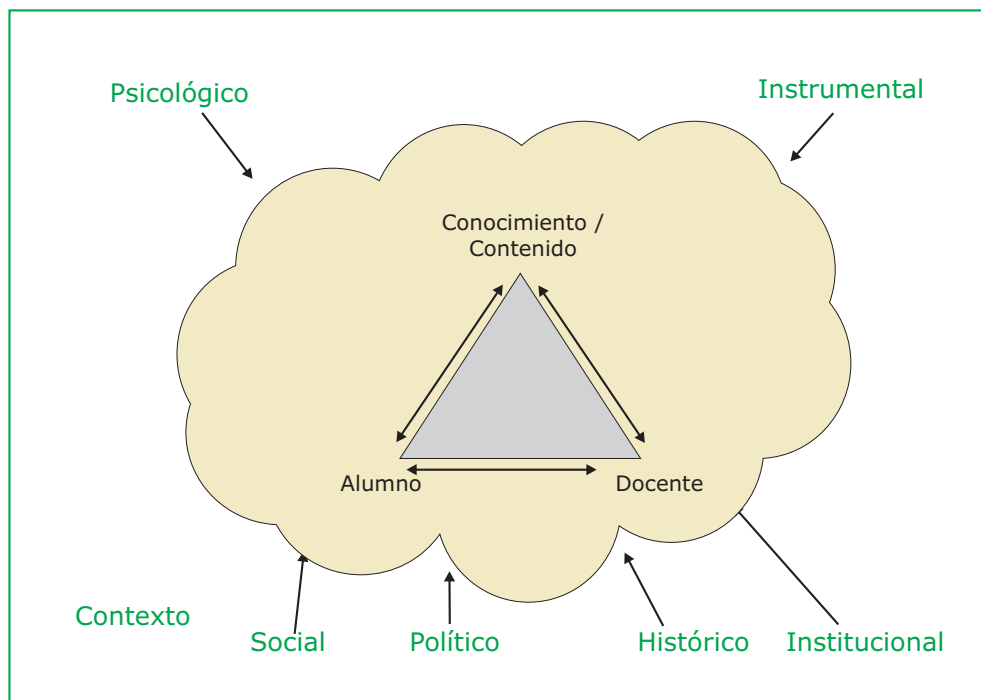


Figura 4.1: Triángulo didáctico

de e-learning).

Se considera una plataforma de e-learning a aquellas herramientas que combinan hardware y software para ofrecer todas las prestaciones necesarias para la formación basada en la Red [35].

En particular, en la facultad se está tratando de implantar la plataforma LRN (Learning Resources Nterchange).

En el segundo cuatrimestre del año 2007, el área de Servicios de Computación de la Facultad planea incrementar la cantidad de cursos que utilizan la plataforma. El sistema permite mediar diferentes aspectos del proceso de enseñanza y de aprendizaje en el que se involucran alumnos y docentes. Cuenta con herramientas para la comunicación sincrónica y asincrónica, herramientas para la gestión de materiales de aprendizaje y herramientas para la gestión, seguimiento y evaluación de los estudiantes.

4.2. Modificaciones propuestas

El inconveniente de las plataformas es que no poseen la capacidad de brindar a cada estudiante un tratamiento personalizado del contenido didáctico.

Los Sistemas Hipermedia Adaptativos basados en Web permiten solucionar este problema. Estos sistemas pueden

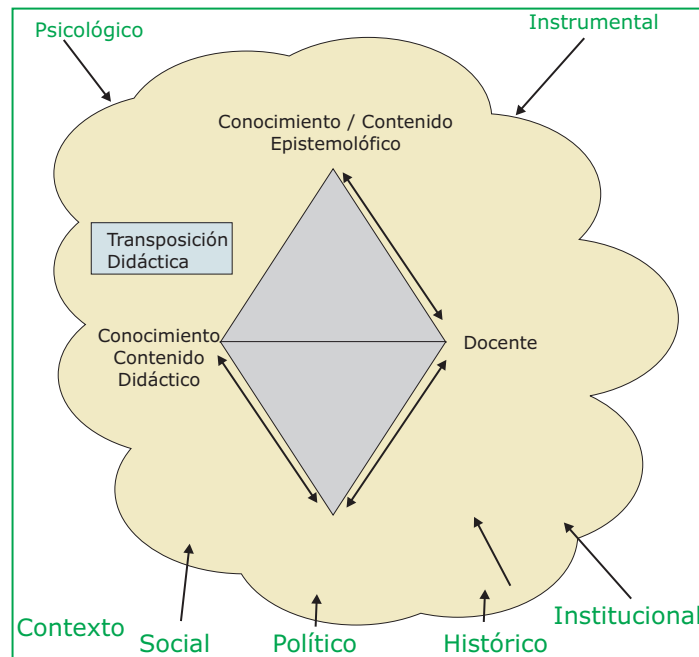


Figura 4.2: Incorporación del Contenido Didáctico al Triángulo Didáctico

- monitorear las actividades de los usuarios,
- interpretarlas mediante modelos específicos del dominio,
- inferir los requisitos del usuario y sus preferencias a partir de las actividades interpretadas,
- actuar a partir del conocimiento disponible sobre los usuarios y el dominio

para facilitar el proceso de aprendizaje en forma dinámica [33].

En la figura 4.3 se representan los procesos que se llevan a cabo en una aplicación adaptativa: La utilización de un sistema adaptativo en el desarrollo de un curso permite adaptar el contenido a todos y cada uno de los usuarios que lo ejecuten de forma personalizada. Si a la información recolectada del entorno educativo se le aplican técnicas de Minería de Datos el docente puede reestructurar los contenidos del sitio web para personalizar los cursos, con el objetivo final de mejorar los procesos de enseñanza-aprendizaje. En la figura 4.4 se representa la relación entre profesores, estudiantes, entorno educativo y minería de datos.

Teniendo en cuenta que las falencias observadas en el aprendizaje de los alumnos son:

- Bajo porcentaje de alumnos que aprueban la cursada (aproximadamente un 30 % del total de inscriptos).
- Bajo porcentaje de consultas (mejorado a través de la asesoría virtual).

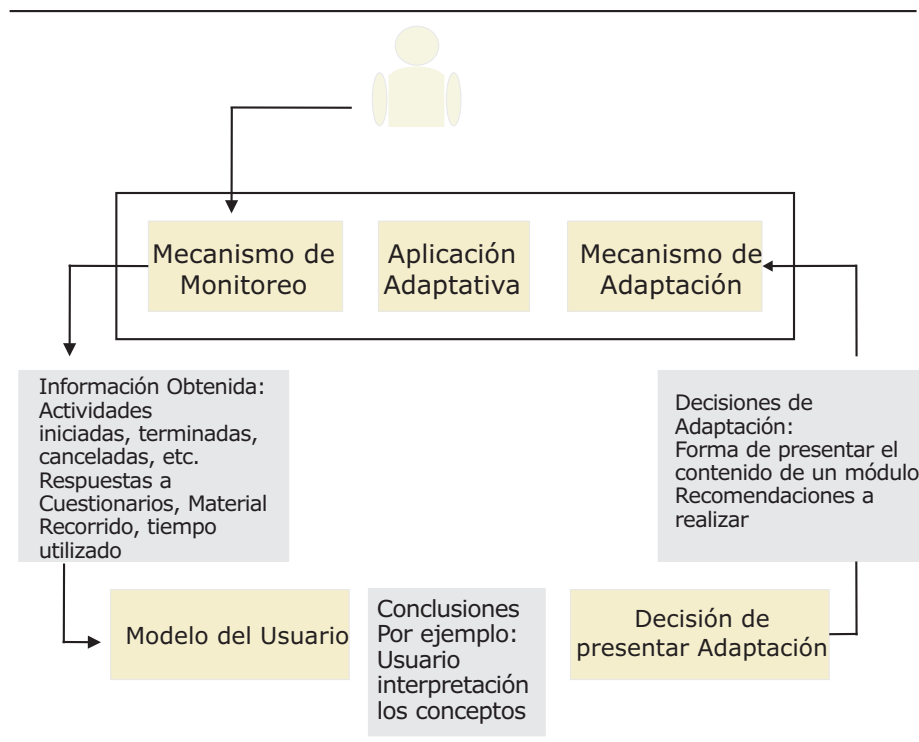


Figura 4.3: Aplicación Adaptativa

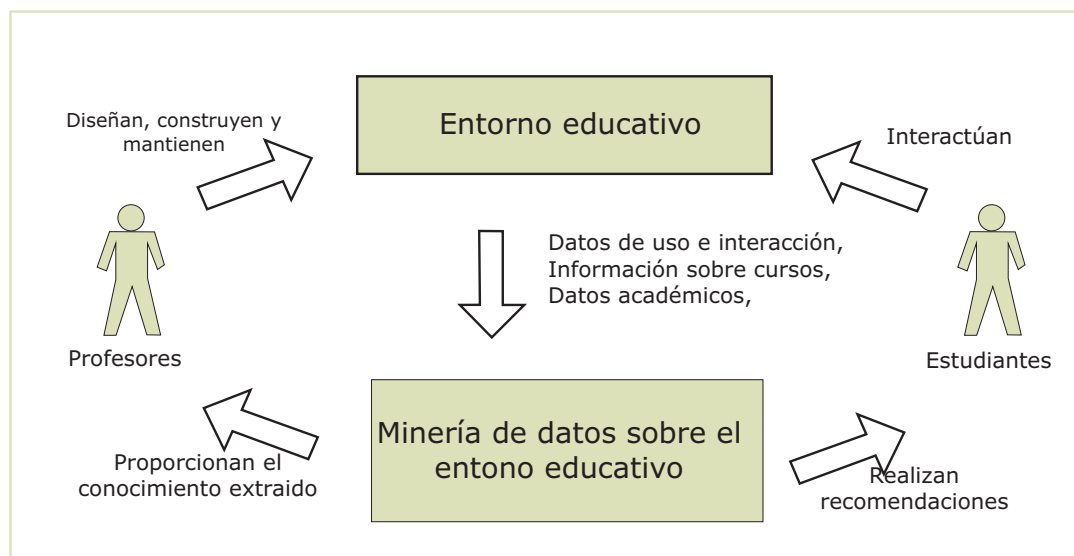


Figura 4.4: Relación entre profesores, estudiantes, entorno educativo y Minería de Datos

- Elevado porcentaje de alumnos que no alcanzan a cumplimentar la totalidad de instancias de evaluación (aproximadamente un 30% del total de inscriptos)

en el presente trabajo se analiza un cambio en la mediación que permita llevar a cabo un mayor seguimiento del proceso de enseñanza-aprendizaje y que trate de impulsar un tipo de enseñanza centrada en el estudiante ampliando el tiempo dedicado a actividades participativas.

En el siguiente capítulo se parte del análisis del rol del docente como diseñador y constructor de un Curso Hipermedia Adaptativo basado en Web para luego hacer una propuesta de diseño de un curso de computación bajo esta modalidad. Este desarrollo tiene en cuenta la interacción que puede tener el alumno con el sistema de acuerdo a su perfil, y describe la clase de información útil que se desea obtener para luego llevar a cabo los cambios pertinentes para realimentar el proceso de enseñanza-aprendizaje.

Capítulo 5

Propuesta de Diseño de un curso de Computación

Para el diseño y la construcción de un sistema hipermedia adaptativo basado en Web, se deberán tomar decisiones docentes

- Propias de un curso presencial
 - determinar cuáles son los principios básicos de la materia,
 - seleccionar y organizar el contenido (conceptos, procedimientos y actitudes) a enseñar.
 - diseñar las actividades que se van a proponer a los estudiantes para cada contenido.
 - diseñar la evaluación.
- Propias de un sistema adaptativo
 - cuántos grados de dificultad debe tener cada tema,
 - qué conceptos, dentro de un tema, se van a colocar en cada grado de dificultad.

A continuación, se plantea una estrategia para el diseño de un curso de computación basada en el sistema AHA! y en sus mejoras llevadas a cabo en la Universidad de Córdoba por Cristóbal Romero.

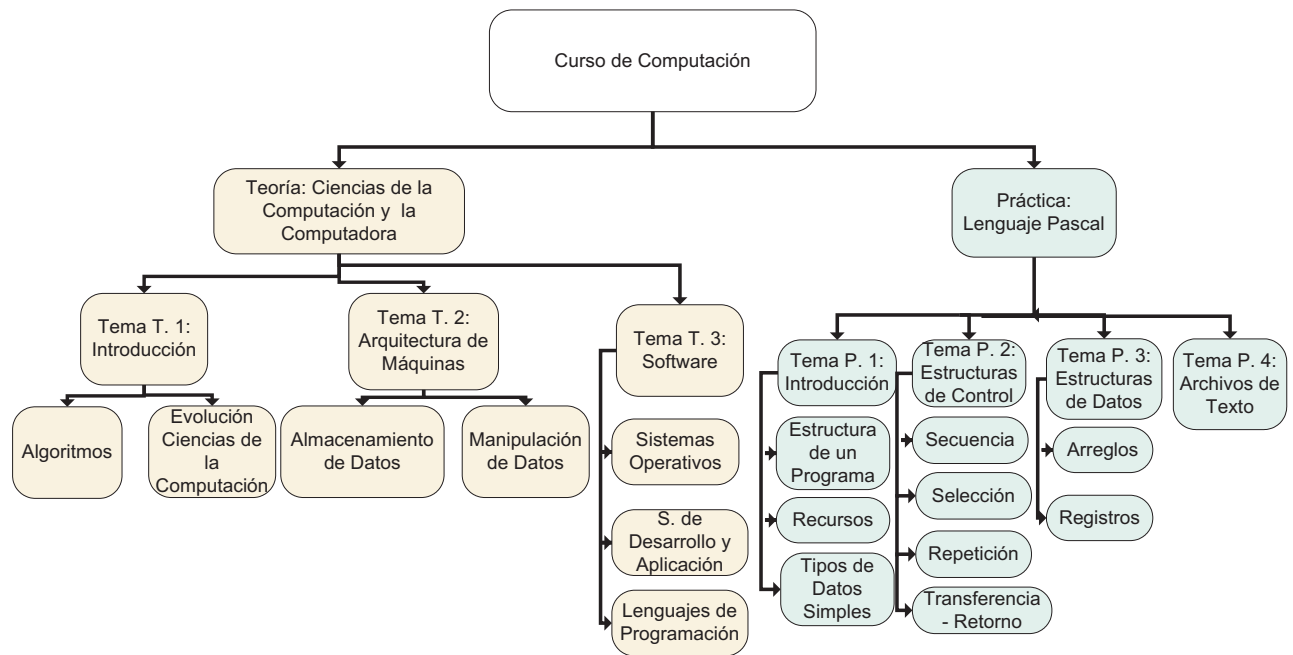


Figura 5.1: Estructura del Curso de Computación

5.1. Diseño del Modelo Adaptativo

5.1.1. Módulo del Dominio

Se propone estructurar el Curso de Computación en base a temas, cada uno de los cuales estará formado por conceptos.

La figura 5.1 ilustra una posible organización. En dicha figura, el tema *P,2.* corresponde a la práctica de Estructuras de Control y se encuentra formado por varios conceptos: Secuencia, selección, repetición, etc. A su vez, estos conceptos contienen sub-conceptos.

Un concepto va a estar formado por una o varias páginas Web relacionadas. Cada página Web va a ser un archivo XML (eXtensible MarkupLanguage).

Además, cada tema tendrá tres grados de dificultad: alto, medio y bajo y los conceptos de cada tema se abordarán desde tres niveles de complejidad. Los archivos que componen este módulo podrán contener




- desarrollos teóricos de conceptos,
- test iniciales sobre temas
- actividades sobre conceptos,
- test final sobre un tema

A continuación se ejemplifica la presentación del concepto Algoritmo según el nivel del usuario:

Recordando la definición de Algoritmo:

Conjunto finito de instrucciones ejecutables, no ambiguas, que dirige a una actividad que termina o resuelve un problema en tiempo limitado.

Veamos algunos ejemplos:








Haz clic en cada imagen para conocer su descripción.

Recordando la definición de Algoritmo:

Conjunto finito de instrucciones ejecutables, no ambiguas, que dirige a una actividad que termina o resuelve un problema en tiempo limitado.

Veamos algunos ejemplos:

Preparar una receta de cocina.

Levantar una pared de ladrillos.

Sumar dos números.

Haz clic en cada imagen para conocer su descripción.

Analicemos el siguiente problema:

Se desea cambiar la rueda de un auto. Se supone que el auto está detenido en un lugar seguro y que se cuenta con todos los elementos para resolver el problema. Se considera también que la persona que debe cambiarla (interlocutor) no tiene ningún conocimiento sobre el tema.

En el video se observan:

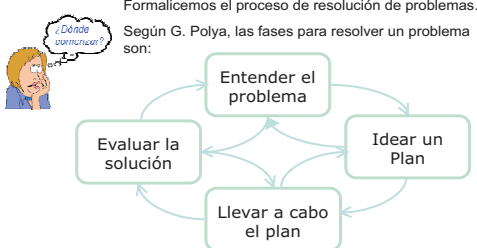
<p>Recursos</p> <ul style="list-style-type: none"> • Auto • Criquet • Rueda pinchada • Rueda de auxilio • Tuercas: • Balizas 	<p>Interlocutor</p> <ul style="list-style-type: none"> • Persona que cambia la rueda <p>Acciones</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> • Quitar la rueda pinchada
---	--

[Video que enseña a cambiar la rueda de un auto](#)

El video presenta un "método para cambiar una rueda pinchada". Se trata de un algoritmo.

Formalicemos el proceso de resolución de problemas.

Según G. Polya, las fases para resolver un problema son:



La resolución de un problema implica un proceso dinámico en el que se deben repasar una y otra vez las fases en forma evolutiva retrocediendo y avanzando hasta que la solución sea satisfactori a. El orden de ejecución de las fases puede variar.

Figura 5.2: Nivel Principiante de presentación del concepto algoritmo

- Para un usuario principiante (figura 5.2)
- Para un usuario de nivel medio (figura 5.3)
- Para un usuario experto (figura 5.4)

5.1.2. Módulo del Alumno

El Módulo del Alumno contiene información sobre cada alumno. Esta información incluye preferencias y objetivos del alumno e información generada durante su interacción con el sistema.

En la tabla 5.1 se presenta un ejemplo con Niveles de Conocimiento de Temas y Conceptos aplicada al desarrollo del Curso.

En cada celda se colocan los atributos (leído, nivel de conocimiento). El atributo "leído" varía entre 0 y 1 (0: no leído, 1: leído) y el atributo "nivel de conocimiento" varía entre 0 y 100 (a mayor nivel de conocimiento, mayor puntuación).

Por ejemplo, según la información que aparece en la tabla 5.1, el alumno Alejandro Acosta ha leído los conceptos del tema "Estructura deControl" y posee un nivel de conocimiento intermedio (valor 50).

Un algoritmo se puede definir como:

Conjunto finito de instrucciones ejecutables, no ambiguas, que dirige a una actividad que termina o resuelve un problema en tiempo limitado.

Veamos un ejemplo: El algoritmo euclidiano para hallar el máximo común divisor de dos enteros positivos.

M, N y R designan valores enteros positivos (los valores asociados a estos nombres cambiarán durante la ejecución del algoritmo). Luego, sigamos los siguientes pasos:

1. Obtener el número mayor y llamarlo M.
2. Obtener el número menor y llamarlo N.
3. Dividir M entre N y asignar el valor del resto a R.
4. Si R es 0 seguir por el paso 8.
5. Asignar a M el valor de N.
6. Asignar a N el valor de R.
7. Seguir por el paso 3.
8. El máximo común divisor de los valores originales de M y N es el actual valor de N.
9. Terminar.

Para crear un algoritmo básicamente hay que:

1. Plantear bien el problema a resolver.
2. Establecer los recursos que se necesitarán para resolverlo (en este caso, nombres para designar valores numéricos, verificando cuáles de ellos estarán asociados a valores que deben ingresar al algoritmo, cuáles a valores intermedios o de procesamiento, y cuáles a valores resultantes).
3. Seguir alguna estrategia para encontrar y desarrollar la solución (como por ejemplo individualizar, aislar y ordenar las distintas actividades para conseguir una solución).
4. Comprobar que la solución obtenida es buena.

El algoritmo de Euclides contiene::

Recursos	Acciones	Interlocutor
<ul style="list-style-type: none"> • M (valor ingresado) • N (valor ingresado y resultante) • R (valor intermedio) 	Por ejemplo: • Asignar a N el valor de R	• Persona que sigue el algoritmo.

Formalicemos los conceptos que se desprenden del ejemplo planteado:

Interlocutor
Entidad capaz de comprender un enunciado y ejecutar el trabajo.

Recurso
Medio al que se puede recurrir para lograr algo.
El conjunto de todos los recursos necesarios para la ejecución de un trabajo constituye el **ambiente** del trabajo.

Acción
Evento que modifica el ambiente. Una acción es **primitiva** para un interlocutor dado, si su enunciado es suficiente para que pueda ejecutarla sin información adicional.

Formalicemos los conceptos que se desprenden del ejemplo planteado: resolución de problemas.

Según G. Polya, las fases para resolver un problema son:

```

graph TD
    A[Entender el problema] --> B[Idear un Plan]
    B --> C[Llevar a cabo el plan]
    C --> D[Evaluar la solución]
    D --> A
    D --> B
    D --> C
    
```

La resolución de un problema implica un proceso dinámico en el que se deben repasar una y otra vez las fases en forma evolutiva retrocediendo y avanzando hasta que la solución sea satisfactoria. El orden de ejecución de las fases puede variar.

Figura 5.3: Nivel Medio de presentación del concepto algoritmo

Apellido	Nombre	Estructuras de Control	Secuencia	Selección	Repetición	Transferencia o Retorno
Acosta	Alejandro	(1,50)	(1,50)	(1,50)	(1,40)	(1,60)
Alberdi	Fermín	(0,10)	(0,10)	(0,20)	(0,10)	(0,0)
Alfonsín	Diego					
...						
Zachalewicz	Alejandra	(1,60)	(1,100)	(1,50)	(1,50)	(0,40)

Cuadro 5.1: Muestra de Base de Datos de Módulo del Alumnos

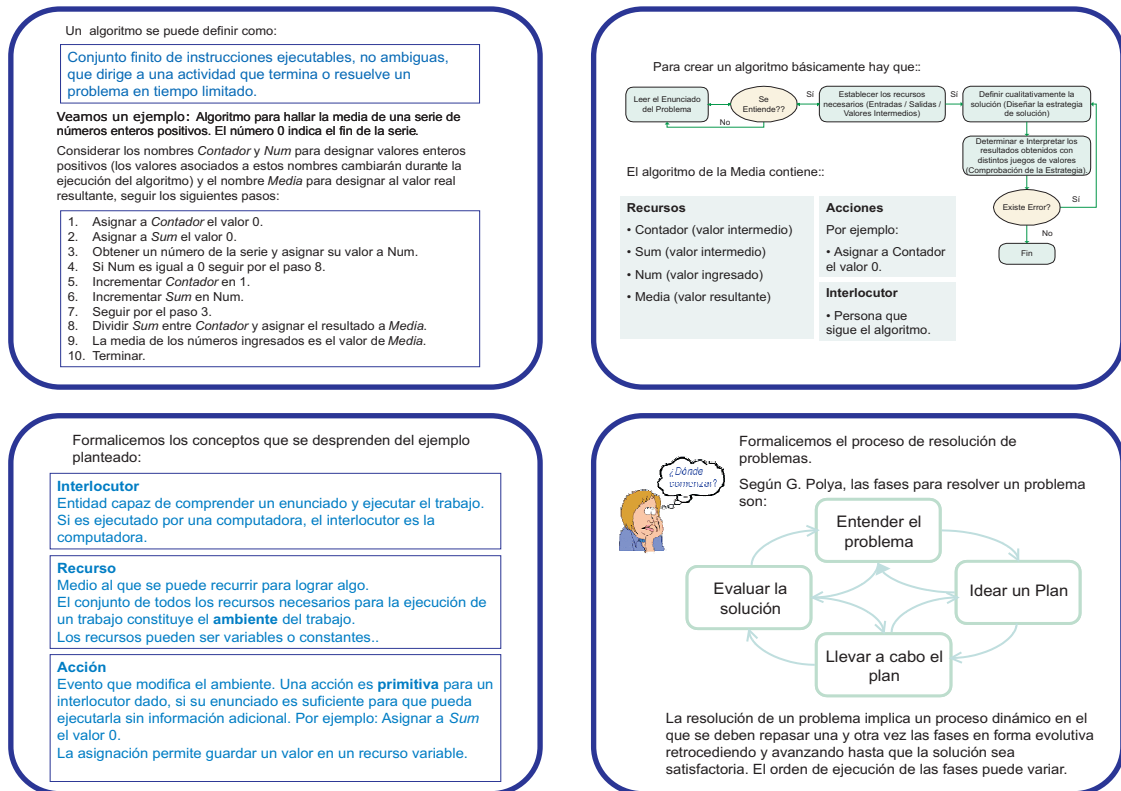


Figura 5.4: Nivel Experto de presentación del concepto algoritmo

5.1.3. Módulo del Tutor o de Adaptación

La adaptación de la información se basa en un conjunto de reglas de adaptación. Estas reglas establecerán la relación entre el modelo del dominio, el modelo del usuario y la presentación personalizada que se generará para cada usuario. La sintaxis es del tipo:

```
SI <Antecedente> ENTONCES <Consecuente>
siendo
    <Antecedente> : nivel de conocimiento obtenido por el alumno en
                    los conceptos y temas.
    <Consecuente>: acciones de adaptación que va a realizar el sistema.
```

Las reglas se obtienen partiendo del nivel de conocimiento obtenido por el alumno en los tests iniciales y finales sobre cada tema y en las actividades. Por ejemplo, para determinar el nivel de complejidad que se utilizará para presentar un tema, se establecen reglas del siguiente tipo:

```
SI Nivel_Test_Inicial_Algoritmos=10 THEN
    NivelPrincipiante de presentación del tema Algoritmos
```

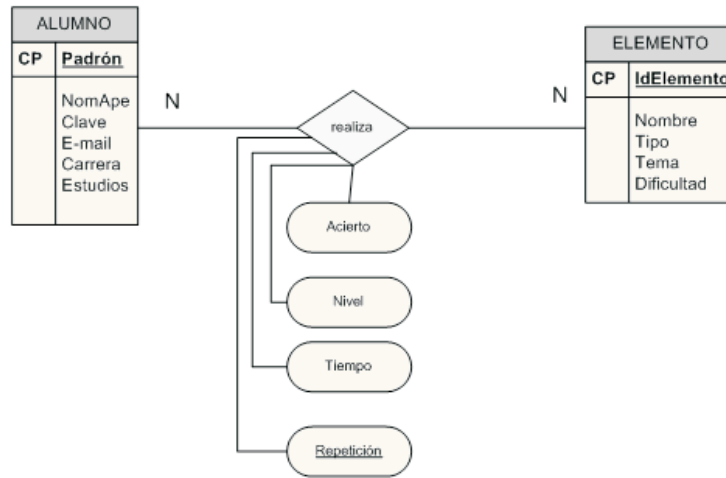


Figura 5.5: Modelo Entidad Relación

En tanto que, para determinar el nivel de conocimiento final alcanzado por el usuario en un tema se establecen reglas del siguiente tipo:

```
SI Valor_Test_Final_Algoritmos=50 THEN
    NiveldeConocimiento Intermedio en el tema Algoritmos.
```

5.2. Integración de los Datos

La información de uso, referente a tiempos de acceso, respuestas correctas y niveles de conocimiento de cada estudiante se recoge del sistema en archivos log. Un archivo log [19] es un archivo de texto en el que se almacena cada petición de una página que se le hace al servidor web.

Del sistema se obtiene también información general de cada alumno: nombre, e-mail, padrón, clave de acceso, universidad, carrera, estudios. Los datos recolectados se integran en una base de datos relacional (figura 5.5).

En el diagrama de la figura 5.5 se observan las siguientes entidades:

- **Alumno**

Representa a las personas que finalizan el curso adaptativo de computación y el profesor las selecciona para analizar sus actuaciones en el desarrollo del curso.

Sus atributos son: Padrón, NombreApe, Clave, E-mail, Carrera y Estudios.

El atributo Padrón (clave principal), es de tipo entero y los restantes son de tipo cadena de caracteres.

- **Elemento**

Representa la información referente a tests, actividades y contenido y sus niveles de complejidad para cada uno de los temas que componen el curso.

Sus atributos son: IdElemento, Nombre, Tipo, Tema y Dificultad El atributo IdElemento (clave principal), es de tipo entero y los restantes son de tipo cadena de caracteres.

- La relación “**realiza**”, entre las entidades ALUMNO y ELEMENTO tiene una cardinalidad de N:N (1 ALUMNO que finaliza el curso pudo haber realizado un elemento determinado una o más veces pues es condición realizarlo para finalizar el curso. En la relación en sentido inverso se cumplirá que un mismo ELEMENTO puede ser realizado por uno o más ALUMNOS). Esta relación posee los atributos acierto,nivel, tiempo, IdRepetición. Acierto toma los siguientes valores: SI, NO. El atributo Nivel puede tomar los valores: PRINCIPIANTE, MEDIO, EXPERTO. El atributo Tiempo puede tomar los valores ALTO, MEDIO y BAJO. Finalmente, el atributo Repetición representa el número de veces que el alumno repite un elemento.

La representación tabular de las entidades es:

```
ALUMNO(Padrón, NombreApe, Clave, E-mail, Carrera, Estudios)
ELEMENTO(IdElemento, Nombre, Tipo, Tema, Dificultad)
```

En tanto que la representación tabular de la relación es:

```
Realiza(NombreApe,IdElemento, IdRepeticion, Tiempo, Nivel, Acierto)
```

5.3. Información a Descubrir

Al finalizar cada cuatrimestre, a través de la información recogida acerca de la actuación de los alumnos inscriptos en el curso, se obtienen datos estadísticos como la media de alumnos que aprobaron la materia, las notas máximas y mínimas alcanzadas, la cantidad de alumnos que no finalizaron el curso, etc.

La aplicación de algoritmos de minería de datos para el descubrimiento de reglas de predicción, pretende obtener información referente a relaciones entre los datos de utilización. En particular, la información que se desea encontrar está relacionada con los atributos tiempo, acierto y nivel.

En la Universidad de Córdoba (España), Cristóbal Romero,Sebastián Ventura Soto y Carlos de Castro, desarrollaron una herramienta, denominada EP-Rules (Education Prediction Rules) [5], que facilita el proceso de descubrimiento de información sobre los datos de un sistema educativo basado en Web. El docente, a través de la utilización de esta herramienta puede

```

Regla ::= "SI" (<antecedente>) "ENTONCES" <consecuente>
<antecedente> ::= <condición> | <condición> "Y" <antecedente>
<consecuente> ::= <condición>
<condición> ::= <atributo> <operador> <valor> | <atributo> <operador> <atributo>
<atributo> ::= Cada uno de los posibles atributos del conjunto
<valor> ::= Un valor del dominio del atributo correspondiente
<operador> ::= "=" | "<" | ">" | "≤" | "≥" | "≠"
    
```

Figura 5.6: Formato genérico de una regla de predicción utilizando la notación EBNF (Forma de Backus y Naur Extendida)

- Abrir su base de datos.
- Seleccionar el algoritmo de descubrimiento de reglas (ID3, A priori y diferentes versiones de algoritmos de programación genética), sus parámetros de ejecución y las restricciones que deben cumplir las reglas.
- Visualizar los datos de utilización del curso, las reglas de predicción descubiertas y resultados estadísticos.

La figura 5.6 muestra el formato genérico de una regla de predicción. Como puede observarse, el consecuente sólo presenta una condición y su contenido puede ser de cualquier tipo.

Según el significado semántico de una regla de predicción, si todas las condiciones especificadas por el antecedente de la regla son satisfechas por los atributos predictores de un ejemplo, la regla predice que el atributo objetivo (el que aparece en el consecuente) para esa instancia [7]. Este funcionamiento es lo que le da su nombre a este tipo de reglas, ya que a partir de un ejemplo dado, que verifique el antecedente, puede predecir el valor del atributo que aparece en el consecuente. Las reglas de predicción relacionadas con los atributos Nivel, Tiempo y Acierto, tendrán la forma:

```

SI Nivel/Tiempo/Acierto Y ... ENTONCES Nivel/Tiempo/Acierto
donde
    Nivel: es una condición de tipo nivel.
    Tiempo: es una condición de tipo tiempo.
    Acierto: es una condición de tipo acierto.
    
```

En el esquema de la figura 5.7 se muestran algunas de las posibles reglas de predicción que se pueden obtener.

Una de las reglas es la siguiente

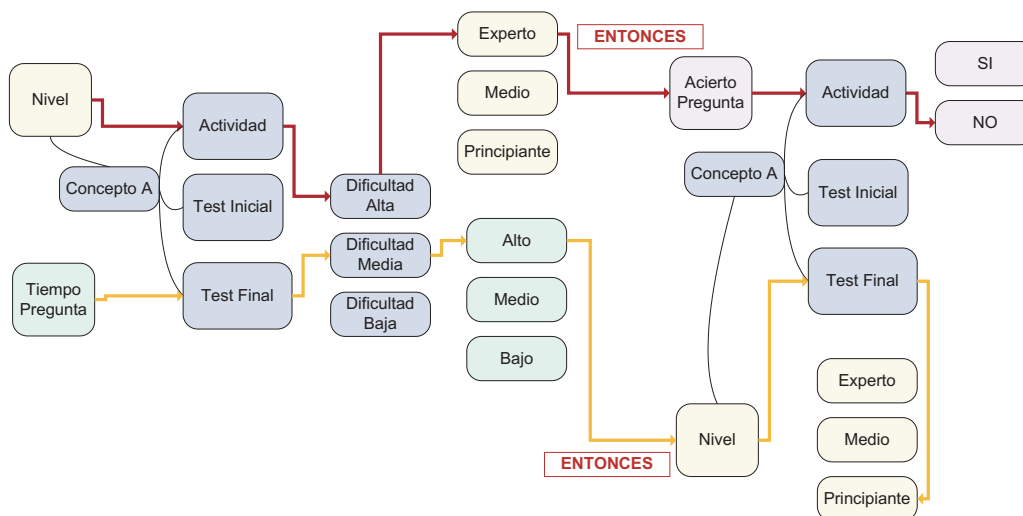


Figura 5.7: Ejemplo de Reglas de Predicción

SI Nivel_Actividad_ConceptoA_DificultadAlta= Experto ENTONCES
 AciertoPregunta_Actividad_ConceptoA = NO

Considerando que *A* representa al concepto Algoritmo, la regla anterior muestra la relación entre una Actividad del concepto Algoritmo con dificultad Alta y en la que el alumno obtiene el nivel Experto con el desacierto a una pregunta de esa actividad. En base a las métricas de calidad que se obtengan, se podrán tomar decisiones como: modificar la pregunta si se llega a la conclusión de que es confusa, o modificar su grado de dificultad si se considera que no refleja el nivel desarrollado en la presentación teórica del concepto.

Otro ejemplo de regla de predicción planteado en el esquema de la figura 5.7 es:

SI TiempoPregunta_TestFinal_ConceptoA_DificultadMedia = Alto ENTONCES
 Nivel_ConceptoA = Principiante

Esta regla muestra la relación entre un elevado tiempo de visualización de una pregunta del test final del concepto algoritmo con grado de dificultad media y el nivel alcanzado en el test final del concepto algoritmo; en este caso es Principiante.

En base a las métricas de calidad que se obtengan, se podrán tomar decisiones como: modificar la pregunta si se llega a la conclusión de que es confusa o errónea o no corresponde al nivel de dificultad media.

Lo más importante de este enfoque es la posibilidad de obtener estas reglas de manera automática. En general, es importante utilizar métodos que aporten un conjunto de reglas reducido y significativo.

Los valores indicados en los parámetros determinarán la eficiencia y eficacia de las reglas.

De la combinación de los resultados obtenidos para distintos valores de soporte, confianza, interés, etc, es posible construir un modelo de la información disponible que, como se sugiere en este caso, permita mejorar la información presentada a los alumnos y obtener así un mejor rendimiento por parte de ellos.

Capítulo 6

Conclusiones y Perspectivas

En este trabajo se ha partido de una situación de enseñanza concreta: el desarrollo de un curso de la asignatura Computación en la Facultad de Ingeniería de la Universidad de Buenos Aires, y se han descripto los protagonistas de esta relación: docente, alumno y contenido, y el contexto en el cual se encuentran inmersos.

Se han detallado cuáles son los objetivos de la materia y cómo, a partir del programa sintético establecido por la institución, se determinan sus principios procedimentales y conceptuales y se organiza el contenido en unidades temáticas que reflejan su estructura interna. De hecho, lo que se logra es adecuar el programa institucional a una situación particular.

También se han planteado las estrategias y estilos de enseñanza actuales del docente y los diferentes intereses y estilos cognitivos de los alumnos. En particular, se ha sentado la convicción docente de la importancia de

- Organizar los contenidos a través de mapas conceptuales que doten al conocimiento de la estructura necesaria para que pueda ser fijado con facilidad.
- lograr en el alumno un aprendizaje por recepción pero significativo.

Asimismo, se ha descripto cómo la incorporación de recursos tecnológicos como la utilización de presentaciones multimedia y la asesoría virtual a través del correo electrónico han permitido al docente acercarse más a los objetivos planteados. La comunicación asincrónica produjo un cambio positivo en lo referente a la modalidad de participación del alumno, teniendo éste un rol más activo y creativo.

También se han detallado las falencias que se detectan en el aprendizaje de los alumnos y con el fin de seleccionar una metodología para la mejora de la enseñanza, se han analizado diversos sistemas que aplican la Inteligencia Artificial en la educación con el objetivo de obtener sistemas de enseñanza inteligentes.

En particular, el análisis se ha enfocado a los Sistemas Hipermedia Adaptativos basados en la WEB como una herramienta para lograr una enseñanza individualizada y flexible. Bajo esta modalidad se ha propuesto el diseño de un curso de la asignatura Computación

utilizando como base el sistema AHA! junto a las mejoras llevadas a cabo en la Universidad de Córdoba por Cristóbal Romero.

Se presentaron cada uno de los módulos que componen el diseño del modelo adaptativo propuesto; estos son:

- El Modelo del Dominio partiendo de la estructura del curso y se desarrollaron presentaciones de conceptos que se adaptan a tres niveles de conocimientos de los alumnos: principiante, medio y experto. Se procuró componer el significado de las interfaces utilizando un lenguaje claro y simple, acompañado de imágenes que grafiquen, contextualicen y complementen. La presentación de los conceptos mediante organizadores previos genera un puente cognitivo entre los conocimientos que el alumno ya posee en su estructura cognitiva y los conceptos a ser aprendidos durante el desarrollo de la asignatura.
- El Modelo del Alumno, módulo en el cual se recoge y cuantifica la actuación de cada alumno junto con sus preferencias.
- El Módulo del Tutor que permite adaptar el contenido a cada alumno mediante reglas. Es decir, este módulo de acuerdo al módulo del alumno adapta el módulo del dominio.

Se describió de qué manera se recoge la información en archivos log y cómo se integra a los datos de cada alumno (nombre, e-mail, padrón, etc.) mediante una base de datos relacional.

Para analizar el resultado del curso, se propuso el uso de técnicas de Minería de Datos aplicadas a la educación con el objetivo de obtener no sólo información útil acerca del alumno sino también información útil para el profesor y la institución.

Se seleccionó la herramienta EP-Rules (Education Prediction Rules) que permite descubrir reglas de predicción entre los datos de utilización del curso. El análisis de estas reglas por parte del profesor le permitirán decidir las modificaciones que deberá hacer en el curso para lograr una mejora en el proceso de enseñanza-aprendizaje.

Se prevé incorporar al diseño del curso el sistema adaptativo propuesto, utilizándolo como complemento de la enseñanza y monitoreo de alumnos. Por su intermedio se espera descubrir conocimiento sobre el curso que ayude a la toma de decisiones para su mejora.

En lo personal, considero sumamente interesante el uso de estas herramientas adaptativas ya que si bien a través de esta propuesta se busca modelizar la información disponible referida al desempeño de los alumnos frente a la plataforma, presenta un amplio abanico de posibilidades futuras.

Si se analiza su impacto en la práctica docente y se parte de la etapa de planificación del curso, de acuerdo a lo expuesto por Beauchamp (1981) y citado en [21]: “La tarea del que planifica el currículum consiste en establecer la estructura fundamental de un ambiente en el que los que aprenden puedan tener experiencias de aprendizaje”. En particular, para el diseño de un curso adaptativo, el docente debe tomar decisiones acerca del contenido

y la transposición didáctica del mismo, en tanto recibe el aporte del sistema en las tareas de observación y registro del proceso de aprendizaje de cada alumno, facilitándole la tarea de análisis de los resultados. Por su parte, los alumnos, a través de esta propuesta se sumergen en una experiencia de enseñanza-aprendizaje dinámica, caracterizada por la comunicación y la interacción continua.

Uno de los temas que más ha llamado mi atención es la posibilidad de incorporar conocimiento al modelo de manera permanente. Es decir, que en lugar de trabajar con toda la información del problema a priori, es posible continuar incorporando información que se refleje en el modelo con una cierta periodicidad. Esto permitiría cambiar el comportamiento del sistema inteligente a largo plazo. En el caso del ejemplo planteado para el curso de Computación, que ha sido motivo de estudio a lo largo de este trabajo, esto se traduciría en la posibilidad de continuar observando el comportamiento de los alumnos frente a la plataforma luego de haber obtenido un modelo de reglas inicial, arribando de esta forma a nuevas reglas de predicción.

En resumen, el proceso adaptativo se realizaría de manera continua, ya que a lo largo de su uso se podrían incorporar modificaciones al material de estudio lo que redundaría en un desempeño diferente, presumiblemente mejor, por parte de los alumnos. Resultaría de interés que esta tarea se lleve a cabo sin tener que volver a procesar todos los datos otra vez; es decir, que simplemente se agregue o se modifique el modelo existente para poder representar la nueva información.

Apéndice A

Tareas y Técnicas de Minería de Datos

A continuación se muestran algunas tareas de Minería de Datos y algunas técnicas o algoritmos que pueden abordarlas:

Nombre	Predictivo		Descriptivo		
	Clasificación	Regresión	Agrupamiento	Reglas de Asociación	Correlación
Redes Neuronales	✓	✓	✓		
Arboles de decision CART	✓				
Otros árboles de decisión	✓	✓	✓	✓	
Redes de Kohonen			✓		
Regresión Lineal y logarítmica		✓			✓
K-Medias			✓		
A priori				✓	
Naive Bayes	✓				
Vecinos más próximos	✓	✓	✓		
Algoritmos Genéticos y Evolutivos	✓	✓	✓	✓	✓
Máquinas de Vectores de Soporte	✓	✓	✓		
CN2 rules (cobertura)	✓			✓	
Análisis Discriminante Multivalente	✓				

Apéndice B

Herramientas de Software para Minería de Datos

B.1. SPSS Clementine

Sitio : <http://www.spss.com/spssbi/clementine>

Característica	Descripción
Empresa	SPSS
Fuentes de Datos	ODBC, tablas Excel, Archivos ASCII, archivos SPSS
Técnicas de Aprendizaje	Arboles de decisión, redes neuronales, agrupamiento, reglas de asociación, regresión lineal y combinación de modelos
Visualización de resultados	Grafica desde el análisis del problema hasta la imagen final del modelo aprendido.

B.2. WEKA (Waikato Environment for Knowledge Análisis)

Sitio : <http://www.cs.waikato.ac.nz/ml/weka/>

Característica	Descripción
Empresa	Licencia GNU de libre distribución
Fuentes de Datos	ARFF, CVS, Archivos ASCII
Técnicas de Aprendizaje	Arboles de decisión, tablas de decisión, vecinos más próximos, máquinas de vectores soporte, reglas de asociación, métodos de agrupamiento, modelos combinados
Visualización de resultados	Interfaz del usuario gráfica y de línea de comandos.
Exportación	No presenta problemas de portabilidad. Implementada en Java

B.3. Kepler

Sitio : <http://www.dialogis.de>

Característica	Descripción
Empresa	Dialogis
Fuentes de Datos	Distintos formatos incluyendo información estructurada en BBDD.
Técnicas de Aprendizaje	Arboles de decisión, redes neuronales, regresión no lineal, vecinos más próximos, algoritmos multirelacionales, utilidades estadísticas.
Visualización de resultados	Interfaz gráfica.
Exportación	No tiene problemas de portabilidad. Datos cartográficos se pueden visualizar con Descartes.

B.4. ODMS (Oracle Data Mining Suite) Darwin

Sitio : <http://otn.oracle.com/products/datamining>

Característica	Descripción
Empresa	Oracle
Fuentes de Datos	Bases de datos relacionales, Archivos planos, conjunto de datos SAS
Técnicas de Aprendizaje	Redes neuronales para clasificación y regresión, regresión lineal, inferencia de árboles de decisión usando el criterio CART, vecinos más próximos, aprendizaje bayesiano, técnicas de agrupamiento.
Visualización de resultados	Interfaz gráfica. Visualizador interactivo de árboles de decisión.
Portabilidad	No tiene problemas de portabilidad.

B.5. DBMiner

Sitio : <http://www.dbminer.com>

Característica	Descripción
Empresa	DBMiner Technology Inc.
Fuentes de Datos	Puede utilizar distintos tipos de Bases de Datos: relacionales, Orientadas a Objetos, texto y heterogéneas.
Técnicas de Aprendizaje	Arboles de Decisión.
Visualización de resultados	Interfaz gráfica. Lenguaje script

B.6. DB2 Intelligent Miner

Sitio : <http://www.3.ibm.com/software/data/iminer>

Característica	Descripción
Plataforma	Aplicaciones servidores : Windows(2000, NT, XP) y z/OS Aplicaciones Cliente : Windows y AIX
Empresa	IBM
Fuentes de Datos	DB2 Database
Tipos de Tareas	Agrupamiento, Asociaciones, Patrones, Clasificación, Predicción, Análisis de series temporales.

B.7. SAS Enterprise Miner

Sitio : <http://www.sas.com>

Característica	Descripción
Plataforma	Windows, Linux, Solaris, HP-UX, Digital Unix
Empresa	SAS Institute
Fuentes de Datos	Archivo propio de SAS. Archivos de BD: Oracle, DB2, Sybase, etc.
Técnicas de Aprendizaje	Árboles de decisión, Regresión lineal y logística, Redes neuronales, Construcción de modelos múltiples.
Visualización de resultados	Interfaz gráfica. Generador automático de informes. Formato HTML.
Portabilidad	No presenta problemas.

B.8. STATISTICA Data Miner

Sitio : <http://www.statsoft.com/dataminer.html>

Característica	Descripción
Plataforma	Windows
Empresa	StatSoft Ltd.
Fuentes de Datos	MS Excel, tablas de dBase, ASCII, Lotus, BD Oracle, MSsql, Server y Sybase. Formato de archivos propio.
Técnicas de Aprendizaje	Reglas de asociación, Árboles de decisión, Agrupamiento, Redes Neuronales
Visualización de resultados	Interfaz gráfica. Representación gráfica de modelos. Gráficos estadísticos.

Índice de figuras

1.1. Estrategia a seguir cuando no se logra resolver un problema	6
1.2. Organización del Contenido de la Asignatura Computación	6
2.1. Evolución de los Sistemas de Enseñanza	10
2.2. Arquitectura de un STI	11
2.3. Ejemplo de Interfaces	15
2.4. Ejemplo de estructura de un Sistema Adaptativo	16
2.5. Tecnologías de Adaptación utilizadas en Sistemas Hipermedia Adaptativos	17
2.6. Arquitectura del Sistema AHA!	18
2.7. Técnicas de Adaptación utilizadas en AHA!	19
3.1. Fases del proceso de descubrimiento de conocimiento en bases de datos KDD	21
3.2. Proceso ideal de Minería de Datos	22
3.3. Ejemplo de Arbol de Decisión	26
3.4. Computación Evolutiva	32
4.1. Triángulo didáctico	35
4.2. Incorporación del Contenido Didáctico al Triángulo Didáctico	36
4.3. Aplicación Adaptativa	37
4.4. Relación entre profesores, estudiantes, entorno educativo y Minería de Datos	37

5.1. Estructura del Curso de Computación	40
5.2. Nivel Principiante de presentación del concepto algoritmo	41
5.3. Nivel Medio de presentación del concepto algoritmo	42
5.4. Nivel Experto de presentación del concepto algoritmo	43
5.5. Modelo Entidad Relación	44
5.6. Formato genérico de una regla de predicción utilizando la notación EBNF (Forma de Backus y Naur Extendida)	46
5.7. Ejemplo de Reglas de Predicción	47

Bibliografía

- [1] Rakesh Agrawal and Ramakrishnan Srikant. "mining sequential pattern". *11th International Conference on Data Engineering*, pages 3–14, 1995.
- [2] Rakesh Agrawal and Ramakrishnan Srikant. Method and system for mining generalized sequential patterns in a large database. *United States Patent 5742811*, 1998.
- [3] Rakesh Agrawal and Tomasz Imielinski and Arun Swami. "mining association rules between sets of items in large databases". *ACM SIGMOD Conference on Management of Data*, pages 207–216, 1993.
- [4] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Model. User-Adapt. Interact.*, 6(2-3):87–129, 1996.
- [5] Romero C, Ventura S, and Hervás C. Descubrimiento de reglas de predicción en sistemas de e-learning utilizando programación genética. *Tendencias de la Minería de Datos en España*, 2004.
- [6] Daniel Prieto Castillo. *Comunicación con los medios y materiales*. CICCUS la Crujia, 1999.
- [7] Mihaela Cocea and Stephan Weibelzahl. Can log files analysis estimate learners' level of motivation?". *Proceedings of the Workshop Week Lernen - Wissensentdeckung - Adaptivität (LWA2005)*, pages 32–35, 2006.
- [8] Julia M. Denazis. Teorías psicológicas con énfasis en el aprendizaje. Curso de Complementación para docentes: Introducción a la didáctica en la enseñanza superior. FIUBA, 2002.
- [9] Chen Gwo Dong, Chen Chung Liu, Kuo Liang Ou, and Ming Song Lin. Discovering decision knowledge from web log portfolio for managing classroom processes by applying decision tree and data cube technology. *Educational Computing Research*, 2000.
- [10] R.O Duda and Hart P.E. *Pattern Classification and scene analysis*. John Wiley and Sons, 1973.

- [11] M. Kilfoil et al. Toward an adaptive web: the state of the art and science. *Procs of the Annual Conference on Communication Networks and Services Research*, pages 119–130, 2003.
- [12] Jonathan E. Freyberger, Neil Heffernan, and Carolina Ruiz. Using association rules to guide a search for best fitting transfer models of student learning. Master's thesis, Worcester Polytechnic Institute, 2004.
- [13] Brookshear J. G. *Introducción a las ciencias de la Computación*. ADDISON-WESLEY IBEROAMERICANA, 1995.
- [14] Selma Pimenta Garrido. Educação, pedagogia e didática. *Anais do VII Encontro Nacional de Didática e Prática de Ensino*, 1:44–64, 1994.
- [15] Carina Soledad González. Sistemas inteligentes en la educación: una revisión de las líneas de investigación y aplicación actuales. *Revista Electrónica de Investigación y Evaluación Educativa RELIEVE*, 10(1):3–22, 2004.
- [16] Pere Marqués Graells. Las comunidades virtuales de profesores.
- [17] Bou Bouzá Guillem. *El Guión Multimedia*. ANAYA, 1997.
- [18] Cabero Almenara Julio. Navegando, construyendo: la utilización de los hipertextos en la enseñanza. *II Jornadas sobre Medios de Comunicación, Recursos y Materiales para la Mejora Educativa : Sevilla*, 2:201–243, 1996.
- [19] Cabral Jose L. *A Data Mining Model to Capture User Web Navigation Patterns*. PhD thesis, London, 2000.
- [20] Maite Urretavizcaya Loinaz. Sistemas inteligentes en el ámbito de la educación. inteligencia artificial. *Revista Iberoamericana de Inteligencia Artificial*, (12):5–12, 2001.
- [21] Marisa Martín. El profesor como gestor de su práctica docente. Manual del Curso: Rediseño de la Práctica Docente con base en la Misión del 2005. Módulo 2. ITESM, 2005.
- [22] Marisa Martín. Selección y estructuración de contenidos. Manual del Curso: Rediseño de la Práctica Docente con base en la Misión del 2005. Módulo 5. ITESM, 2005.
- [23] Merceron and K. Yacef. A web-based tutoring tool with mining facilities to improve learning and teaching. *Procs of 11th International Conference on Artificial Intelligence in Education (AIED03)*, page 201 – 208, 2003.
- [24] Merceron and K. Yacef. Educational data mining: a case study. *Procs of 12th Conference on Artificial Intelligence in Education (AIED03)*, page 467 – 474, 2005.
- [25] Enric Mor and Juliá Minguillón. E-learning personalization based on itineraries and longterm navigational behavior. *International World Wide Web Conference archive Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters*, pages 264–265, 2004.

- [26] Cristóbal Romero Morales, Sebastián Ventura Soto, and César Hervás Martínez. Aplicación de algoritmos evolutivos como técnica de minería de datos para la mejora de cursos hipermedia adaptativos basados en web. *RIED: revista iberoamericana de educación a distancia*, 6(2):142–163, 2003.
- [27] Cristóbal Romero Morales, Sebastián Ventura Soto, and César Hervás Martínez. Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web. *Actas del III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA2005*, pages 49–56, 2005.
- [28] Raymond Nickerson, David Perkins, and Edward Smith. *Enseñar a pensar - Aspectos de la aptitud intelectual*. PAIDÓS, 1990.
- [29] José Hernández Orallo, José Ramírez Quintana, and César Ferri Ramírez. *Introducción a la Minería de Datos*. PEARSON PRENTICE HALL, 2004.
- [30] De Bra P, Aerts A., Berden B., de Lange B., Rousseau B., Santic T., D. Smits, and N. Stash. Aha! the adaptive hypermedia architecture. *ACM Conference on Hypertext and Hypermedia*, page 81 – 84, 2003.
- [31] Langley P. and Thompson K. Iba W. An analysis of bayesian classifiers. *Proceedings of the 10th National Conferences on Artificial Intelligence*, pages 223–223, 1992.
- [32] Fernández Palacios, Arenas Gutiérrez, and Pérez LLanes. Sistemas hipermedia adaptativos: una aproximación al tema. 2002.
- [33] A. Paranythis and S. Loidl-Reisinger. Adaptive learning environments and elearning standards. *Electronic Journal on e-Learning*, 2(1):181–194, 2004.
- [34] Calderón Sánchez Raimundo. Constructivismo y aprendizajes significativos.
- [35] Guillermo Ricci, Cecilia Sanz, and Armando De Giusti. Herramientas de comunicación sincrónica coordinada en educación a distancia. *Primeras Jornadas de Educación en Informática y TICS en Argentina JEITICS 2005*, pages 134–138, 2005.
- [36] Jun-Ming Su, Shian-Shyong Tseng, Wei Wang, and Jui-Feng Weng. Learning portfolio analysis and mining for scorm compliant environment. *Educational Technology and Societ*, 9(1):262–275, 2004.
- [37] Enrique Javier Carmona Suárez. Algoritmos evolutivos para minería de datos. *Segunda Escuela del Observatorio Virtual Español, UNED*, 2007.
- [38] R. Taylor. The computer in the school: tutor, tool, tutel. *USA: Teacher's College Pres. Columbia University*, 1980.
- [39] Luis Talavera y Elena Gaudio. Mining student data to characterize similar behavior groups in unstructured collaboration spaces. *Artificial Intelligence in Computer Supported Collaborative Learning, Workshop 21, ECAI04*, pages 17–23, 2004.

- [40] O. Zaine and R. Luo. Towards evaluating learners' behaviour in a web-based distance learning environment. *Procs. of the Advanced Learning Technologies*, 2001.
- [41] Alejandra Zangara. Material del seminario de educación a distancia. facultad de informática. unlp. 2005.