

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA



TESIS DE GRADO

“SISTEMA TUTOR PARA ENSEÑANZA
DEL LENGUAJE SQL”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: RUDDY GABRIEL CHAVARRIA CHOQUE

TUTOR: LIC. MARIO LOAYZA MOLINA

REVISOR: LIC. JUAN CONTRERAS CANDIA

LA PAZ - BOLIVIA

2009

Dedicatoria

Por y para mi padre "Nachito" por su toda su paciencia y comprensión y cariño que siempre me brindo.

A mi Hermanita "Magali" por el apoyo y Amor que siempre me tiene, a mis sobrinitos.

A Dios el ser supremo, por la familia que me dio.

A un amigo muy especial, que siempre estará en el corazón aunque ya no estés con nosotros "Gabriel".
En mi memoria vivirás.

A María una persona muy especial en mi vida.

Agradecimientos

En primer lugar expresar mi agradecimiento al Lic. Mario Loayza Molina por su apoyo y consejos, al desarrollo del Presente trabajo. Por brindar su sabiduría, toda su experiencia de forma desinteresada y absoluta.

Agradecer de antemano al Lic. Juan Contreras Candía por todo el tiempo dedicado, quien me apoyo de incesante manera, y me colaboro de forma sustanciosa en la revisión secuencial de la tesis, con observaciones siempre oportunas.

¹ Tesis de Grado “Sistema Tutor para la Enseñanza del Lenguaje SQL” Postulante Ruddy Gabriel Chavarria Choque Email: gabrielchavarria25@hotmail.com, gabrielchavarria25@gmail.com

RESUMEN

Se ha observado la existencia de dificultades en los aprendizajes de los estudiantes que se inician en el estudio de un lenguaje de programación, puesto que no pueden interpretar los algoritmos debido a su mala preparación previa y por falta de metodologías de trabajo. En este contexto se busca modelar un tutor que permita definir y aplicar una estrategia pedagógica para la enseñanza de las estructuras básicas de programación, en caso particular el Lenguaje SQL (lenguaje estructurado de consulta) el cual permite implementar en forma flexible un Sistema Tutor Inteligente, que asiste a los estudiantes, que cursan la materia de taller de base de datos.

El tutor es una herramienta automatizada, dinámico y de fácil aplicación que sirve de apoyo al proceso de aprendizaje, ya que ofrece un marco conceptual para comprender y utilizar este lenguaje, puesto que la explosiva popularidad del lenguaje, es una de las más importantes en la industria informática hoy en día.

Entre las muchas ventajas del tutor, es reducir las limitaciones de tiempo, como recursos, puede ser practicado cuando así lo requieran, puesto que estará disponible en todo momento, con lo que se pretende dar los lineamientos que deben regir el comportamiento del tutor para adaptarse a las necesidades del estudiante y pueda mejorar su rendimiento en la materia sobre todo en cursos bastante numerosos.

Contenido de la Tesis

	Página
CAPÍTULO 1: Introducción	
1.1. Introducción	1
1.2. Antecedentes	2
1.3. Planteamiento del problema	4
1.3.1. Problema general	6
1.3.2. Problemas específicos	7
1.4. Objetivos	9
1.4.1. Objetivo general	9
1.4.2. Objetivos específicos	9
1.5. Hipótesis	10
1.6. Justificación	10
1.6.1. Social	10
1.6.2. Técnica	10
1.6.3. Económica	11
1.7. Metodología	11
1.8. Alcance y límites	12
1.9. Aportes	14
CAPÍTULO 2: Marco Teórico	

2.1. Introducción	15
2.2. Inteligencia Artificial	16
2.3. Sistemas Tutores Inteligentes	18
2.4. Teorías de Aprendizaje-Enseñanza	20
2.4.1. La teoría conductiva	20
2.4.2. La teoría cognoscitiva	20
2.4.3. La teoría constructiva	21
2.4.4. Técnicas Educativas Aplicables	26
2.4.5. Que es enseñanza	29
2.4.5.1. Como definimos el proceso de enseñar	29
2.5. Las características de los agentes del STI	30
2.6. Agentes inteligentes	31
2.6.1. Propiedades de los agentes inteligentes	32
2.6.2. Estructura de los agentes inteligentes	34
2.6.3. Modelos de implementación de Sistemas Tutores Inteligentes	36
2.6.4. Trabajos similares	36
2.6.4.1. El proyecto CirSim	37
2.6.4.2. STI orientados a la enseñanza de lenguajes de programación	38
2.7. Metodología	39
2.7.1. Método científico	39
2.7.2. Método de ingeniería de software orientado a objetos	40
2.8. Sistemas basados en el conocimiento	40
CAPÍTULO 3: Arquitectura del Sistema Tutor SQL	
3.1. Introducción	42

3.2. Arquitectura del sistema tutor	42
3.2.1. El módulo estudiante del STI	45
3.2.2. El módulo del dominio	45
3.2.3. El módulo del tutor del STI	47
3.3. Análisis de los motivos del módulo tutor	49
3.3.1. Submódulo protocolos pedagógicos	50
3.3.2. El analizador de perfil	51
3.4. Módulo Evaluación	51
3.4.1. Interfaz	54
3.4.2. Interrelación entre componentes	55
3.5. Desarrollo orientado a objetos con UML	55
3.6. Fases de planificación y especificación de requisitos	56
3.6.1. Requerimientos del usuario	57
3.6.2. Requerimientos de procesos de datos	58
3.6.3. Requerimientos del sistema	58
3.6.3.1. Requerimientos de Software	58
3.6.3.2. Requerimientos de Hardware	58
3.6.4. Requisitos de interfaz de usuario	59
3.6.5. Funciones del sistema	59
3.6.6. Casos de uso	61
3.6.7. Diagrama de caso de uso	66
3.6.8. Diagrama de secuencia	66
3.6.9. Diagrama de estados	68
3.7. Fases de construcción: diseño	69
3.7.1. Diagrama de interacción	69
3.7.2. Diagramas de colaboración	69

3.7.3. Diagrama de clases de diseño	70
3.7.4. Diagrama de diseño	71
3.8. Fase de construcción: implementación y pruebas	72

CAPÍTULO 4: implementación del prototipo y medición de la calidad del software

4.1. Implementación del prototipo	73
4.2. Métricas orientadas a la función	78
4.3. Métricas de mantenimiento	79
4.4. Portabilidad	80
4.5. Adaptabilidad	80
4.6. Prueba del sistema tutor SQL	80

CAPÍTULO 5: Conclusiones y Recomendaciones

5.1. Estado de la hipótesis	84
5.2. Recomendaciones y futuras líneas de investigación	84

BIBLIOGRAFIA

ANEXOS

DOCUMENTACION



Lista de tablas

	Páginas
Tabla 2.1: Paralelismo entre la concepción de Piaget y la de Vigosky	24
Tabla 3.1: Escala de evaluación	52
Tabla 3.2: Funciones Principales del Sistema	60
Tabla 4.1: Cuestionario preguntas respuestas	81
Tabla 4.2: Recuento de respuestas	81
Tabla 4.3: Porcentaje total	82
Tabla A.1: Calculo Punto Función	Anexo b
Tabla A.2: Características ambientales del sistema	Anexo b



Lista de figuras

Páginas

Figura 2.1: Relación (IA), la Ingeniería del conocimiento y otras aéreas	
Figura 2.2: Funciones de un agente	31
Figura 2.3: Esquema de funcionamiento de un agente	32
Figura 2.4: Propiedades de un agente	33
Figura 3.1: Arquitectura básica de un STI	44
Figura 3.2: Módulos y submódulos del STI con módulo evaluador adicional	49
Figura 3.3: Submódulo de Protocolos Pedagógicos	50
Figura 3.4: Arquitectura del tutor	54
Figura 3.5: Interrelación entre componentes del tutor	55
Figura 3.4: Diagrama de casos de uso	65
Figura 3.5: Diagrama de Secuencia: Acceso al Sistema Tutor	67
Figura 3.6: Diagrama de Secuencia: Actualizar Temas	67
Figura 3.7: Diagrama de Estados	68
Figura 3.8: Diagrama de Colaboración para el caso: Registro de estudiante.	69
Figura 3.9: Diagrama de Colaboración: Solicitud de Avance y Seguimiento	70
Figura 3.10: Diagrama de Clases principal del Sistema	70
Figura 3.11: Diagrama de diseño lógico del Sistema	71
Figura 4.1: Pantalla Inicial del Sistema	74
Figura 4.2: Pantalla selección de tareas a realizar	74
Figura 4.3: Índice da la curricular	75
Figura 4.4: Pantalla de temas	75
Figura 4.5: Botones de navegación	76
Figura 4.6: Pantalla consola de sqlplus	77
Figura 4.7: Selección de evaluación	77
Figura 4.8: Evaluación al estudiante y los resultados	78

[CAPITULO 1]

INTRODUCCION

“Lo último que uno sabe
Es por donde empezar”

Blaise Pascal

1.1. Introducción

El incesante crecimiento e introducción de tecnologías, nos hace pensar e imaginar una forma de aprendizaje más efectiva, puesto que la utilización de herramientas informáticas se hacen necesarias para dirigir las al campo educacional.

El Sistema Tutor ofrece un tratamiento completo, paso a paso del lenguaje SQL, que va desde consultas simples hasta los conceptos más avanzados, le dará una perspectiva, sobre el impacto que el lenguaje está teniendo en todos los segmentos del mercado informático.

La motivación de este trabajo radica en la dificultad que tienen los alumnos que cursan la asignatura de base de datos de la carrera de informática de la Universidad Mayor de San Andrés.

Con el presente trabajo, se pretende aportar con nuevas herramientas de aprendizaje, en este caso especial para aprender todo lo relacionado al lenguaje SQL.

El mundo competitivo en la actualidad, ha definido un nuevo escenario en los métodos de estudio, obligando en muchos casos, al establecimiento de vínculos entre distintas tecnologías, donde el intercambio de información resulta de vital importancia, el uso de estas tecnologías es una aproximación de solución a la mejor comprensión del tema en interés.

Para la elaboración de este Sistema Tutor se hace uso de herramientas y metodologías, que aporta la informática y se verán en estrecha relación con otras ciencias como son la pedagogía.

1.2. Antecedentes

Desde que se fueron desarrollando nuevas tecnologías informáticas, el uso de la computadora se ha tenido muy en cuenta, y una ascendente aplicación ya que en principio solo se la utilizaba como herramienta de cálculo, editor de texto, como procesador de palabras, archivos, programas que hasta el día de hoy combinan texto, sonido e imagen gracias a tecnologías multimedia.

La facilidad que brindan estas tecnologías permite desarrollar diversas aplicaciones, es así que en este campo fascinante de la educación se ha desarrollado sistemas, como ser sistemas de instrucción basadas en computadoras (Computer Based Instruction CBI). Como el más reciente, la instrucción asistida por computadoras (Computer Aided Instruction CAI). Lo que dio origen a los sistemas tutores inteligentes (Intelligent Tutorial System STI). La instrucción asistida por computadoras evoluciono hacia los sistemas de tutores inteligentes pasando pruebas de inteligencia: El dominio de la aplicación, que debe ser entendido por el Sistema Tutor, lo suficiente bien para que este experto convertido haga inferencias o resuelva problemas en el dominio. El sistema debe ser capaz de deducir a una aproximación del educando que aprende ese conocimiento. La estrategia tutorial o pedagógica debe ser inteligente en el sentido que el instructor pueda implementar estrategias para reducir la diferencia entre la habilidad del experto y el estudiante.

La inteligencia artificial aplica enseñanza formal por medio de Sistemas Tutores Inteligentes (STI), que según [Millan, 05], se definen como; “un sistema que posee un amplio conocimiento sobre cierta materia y cuyo objetivo es transmitir este conocimiento al estudiante por medio de un proceso interactivo personalizado”. Otros la definen como “Aquel sistema capas de ayudar al estudiante en el aprendizaje de diversos conocimientos” [Vernet, 05].

Un tutor inteligente, *“es un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo”* [Van Lehn, 88].

En el transcurso del tiempo se han ido desarrollando sistemas tutores inteligentes dentro de nuestro medio, como es la Carrera de Informática y entre algunos de ellos citamos:

“Tutor inteligente de Auto aprendizaje en Matemáticas” [Arratia-Saravia, 03], que combinan los agentes inteligentes, con los tutores inteligentes, para obtener mejor rendimiento en el aprendizaje de las matemáticas en secundaria.

“Sistema Tutor para la Enseñanza de la Ortografía” [Delgado, 03], adapta la enseñanza tradicional a un sistema tutor inteligente para mejorar el proceso de enseñanza-aprendizaje de la ortografía.

“Interfaz de aprendizaje mediante tutores inteligentes” [Medrano, 05], quien combina un entorno de realidad virtual y un modelo de tutor inteligente utilizando el modelo o arquitectura de un sistema tutor inteligente tradicional.

“Sistema Tutor para la Enseñanza en lengua Aymara” [Quisbert, 06], esta es una herramienta que adapta la enseñanza tradicional a un sistema tutor inteligente para mejorar el proceso de enseñanza-aprendizaje de los alumnos de habla Aymara. Que va dirigido a niños del área de la educación rural Y otros.

1.3. Planteamiento del problema

La creciente necesidad por automatizar la educación, en la actualidad, la informática como ciencia es una herramienta que apoya a las demás ciencias por todo el caudal de facilidades que ofrece, por lo mismo la educación no es una excepción.

A partir de un esquema básico, de interacción de los estudiantes con los docentes, se estudian las diferentes características de los agentes para llevar a cabo una sesión de aprendizaje. *Se propone un modelo de agente docente en el módulo tutor*, y se reconoce a la herramienta a utilizar para el diseño de este sistema en el campo de los agentes, a fin de continuar con la siguiente etapa en la investigación.

Se ha observado que los Sistemas Tutores y Asesores Inteligentes disponibles en la actualidad no han encontrado soluciones eficientes en referencia a la flexibilidad en cuanto a modalidades de enseñanza, aunque en los últimos años, con los avances de la psicología cognitiva [Gardner, 03], las neurociencias y los nuevos paradigmas de programación, los sistemas facilitadores de la enseñanza evolucionaron desde una propuesta instructiva hacia entornos de descubrimiento y experimentación del nuevo conocimiento [Bruner, 90] desde una visión constructivista de los procesos de aprendizaje. Es decir, se pasa de una postura conductista con base en la teoría de Skinner a otra centrada en la psicología cognitiva.

En un Sistema Tutor Inteligente (STI), el modelo del tutor es el encargado de definir y de aplicar una estrategia pedagógica de enseñanza (de tipo socrática, orientador, etc.), de contener los objetivos a ser alcanzados y los planes utilizados para alcanzarlos. Es el responsable de seleccionar los problemas, de monitorear y de criticar el desempeño, de proveer asistencia cuando se la requiera y de seleccionar el material de aprendizaje al estudiante. Integra el conocimiento acerca del método de enseñanza (ya sea deductivo, inductivo, analógico, analítico, sintético, de trabajo colectivo, etc.), las técnicas didácticas (expositiva, discusión, demostración, diálogos, instruccional, responder preguntas, etc.) y del dominio a ser enseñado.

Un sistema de este tipo debe tratar además, los aspectos esenciales del curricular y de la planificación, ya que los aspectos del curricular, involucran la representación, la selección y la secuenciación del material a ser utilizado y la planificación se refiere a cómo ese material va a ser presentado. En general, tienen una muy buena representación en forma explícita del dominio pero no siempre la curricular está dado en forma explícita, es decir desde la visión y la secuenciación, lo que se obtiene es un bajo rendimiento.

Esta selección y secuenciación de la curricular para los estudiantes requiere del uso de mecanismos de planificación sofisticados que deben tener en cuenta la teoría de tutorizado empleada haciendo hincapié en las necesidades del estudiante.

Aún hoy día, la mayoría de los desarrolladores de programas “didácticos” se basan en modelos instruccionales de corte conductista [Perkins, 95] para entrenamientos. En este sentido, la intención de la investigación es emular a un tutor humano, pero orientado hacia la psicología cognitiva, es decir, teniendo en cuenta como señala Perkins [Perkins, 95] los estilos más apropiados de enseñanza tales como la instrucción didáctica, el entrenamiento y la enseñanza socrática.

1.3.1. Problema general

A través de investigaciones previas [Cataldi et al., 00] se ha observado que existen diversos factores que inciden en el bajo rendimiento académico de los alumnos en programación básica, algunos de ellos concernientes al docente y otros al alumno que permiten arribar a la problemática que se tiene hoy en día en las aulas.

Los problemas relativos al docente incluyen falta de capacitación en didáctica de los ayudantes nuevos, que se integran al plantel docente quienes deberían estar muy atentos a las fallas recurrentes de los alumnos y la detección de sus errores comunes: a veces se utilizan técnicas poco efectivas para revertir estos problemas, con una retroalimentación débil del estudiante hacia el docente. Esto se agrava en los casos del dictado de clases para grandes cantidades de estudiantes donde no se puede satisfacer las demandas particulares de cada uno de ellos por falta de tiempo y de recursos adecuados al curriculum, entre otros.

Por otra parte, existen problemas en el proceso educativo que pueden atribuirse al estudiante y que incluyen, desde algunas actitudes negativas hacia el aprendizaje hasta el desconocimiento de metodologías para evaluar las soluciones propuestas por los estudiantes y las formas para optimizarlas.

Debido a la problemática planteada se piensa, en sentar las bases para un sistema de aprendizaje por refuerzo; ya que justamente el objetivo buscado es que el estudiante le encuentre significado a sus aprendizajes, que supere sus dificultades, incorporando lo nuevo de manera que sea significativo y permanente. [Ausubel *et al*, 83].

En la literatura analizada se han encontrado dos posturas para la implementación de los conocimientos: una se basa en la estructura sintáctica de lo producido por los tutores humanos [Seu Jai *et al.*, 91] y la otra en las metas pedagógicas que deben cumplir a *La Teoría Uno a uno*, no es un modelo, ni un método de enseñanza, sino un conjunto de recomendaciones compatibles con cualquier teoría. Ella estipula que "*La gente aprende más cuando tiene una oportunidad razonable y una motivación para hacerlo*" [Perkins, 95].

1.3.2. Problemas específicos

Si bien no está dentro de los objetivos de este trabajo el desarrollo de nuevos marcos metodológicos para mejorar la interacción entre el docente y el alumno en el ámbito de la enseñanza, se busca utilizar modelos y técnicas educativas probadas.

Esta investigación se inscribe dentro de los llamados Sistemas Tutores Inteligentes (STI) y se trata que todas las consideraciones tomadas para un modelar un tutor inteligente aplicado al dominio del aprendizaje de Programación Básica puedan ser utilizadas en otros dominios sin realizar modificaciones importantes en el modelo planteado. Para ello, es necesario realizar una redefinición de los módulos que componen un Sistema Tutor Inteligente (STI) y de las interfaces entre estos módulos, identificando cuáles son los generadores de los datos y cuáles son los que los utilizan. Aquí se debe realizar un aporte dentro del marco teórico global de los (STI) para poder resolver el problema planteado. Por lo tanto, se tiene un problema que presenta características prácticas y teóricas, las que se pueden describir como:

- **Características Prácticas:** Son las que incluyen el desarrollo de los distintos submódulos del módulo del tutor de un Sistema Tutor Inteligente (STI) que se adapte a la definiciones previas como la de Carbonell [Carbonell, 70]. Además se deberán tener en cuenta los métodos de enseñanza más adecuados para la Programación Básica, así como también los instrumentos para verificar su validez y confiabilidad. Se deben poder seleccionar los protocolos pedagógicos más eficientes para cada sesión pedagógica a fin de impartirla del mejor modo posible con los recursos didácticos disponibles. Éstos se pueden obtener del módulo del estudiante (características del estudiante) y del módulo del dominio (que posee la información concerniente a los conceptos que se utilizarán en la lección, así como también las imágenes, ejercicios, videos, etc.).
- **Características Teóricas:** Se deberá redefinir el modelo clásico de los Sistemas Tutores Inteligentes propuesto por Carbonell [Carbonell, 70] para detectar en la superposición de funcionalidad de los módulos básicos, modificando los subcomponentes de esta estructura para lograr un Sistema Tutor Inteligente (STI) que sea independiente del dominio. Se deben

incorporar instrumentos para medición de rendimiento académico de los estudiantes, tales como evaluaciones, cuestionarios y ejercitación que conviertan al Tutor Inteligente en una herramienta diagnóstica con un valor educativo que permita mejorar la interacción entre el estudiante y el tutor. También se deben seleccionar y adecuar las herramientas pertenecientes al campo de la Inteligencia Artificial que mejor solucione la problemática planteada, realizando un análisis de los beneficios y los costos que la introducción de estas técnicas pueden presentar respecto de las estructuras clásicas y rediseñadas de los Sistemas Tutores Inteligentes.

De esta manera se resolverá un problema teórico-práctico de carácter interdisciplinario, basado en un marco teórico educativo y en las herramientas informáticas que aporta la Inteligencia Artificial (IA) que pueden soportar estas teorías.

1.4. Objetivos

1.4.1 Objetivo general

Proponer un modelo de tutor inteligente capaz de adaptarse a las necesidades y preferencias de los estudiantes, según sus estilos de aprendizaje, para proveerles de una herramienta adicional a través de la cual puedan recibir asesoría en el aprendizaje, independientemente del tiempo y la distancia en que éstos se encuentren.

1.4.2. Objetivos específicos

- Determinar el estado actual de las investigaciones en el tema del tutorizado humano e inteligente, relevantes para la tarea que desempeña el módulo correspondiente al tutor en un Sistema Tutor Inteligente (STI).
- Establecer los fundamentos que sustentan las teorías seleccionadas.
- Seleccionar los componentes del módulo del tutor, mediante el uso de submódulos, que permitan guiar a los estudiantes en el proceso de aprendizaje de manera de poder responder a sus necesidades.
- Modelar el subsistema tutor enmarcado dentro de la arquitectura de los Sistemas Tutores Inteligentes (STI), con sus submódulos e interfaces, utilizando las herramientas que provee la Ingeniería de Software y la Inteligencia Artificial (IA).
- Validar el modelo planteado con datos característicos de los estudiantes, tales como los estilos de aprendizaje.
- Evaluar los cambios ocurridos en el desempeño del estudiante, como consecuencia directa de la utilización del tutor

1.5 Hipótesis

“El Sistema Tutor para la enseñanza del lenguaje SQL, contribuirá a mejorar el nivel de aprendizaje, en los estudiantes de taller de bases de datos”.

1.6. Justificación

El aporte del presente trabajo permitirá esclarecer algunos interrogantes respecto de cómo enseñan los tutores humanos y podría dar solución a los estudiantes de clases muy numerosas, que no pueden acceder al docente durante el horario habitual de clases o por un sin fin de complicaciones.

1.6.1. Social

Mediante el crecimiento de la tecnología, se ha definido un nuevo escenario en los medios de educación, obligando, en muchos casos, al establecimiento de vínculos de distintos métodos de enseñanza. Se ha observado que en los cursos universitarios de los primeros semestres de las carreras de Informática, que la cantidad de tutores humanos no es suficiente, donde la relación estudiantes/docentes es muy alta y que existe una gran diferencia en el nivel de conocimientos previos que traen los alumnos. Se piensa que un sistema que pueda emular al tutor humano y que además provea al estudiante de cierta flexibilidad para la elección del tipo de tutorizado más adecuado, podría ser una solución factible para el problema planteado.

1.6.2. Técnico

El desarrollo del presente trabajo se justifica técnicamente, por el uso de la tecnología de computadoras que permita apoyar mejor la enseñanza en el área de manejo de Base de Datos. Siendo el principal apoyo del computador su interactividad, pudiendo en estos modificarse los contenidos programáticos de

acuerdo a normas curriculares y adecuarse a los estudiantes, estos son de fácil adaptación para los estudiantes, y están destinados a complementar la enseñanza e incrementar la práctica. El sistema tutor incluye programas de autoevaluación.

1.6.3. Económica

Se pretende alcanzar los objetivos tratando que el sistema tutor sea asequible por todos los estudiantes, tratando de no invertir gastos económicos, cabe señalar que el tutor reduce gastos de espacio como de tiempo, para el aprendizaje como para el mantenimiento, manipulación y ejecución del entorno.

1.7. Metodología

El desarrollo del presente trabajo de investigación se lo llevó adelante en orden de las siguientes actividades:

- a) En la etapa inicial se centra en la revisión de la literatura existente en la temática de: modelado del tutor, protocolos de enseñanza, Sistemas Tutores Inteligentes (STI) en general, así como también se relevarán las posibles herramientas que provee la Inteligencia Artificial (IA) para poder llevar a cabo el modelo planteado. Esta etapa finaliza con la elaboración de un informe que constituye el Marco Teórico, donde se resumen las investigaciones más relevantes en el tema y el estado actual de las herramientas, métodos y técnicas disponibles. Aquí se describen también los fundamentos teóricos que cimentarán la solución elegida a través de la investigación de libros, revistas especializadas, sitios de Internet de centros de investigación de universidades

nacionales e internacionales y eventos de la temática (Simposios, Congresos, Jornadas, Workshops, etc.).

- b) Consiste en el diseño y elección de las herramientas de toma de datos (planillas de estilos de aprendizaje). De acuerdo con Hernández Sampieri [Hernández Sampieri, 01] se determinará la validez y confiabilidad de la herramienta utilizada.
- c) se definirá la arquitectura básica del Sistema Tutor Inteligente (STI) que permita independizar las funciones básicas de los módulos y submódulos que componen a todo el sistema, evitando la superposición de funcionalidad entre los distintos módulos.

Para la elaboración del modelado del prototipo se hará uso:

- para modelar el prototipo del sistema tutor la implantación y pruebas se realizarán en UML (Unified Modeling Lenguaje), herramienta case que permite modelar, construir, visualizar y documentar sistemas de software orientado a objetos.
- También consta de un banco de pruebas que evaluaran al estudiante
- Técnicas de análisis y diseño de datos para el almacenamiento y procesamiento de los datos que se generen.
- Implementaremos un modelo de base de datos relacional, con datos relacionados, para obtener resultados en las consultas de prueba.

Los recursos informáticos a utilizar para la implementación del sistema tutor son:

- Un ordenador
- Lenguajes de programación

- Un servidor de base de datos donde esté instalado el SQL

1.8. Alcance y límites

La presente investigación se orienta a la implementación de un modelo tutor inteligente, denominado Sistema Tutor para la enseñanza del lenguaje SQL. Más allá de las condiciones que se plantea en la presente Tesis cabe hacer notar de sus alcances y limitaciones.

Este sistema tutor permitirá dar respuesta de una forma sistemática y ordenada, a los distintos requerimientos que tengan los estudiantes que deseen practicar y evolucionar sobre el manejo del lenguaje en cuestión, consultas y diseño de una base de datos relacional, que además este tutor esté orientado a estudiantes con o sin experiencia en este campo. Al finalizar con la presentación de un determinado tema, es sistema tutor evaluará al proceso de aprendizaje del estudiante para medir su grado de aprovechamiento.

Es importante aclarar que el tutor inteligente no va a sustituir al docente que enseña al estudiante el lenguaje en sí, se trata de una herramienta que incorpora un agente inteligente para que el estudiante en sus horas libres de practica le sirva de ayuda para poder facilitar su tarea y mejorar su proceso de aprendizaje del lenguaje en cuestión.

También se pretende dar cobertura a personas que estén interesados en un curso de programación ayudándoles a desarrollar sus habilidades de utilización del lenguaje, el tutor está elaborada de tal manera que al finalizar con el proceso de enseñanza de cierto tema, se presentara una evaluación que el estudiante debe

realizar, con el fin de cimentar sus conocimientos adquiridos y al mismo tiempo poder dar una mejor guía al docente instructor y a su vez tutorar a los estudiantes que tengan problemas en ciertas áreas de programación.

También se pedirá que en el momento de avance de los temas, el estudiante tenga que tener algún software de gestor de base de datos (Servidor) instalado en el computador, donde este conectado SQL, para poder practicar o plasmar código, poder ver resultados en tiempo real y ganar experiencia practicando.

1.9. Aportes

El Sistema tutor será de gran manera un aporte muy importante en la educación, ya que contribuirá a mejorar el proceso de enseñanza-aprendizaje. Donde utiliza herramientas, metodologías, etc. Que proporciona la informática para elaborar un software, que irá en directo beneficio a la sociedad puesto que, la sociedad demanda herramientas educativas más flexibles menos tediosas y asequibles que sean fáciles de obtener menos costosas.

Disponer de una herramienta para la enseñanza y proporciona soluciones en un dominio específico del área mediante una base de datos.

El principal es apoyar la labor del docente, porque reduce el trabajo de tener que repetir un tema que el estudiante no haya entendido.

Proponer modelos para la creación de sistemas tutores inteligentes aprovechando tecnologías de la Inteligencia Artificial (I.A.).



[CAPITULO 2]

MARCO TEORICO

2.1. Introducción

Bendfeldt [Bendfeldt, J., 94] cita que: “Un error en la práctica de la medicina puede poner en peligro una vida. Un error en la práctica de la política puede poner en peligro una generación. Más un error en la práctica de la enseñanza puede poner en peligro a miles de generaciones”. Aunque lo pongo muy exagerada, sí da una visión de los efectos de la mala práctica en la enseñanza.

Definimos un marco teórico que sustente el diseño y la evaluación de los STI, con base en la ingeniería de software, los sistemas inteligentes, la psicología cognitiva y las ciencias de la educación y elaborando una extensión metodológica específica que cautele los aspectos inherentes para el diseño de STI orientados al tutorizado.

El uso de la tecnología y la aplicación de agentes inteligentes en ambientes de aprendizaje nos conducen a un modelo de enseñanza-aprendizaje en el cual los alumnos asumen un rol más activo. De forma general se observa que los alumnos que trabajan en entornos de aprendizaje potenciados con agentes tutores demuestran mejorar en las diferentes áreas de conocimiento, aprendizaje cooperativo, pensamiento crítico, pensamiento creativo, resolución de problemas, comportamiento social.

En los últimos años las tecnologías han evolucionado muy rápidamente y la Inteligencia Artificial (IA) surge como una de las ramas de estudio más recientes y promisorias en el campo de las ciencias de la computación. Por todo ello existen aplicaciones, lo cual torna a este campo en un área interesante para los investigadores, alumnos y administradores que puedan utilizar en forma directa los resultados de las investigaciones.

Se observa que cada autor da una definición de la inteligencia: desde la enunciación muy general de Maturana que la ve como “un atributo o propiedad distintiva de organismos” [Maturana, 98], pasando por investigaciones como Piaget [Piaget, 89] que plantea que la inteligencia “es la capacidad de adaptación de un organismo a una situación nueva”.

Es aquí donde la investigación sobre Inteligencia Artificial (IA) intenta asimilar estas definiciones dentro de sus propias estructuras, incorporándolas a los Sistemas tutores inteligentes (STI) que contemplan el aprendizaje humano y como la enseñanza con base pedagógica. Es por ello, que el objetivo de este capítulo es presentar un marco conceptual, que dé lugar al marco teórico de la Inteligencia Artificial (IA) y los Sistemas Tutores Inteligentes (STI).

2.2. Inteligencia Artificial

A diferencia de la filosofía y la Psicología, que trata de entender cómo funciona la Inteligencia en abstracto, Inteligencia Artificial (IA) es un intento por descubrir y aplicar los aspectos de la inteligencia humana que pueden ser estimulados mediante desarrollo, la Inteligencia Artificial (IA), ha presentado productos sorprendentes en sus aplicaciones [Stuart et al., 95].

En la actualidad, el campo de la Inteligencia Artificial (IA) en marca varias subáreas tales como los sistemas expertos, la demostración automática de teoremas, el juego automático, el reconocimiento de la voz de patrones, el

procesamiento del lenguaje natural, la visión artificial, la robótica, las redes neuronales, etc [Castillo et al., 98]. En la fig. 2.1 se puede observar los subcampos dentro de la Inteligencia Artificial (IA) [Garcia-Martines et al., 03].

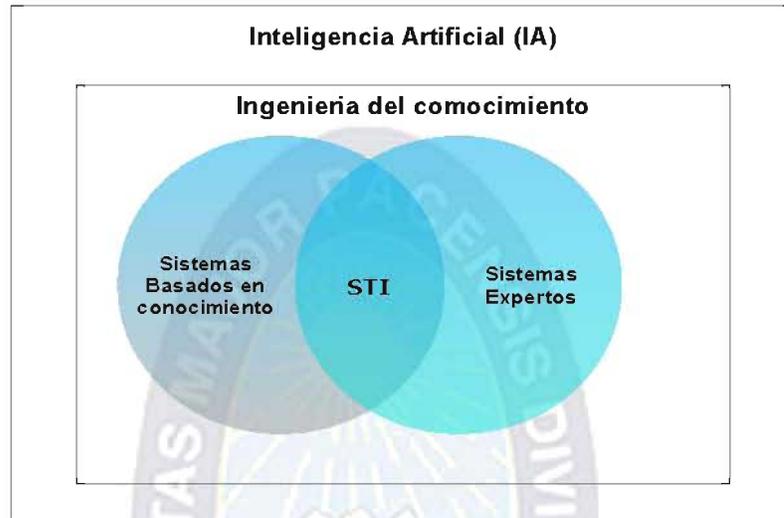


Fig. 2.1 Relación IA, la Ingeniería del conocimiento y otras aéreas
[Garcia-Martines et al., 03].

La Inteligencia Artificial (IA) surge así como una disciplina cuyo objetivo es proveer técnicas para el desarrollo de programas capaces de simular la inteligencia que utilizan los humanos para solucionar problemas en una gran cantidad de dominios [Krishnamoorthy et al., 96], por lo que la (IA) provee un conjunto de formalismos que pueden representar los problemas, las herramientas y técnicas para resolverlos. Según diversos autores [Krishnamoorthy et al, 96; Newell, 69], las actividades esenciales asociadas con la inteligencia son:

- Responder de manera flexible a una gran variedad de situaciones.
- Dar sentido a los mensajes contradictorios y/o ambiguos.
- Reconocer la importancia relativa de los diferentes elementos de la situación problemática planteada.
- Encontrar similitudes entre situaciones, sin importar las diferencias que las separan y puedan parecer.

Por su amplio contenido, es difícil definir a la Inteligencia Artificial (IA), pero resulta de interés para esta investigación un concepto esclarecedor, por lo que se expondrán diferentes posiciones de la (IA).

Si bien existen muchas definiciones de la Inteligencia Artificial (IA) en la que el autor la presenta de una manera ligeramente diferente, aquí se resumirán las más representativas, agrupadas como la propone Stuart [Stuart et al., 95], en dos categorías.

- Las que conciernen a los procesos de pensamiento y razonamiento.
- Las que conciernen al comportamiento.

En general, estas definiciones miden el éxito de la implementación de herramientas de la Inteligencia Artificial (IA) de dos maneras diferentes:

- **En términos de “Racionalidad” o concepto ideal de inteligencia:** Como establece Stuart [Stuart et al., 95] quien define a un sistema como racional “si hace lo correcto”, es decir, una acción a la que llega por medio de procesos lógicos medibles.
- **En términos de performance humana:** Es decir, capacidad de resolución de problemas, capacidad de razonamiento abstracto, etc.

Todo lo anterior nos permitirá observar cómo van surgiendo las nuevas herramientas y aplicaciones que facilitarán los diseños de los Sistemas Tutores Inteligentes (STI).

2.3. Sistemas Tutores Inteligentes

Guardia Robles [Guardia Robles, 93] resume un conjunto de características que deben cumplir todos los Sistemas Tutores Inteligentes (STI):

- Deben ser inteligentes en comparación con los sistemas tradicionales de

instrucción por computadora (CAI), siendo el diferencial de inteligencia los métodos de la rama de la Inteligencia Artificial (IA).

- Deben poseer la capacidad tanto para resolver el problema que se le presenta a un alumno como también la capacidad de explicar cómo le resolvió.
- Como en los (CAI), tradicionales permite una mayor individualización en la instrucción, llegando más lejos, a través del entendimiento de las metas y creencias del alumno.
- Se usan técnicas de (IA) para la planeación, optimización y búsquedas, dejando que el sistema decida el orden de presentación del contenido al estudiante.
- La interacción puede ser muy variada en un (STI): desde sistemas pasivos (que esperan para que el alumno realice una acción), hasta los que constantemente presentan nueva información (tutor oportunista), con casos intermedios en los que se enseñan un concepto en un momento determinado solo cuando el alumno lo pide.
- Utilizan nuevas tecnologías, con los ejemplos de interfaces orientados a la utilización de multimedia y del WWW.
- No basta con indicarle un error al alumno, el sistema debe hacer hipótesis basadas en el historial de errores del alumno y detectar la fuente del problema.

Con las consideraciones presentes Guardia Robles [Guardia Robles, 93], presenta una definición para los Tutores Inteligentes “Un Sistema Tutor Inteligente es un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza; y capaz de comportarse como un experto, tanto en el dominio del conocimiento que enseña (mostrando al alumno como aplicar dicho conocimiento), como el dominio pedagógico, donde es capaz de diagnosticar la situación que se encuentra el alumno y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.”

Villareal [Villareal, 03] plantea que los Sistemas Tutores Inteligentes (STI), simulan a un tutor autoritario que posee una estrategia de enseñanza de los conceptos del dominio del tipo uno a uno. Además es un experto en un dominio de conocimiento determinado y actúa como guía, tutor o entrenador. Este tutor debe poder adaptarse a las necesidades, que surgen a lo largo de la interacción en una sección de tutelado, del alumno.

2.4. Teorías de Aprendizaje-Enseñanza

Desde el punto de vista de la psicología existen tres enfoques que hacen a la teoría del aprendizaje.

2.4.1. La teoría conductiva: Según Torrealva [Torrealva, J., 04]. "Se concentra en el estudio de conductas que se pueden observar y medir". Ve a la mente como una "caja negra" en el sentido de que las respuestas a estímulos se pueden observar cuantitativamente ignorando totalmente la posibilidad de todo proceso que pueda darse en el interior de la mente. Indica que el proceso de aprendizaje es pasivo, en el sentido de que el alumno no lo construye, la figura del profesor tiene mucha relevancia y aquí es quien define lo que debe aprenderse y como aprenderlo.

2.4.2. La teoría cognoscitiva: Según Mergel [Mergel, E., 05]. La teoría cognoscitiva reconoce la importancia del reforzamiento, pero enfatiza su papel como elemento retro alimentador para la corrección de respuestas sobre su función como un motivador, el cognoscitivismo postula que el conocimiento es interno al sujeto y que este conocimiento no existe como *product*, sino que es el sujeto quien lo construye, pero esto no implica que todo el conocimiento sea subjetivo, evitando este último a través del concepto de consensos. Existe el consenso individual, interno al aprendiz donde se ve la consolidación del conocimiento y el consenso externo que se logra vía negociación social con los otros aprendices y el profesor por tanto el proceso de aprendizaje es activo.

2.4.3. La teoría constructiva: según Torrealva [Torrealva, J., 04]. Esta teoría postula que el aprendizaje es un proceso activo, en el cual los aprendices construyen nuevas ideas o conceptos, basados en sus conocimientos anteriores, así el aprendiz selecciona y transforma la información, construye hipótesis, toma decisiones y en este proceso las estructuras cognitivas proveen significado y organización a las experiencias. Tanto docente como alumno deben comprometerse en un diálogo activo, en el que la tarea del docente consiste en introducir la información, según el estado de la comprensión del aprendiz.

Existen diferentes teorías de que explican el proceso de aprendizaje, que se pueden agrupar en:

- **Instruccional:** El profesor o tutor es el centro y el encargado de diseñar el contenido, relegando al alumno a la función de “aprender” el conocimiento, este proceso de aprendizaje es altamente guiado por el docente y no requiere la iniciativa personal del alumno.
- **Aprendizaje por descubrimiento:** Es donde el profesor tiene una tarea de “explorador”, entregándole al alumno las herramientas necesarias para que este descubra el conocimiento a su propio ritmo. Este proceso de aprendizaje no es guiado y requiere de iniciativa por parte del alumno, ya que es este el encargado de “descubrir” el conocimiento por sus propios medios, guiado por el tutor para que se puedan lograr los objetivos que surgen de la sesión pedagógica, pero dando lugar a nuevos objetivos que surgen de la experiencia del alumno.

Los procesos de aprendizaje fueron tema de estudio desde siempre. Las primeras reflexiones las expone Platón, cuando en su libro La República [Platon, 01], explica el mito de las cavernas y aclarando que: “el mundo que se conoce es la proyección de nuestras vidas innatas”. A eso se lo conoce como pensamiento racionalista, por otro lado Aristóteles, rechaza la teoría de las ideas innatas y establece que el conocimiento procede de los sentidos que dotan a la mente de

imágenes que se asocian entre sí según tres leyes: contigüidad, similitud y contraste, iniciando así el pensamiento empirista. Estas teorías son refutadas, por Santo Tomas de Aquino.

Teoría de la instrucción de Gagné

El aprendizaje se define como un proceso que permite a un organismo vivo modificar sus comportamientos en forma rápida y permanente; el aprendizaje se verifica cuando existe un cambio de comportamiento, relativamente estable. Esto supone cuatro elementos: un aprendiz, una situación o entorno que permite el aprendizaje, un comportamiento explícito del aprendiz y un cambio interno.

Esta teoría establece que existen varios tipos de aprendizaje y que por lo tanto son necesarios diferentes modos de instrucción. Estos tipos son: información verbal, destrezas intelectuales: discriminar, formar conceptos, aprender reglas, estrategias cognitivas, actitudes y destrezas motoras.

Para cada tipo de aprendizaje son necesarias diferentes condiciones internas y externas. Una vez que plantea estos factores que afecta al aprendizaje, Gagné puede plantear al proceso de aprendizaje como un sistema, donde dichas entradas afectan la salida; y por lo tanto, el profesor puede mejorar la salida controlando las entradas. Gagné sugiere agregar el concepto de jerarquía como base para su descripción del funcionamiento del sistema.

Teoría de Equilibrio de Piaget

Para Piaget [Piaget, 89], existen dos tipos de aprendizaje, uno el aprendizaje en sentido estricto, a través del cual se consigue información específica y otro, el aprendizaje en sentido amplio, que consiste en el progreso de las estructuras cognitivas. El aprendizaje se produce cuando se presenta un desequilibrio o conflicto cognitivo que da lugar a dos procesos: la asimilación y al acomodación.

La asimilación es la acomodación de elementos exteriores a estructuras en evolución o ya acabadas en el organismo [Piaget, 89]. El individuo interpreta la información del medio de acuerdo a sus conceptos disponibles. A través del proceso de acomodación, el sujeto adapta sus conocimientos a la realidad en tanto y en cuanto contrasta con ella lo asimilado; por otra parte se produce el cambio de estructuras cognitivas del individuo en dos sentidos, el primero como consecuencia de la suma de nuevos conceptos y el segundo como consecuencia de una reinterpretación de la ya sabido a la luz de los nuevos conocimientos.

La teoría de Piaget reduce todo el aprendizaje a adquisiciones espontaneas y necesarias, esto es un fallo puesto que la mayor parte de los conceptos en realidad no son necesarios y además, no pueden adquirirse sin la intervención de la cultura y la instrucción, es decir no son espontáneos.

Teoría sociocultural de Vigotsky

Vigotsky [Vigotsky, 26], plantea que “estudiar las conductas del hombre sin lo psíquico, como pretender la psicología, es tan imposible como estudiar lo psíquico sin la conducta. Por tanto no hay lugar para ciencias distintas. El estado actual de las dos ramas del saber plantea insistentemente la cuestión de tal necesidad y fecundidad de la completa fusión de ambas ciencias”.

El aprendizaje sucede a partir de la interacción social, puede establecer que: el individuo reconstituye su saber debido a que se entremezclan procesos de construcción personal y procesos en colaboración con los otros que intervinieron. En este el maestro debe ser un agente facilitador y mediador de la cultura. En la tabla 2.1 se puede observar paralelismo entre la concepción de Piaget y la de Vigotsky.

Teoría de Piaget	Teoría de Vigotsky
El conocimiento es un proceso de interacción entre el sujeto y el medio entendido físico únicamente	El conocimiento es un proceso de interacción entre el sujeto y el medio social y culturalmente
El ser humano al nacer es un individuo biológico	El ser humano al nacer es un individuo social.
La potencialidad cognoscitiva del sujeto depende de la etapa del desarrollo en la que se encuentre.	La potencialidad cognitiva del sujeto depende de la calidad de interacción social del sujeto.
El ser humano al nacer se encuentra en un estado de desorganización que deberá ir organizando a lo largo de las etapas del desarrollo de su vida.	El ser humano al nacer tiene una percepción organizada puesto que está dotado para dirigirla a estímulos humanos y para establecer relaciones sociales.

Tabla 2.1: Paralelismo entre la concepción de Piaget y la de Vigotsky

Teoría del Andamiaje de Bruner

Esta teoría está relacionada con la de Piaget, ya que enfatiza el aprendizaje por descubrimiento. El alumno descubre, pues esto le causa mayor satisfacción y obtiene una mayor retención. Llega un paso más delante de la teoría de Piaget, desarrollando un proceso de instrucción donde el profesor debe crear un ambiente que favorezca lograr el descubrimiento [Bruner, 90]. Propone que los contenidos a enseñar se presenten como un conjunto de problemas y relaciones que el alumno debe resolver a fin de que se interese y resulte significativo el aprendizaje.

Bruner explica como el alumno llega a dominar los contenidos que el docente le propone construyendo el concepto de *andamiaje*, estrechamente ligado al de *zona de desarrollo próximo* de Vigotsky. Según esta idea, en el proceso de

interacción y diálogo y que se basa la enseñanza, el experto tiende un conjunto de andamios o ayudas por medio de las cuales el alumno elabora las construcciones necesarias para aprender los contenidos. Progresivamente, el maestro ajustará el sistema de ayudas y apoyos necesarios conforme se desarrollen las habilidades del alumno, y así poder ir cediendo el control y el manejo de los contenidos y formas de discurso a aprender.

Teoría del aprendizaje significativo de Ausubel [Ausubel et al., 83]

Trata de la adquisición de aprendizaje significativo, por lo tanto se considera el aprendizaje mecánico (de memoria). Para considerar que un aprendizaje es significativo, requiere tener un sentido para ser incorporado al conjunto de conocimientos que el sujeto posee. También considera el aprendizaje como receptivo, ya que es el profesor que establece los contenidos para facilitar la organización del nuevo contenido en la estructura mental del alumno. Un aprendizaje significativo si pueden incorporarse a las estructuras de conocimiento, es decir los nuevos conceptos adquieren significado para el sujeto dentro de sus estructuras cognitivas. En general un aprendizaje significativo es más eficaz que el memorístico puesto que: produce represión más duradera, facilitan nuevos aprendizajes y produce cambios que persisten más allá del olvido de los detalles concretos.

Otro aporte de Ausubel es el concepto de los organizadores previos, sobre los cuales el alumno se apoya, de tal manera que estos hacen las veces de estructura o andamio entre los conocimientos a adquirir y los que ya poseía.

Inteligencias múltiples de Howard Gardner

Esta teoría de la inteligencia humana, desarrollada por el psicólogo Howard Gardner [Gardner, 93], establece que al menos ocho formas en aquellas personas perciben y entienden el mundo. Llama a cada uno de estas formas inteligencias.

Define una inteligencia como “la capacidad de resolver problemas o de crear productos, quisieron valiosos en uno o más ambientes culturales”. Éste autor fundamenta su estructura en pruebas biológicas y antropológicas y más específicamente, en bases neurológicas, evolucionistas y transculturales. Presenta una calcificación de inteligencias, aunque sugiere que no es terminante:

- Verbal-lingüística
- Lógico-matemático
- Visual-espacial
- Kinestésica
- Musical rítmica
- Interpersonal
- Intrapersonal
- Naturalista

El objetivo de la escuela debe ser desarrollar las inteligencias y ayudar a la gente a alcanzar los fines vocacionales y aficiones que se adecuen a su particularidad espectro de inteligencias.

2.4.4. Técnicas Educativas Aplicables

Una vez presentadas las técnicas educativas más importantes, se debe analizar cuáles de ellas son aplicables a la enseñanza de lenguajes de programación con paradigma lineal estructurado y dentro de ellas, cuáles son implementables a un sistema tutor inteligente. Cualquiera de las teorías de enseñanza explicadas resulta útil para entender el proceso de aprendizaje de lenguajes de programación. La selección de un marco teórico adecuado sólo puede justificarse por medio de razones pedagógicas, correspondientes al tipo de alumnos con el que se va a enfrentar al sistema a las técnicas de programación o características disponibles para el futuro sistema [Perkins, 95].

Como no es el propósito de esta tesis de evaluar los distintos modelos de enseñanza y de aprendizaje, por lo que se selecciono la *Teoría uno* [Perkins, 95], que no es un modelo, ni un método de enseñanza, sino un conjunto de recomendaciones compatibles con cualquier teoría. Básicamente, estipula que *“la gente aprende más cuando tienen una oportunidad razonable y una motivación para hacerlo”*.

Por lo tanto se tomarán la Teoría Uno, como la base teórica del Sistema Tutor Inteligente (STI), en la que se toman las bases de las distintas formas de enseñanzas disponibles. Perkins plantea que si se combinan las condiciones que estipula la Teoría Uno se “personifican” las distintas maneras según el programa del momento. Las cuatro formas principales se enumeran a continuación:

- a) según Perkins es la representación clara y concreta de la información por parte de los maestros y de los textos. Su objetivo se centra especialmente en la explicación donde se exponen *“los que”* y *“los porque”* de un determinado tema. Es fundamental que se aclaren los componentes básicos de una buena explicación, ya que la instrucción didáctica el alumno queda relegado a un papel de sujeto pasivo y el tutor o maestro queda en primer plano de proveedor único de los saberes y conocimientos.
- b) El entrenamiento. Existe un vínculo entre el entrenamiento y la instrucción didáctica y por eso se representa en segundo lugar. Sin una instrucción didáctica que presente cierta base de información sobre los conceptos nuevos, los estudiantes carecerían de los conocimientos básicos para realizar las ejercitaciones prácticas. Se plantea una analogía muy clara con respecto al fútbol *“el entrenador observa desde afuera el desempeño de los deportistas y les da consejos. Elogia los puntos fuertes, detecta los débiles, hace observar ciertos principios, ofrece guía e inspiración y decide que todo tipo de prácticas se deben enfatizar”*. Y es ésta función tan importante en la que se extrapola y se aplica en cualquier dominio, será este fútbol, clase de matemática o literatura.

- c) **La enseñanza socrática.** Ambos métodos vistos con anterioridad poseen un aspecto regulativo, pues sus funciones consisten en modelar y guiar las actividades de los alumnos y cómo se explicó anteriormente se relega al estudiante a un papel secundario, en éste marco surge el método socrático, a través del cual se logra que los estudiantes trabajen de manera más flexible, pero no en forma libre sino que continuamente estén recibiendo el apoyo de sus investigaciones por parte de un maestro o tutor, pero sin el esquema más elegido en el que tutor les dice todo el tiempo lo que tienen que hacer. La enseñanza se realiza cuando el maestro socrático plantea un enigma conceptual e incita a investigar el asunto de manera libre por parte de los estudiantes, haciendo preguntas del tipo: ¿qué piensan al respecto?, ¿Qué posición se podría tomar?, ¿Qué definiciones necesitamos? Luego de la investigación inicial se proponen ideas, criterios y definiciones, donde el maestro actúa como incitador y moderador en la conversación. Uno de los conceptos claves de la teoría uno, es que el maestro socrático no provea abundancia de datos, pero controle la claridad y la calidad de la información suministrada por los alumnos haciéndole preguntas. Cuando los alumnos discuten entre sí sobre una determinada cuestión, el maestro socrático les exige una práctica un clima de reflexión, provee además, realimentación inmediata por medio de estímulos y críticas, e incita aquí todos los participantes de la conversación hagan lo mismo.
- d) **Otros métodos de enseñanza.** Los protocolos pedagógicos puestos en la teoría uno no son los únicos acercamientos que proveen las características requeridas por la teoría uno: “información clara, práctica reflexiva, realimentación informativa y fuerte motivación intrínseca y extrínseca”. Existen otras aproximaciones como pueda serlo el aprendizaje cooperativo y la colaboración entre partes, que vale la pena resaltar en el marco de los asesores inteligentes, donde el sistema debe reaccionar como un “estudiante virtual” que interactúan con los estudiantes usuarios del sistema para resolver los distintos tipos de problemas que se presentan y para lograr definiciones básicas de los conceptos que se pretende enseñar en una sesión pedagógica

a través del “andamiaje” [Ausubel et al., 83] , como lo hará un ayudante alumno.

Perkins sostiene que los niños aprenden mucho mejor en grupos cooperativos bien configurados que en soledad. Por lo general, las agrupaciones cooperativas pueden ayudar a lograr determinados fines, sin la planificación cuidadosa que reúne los métodos como la instrucción didáctica y el entrenamiento los resultados obtenidos son satisfactorios porque el aprendizaje cooperativo exige que todos los participantes se hagan responsables del desempeño del grupo.

2.4.5. Que es Enseñanza

“Si lo que ha de aprenderse evoluciona, y nadie duda de que evoluciona y cada vez a mayor velocidad, la forma en que ha de aprenderse y enseñarse también debería evolucionar” señala Pozo [Pozo, J., 96].

Almeida y colegas [Almeida, S. et al., 97] definen medio de enseñanza como todos los componentes del proceso docente que actúan como soporte material de los métodos (instructivos o educativos) con el propósito de lograr los objetivos planteados.

La enseñanza programada supone que los alumnos pueden aprender cierto material más efectivamente si está programado, o sea si está organizado lógicamente para presentar sólo una idea elemental cada vez, según explica Mateo [Mateo, A., 97]. Cada idea se construye a partir de las previas, cada nuevo elemento de información va seguido por una serie de preguntas y ejercicios con las correspondientes respuestas, de modo que el alumno pueda integrar y revisar las informaciones previas y vigilar por sí mismo la marcha del trabajo. El problema es preparar el material. También, según esta autora, resulta problemático la rigidez del orden prefijado. Se le podría agregar elementos tales que “si resolvió tal problema, realizar tal otro”, lo cual es sencillo de hacer con computadora.

2.4.5.1 Como definimos el proceso de enseñar

Es el acto mediante el cual el maestro o tutor suscita o muestra contenidos educativos (conocimientos, hábitos, habilidades, experiencias) a un alumno a través de medios, en función de unos objetivos y dentro de un marco contextual.

2.5. Las características de los agentes del STI

Un agente es reactivo si no incluye un modelo simbólico del mundo ni usa razonamiento simbólico complejo de ningún tipo y pueden construirse usando sistemas de producción, como propone Nilsson [Nilsson, 98] es decir definiendo conjuntos de reglas de producción y de producciones que indiquen la acción a realizar cuando se verifican determinadas condiciones. No existe comunicación explícita entre ellos, reaccionan a los cambios en el entorno generados por los otros agentes, a medida que los perciben.

Los agentes son deliberativos sí usan un modelo del mundo, explícitamente representado y funcionan siguiendo el paradigma de los sistemas clásicos de planificación en IA (Inteligencia Artificial), basados en el ciclo percepción-planificación-acción. Las intenciones se corresponden con la estrategia de acción que sigue el agente, y pueden ser reconsideradas. Según Rao y Georgeff [Rao y Georgeff, 95], un agente deliberativo lleva a cabo un ciclo infinito que censa sus creencias, considera sus deseos para establecer cuales se pueden alcanzar y los no alcanzables se eliminan, transformándose en intenciones, estableciendo planes de acción para la ejecución de las mismas, lo cual modificar nuevamente el entorno percibido por el agente, y su conjunto de deseos, dado que los alcanzados se eliminaran del conjunto.

Un agente es híbrido si dispone de componentes deliberativos para llevar a cabo razonamientos complejos, desarrollar planes y decidir, y simultáneamente tiene componentes reactivos para tener la reaccionar

inmediatamente ante ciertos sucesos. Se adopta para este trabajo que: tanto los Agentes Docente como los Agentes Alumno son híbridos. Por otro lado, los agentes que trabajan de forma independiente con sus propios objetivos constituyen sistemas independientes. Si tienen objetivos sin correlación son discretos, y en caso contrario son de cooperación emergente, cuando poseen mecanismos explícitos de cooperación. Esta categoría de SMA (sistema multiagente) puede considerarse como comunicativos cuando poseen y usan protocolos y procedimientos de cooperación, pero si usan medios indirectos, serán no comunicativos. El STI planteado es un sistema cooperativo y comunicativo. Dadas las características del sistema, el diseño puede imponer para cada agente un protocolo de interacción y una estrategia, los cuales se construyen en función de los resultados sociales, dado que los agentes deben utilizar las estrategias impuestas.

Para los planes consensuados, existen varios enfoques posibles. Para el caso del STI, se podría aplicar un enfoque de planeamiento multiagente, en el cual se establecen tres etapas, cada una llevada a cabo por uno o más agentes: elaboración de planes, coordinación y ejecución de los planes. Otro mecanismo posible es el de votación, en donde se somete a la decisión de los agentes las acciones posibles.

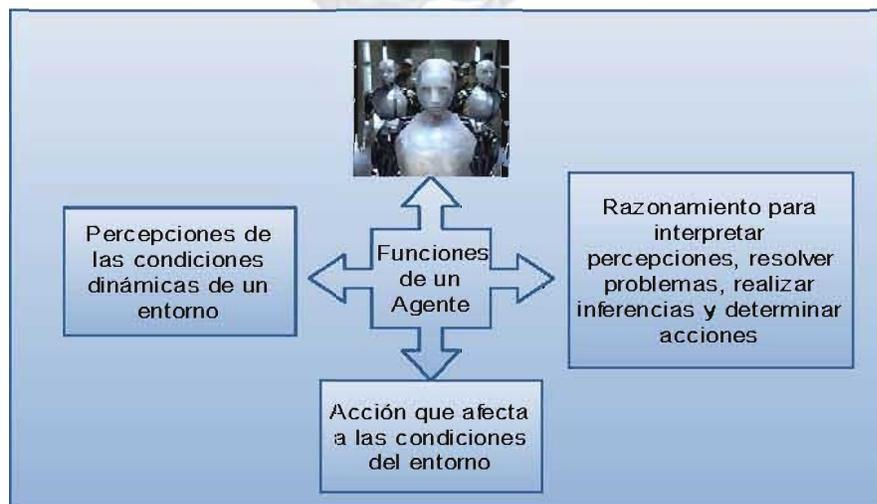


Fig. 2.2 Funciones de un agente

2.6. Agentes inteligentes

Un agente inteligente es una entidad autónoma cuyos sensores y efectores son canales de comunicación con otro agente (humano o computadora) al cual le proporciona un servicio basado en conocimiento.

Un agente inteligente es un sistema anidado y parte integrante de un ambiente (environment) y que detecta o percibe (percepts) datos ambientales, momento a momento, y actúa sobre él con la intención de usar (actions) esos datos para su propia tarea (task) o agenda, afectando así lo que va a detectar en el futuro, sin intervención de terceras partes [Franklin y Greasser, 96]. (Ver figura2. 3)

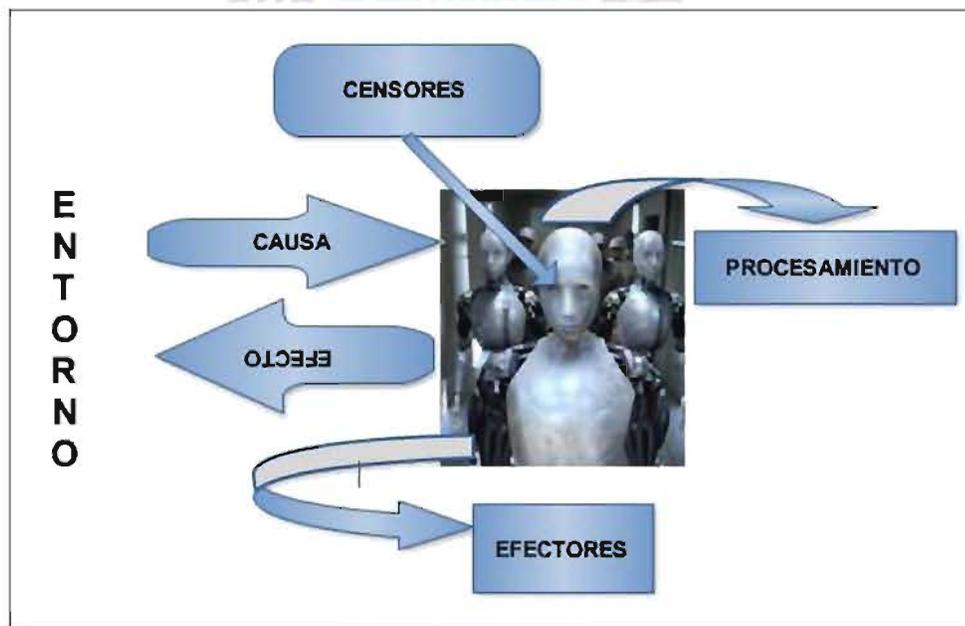


Fig.2.3 Esquema de funcionamiento de un agente

2.6.1. Propiedades y características de los agentes inteligentes

A continuación se describe de manera precisa las propiedades de un agente.

Autonomía: los agentes actúan sin la intervención directa de las personas. Posee la capacidad de razonamiento para generar cursos de acción.

Sociabilidad: los agentes interactúan con otros agentes mediante algún mecanismo de comunicación.

Reactividad: los agentes perciben su ambiente (mundo real, interfaz grafica, conjunto de otros agentes, Internet o combinación de estos) y responden a los cambios de éste.

Proactividad: los agentes no solo actúan en respuesta a su ambiente, sino que son capaces de tener comportamiento orientado a metas.

Además de las propiedades que [Wooldridge, 99] menciona, distingue otras:

Cooperativo: en consecuencia a la sociabilidad, la cooperación entre agentes es una razón de ser para tener múltiples agentes para resolver problemas.

Aprendizaje: los agentes para ser inteligentes requieren la propiedad de poder aprender del ambiente que les rodea.

Veracidad: un agente no comunica deliberadamente información falsa.

Benevolencia: los agentes no tienen metas conflictivas, hará siempre lo que se pida.

Racionalidad: un agente actuara para alcanzar sus metas en la medida que sus creencias, conocimientos y capacidad de razonamiento se lo permitan.

Movilidad: algunos agentes tienen la habilidad de viajar en una red.



Fig.2.4 Propiedades de un agente

Los agentes pueden clasificarse desde diferentes puntos de vista que se mencionan a continuación:

En base a sus capacidades de resolver problemas se clasifican en:

Agentes reactivos: Reaccionan a cambios en su ambiente o a mensajes provenientes de otros agentes. No son capaces de razonar acerca de sus intenciones. Sus acciones se realizan como resultado de reglas que se establecen.

Agentes intencionales: son capaces de razonar acerca de sus intenciones y conocimientos, crear planes de acción y ejecutar dichos planes. Los agentes intencionales pueden ser considerados como sistemas de planeación.

Agentes sociales: poseen la capacidad de los agentes intencionales, mantener los modelos de los otros agentes, razonar sobre el conocimiento incorporado a estos modelos. Tomar decisiones y crear sus planes con respecto a los modelos de otros agentes.

En base a autonomía, aprendizaje y cooperación:

Agentes colaborativos: enfatizan su autonomía y cooperación (con otros agentes) para realizar su tarea. Pueden aprender.

Agentes de interfaz: ponen énfasis en su autonomía y aprendizaje para realizar sus tareas. Los agentes de interfaz asisten y dan soporte al usuario para aprender el uso de una aplicación.

2.6.2. Estructura de los agentes inteligentes

La base de la Inteligencia Artificial es el diseño de un programa de agente,

una función que permita implantar el mapeo del agente para pasar de percepciones a acciones. Este programa se ejecutará en algún tipo de dispositivo de cómputo al que se denominará arquitectura. La arquitectura puede ser una computadora sencilla o un hardware especial y obedece a la siguiente ecuación:

$$AGENTE = ARQUITECTURA + PROGRAMA$$

Donde:

Agente: Es una entidad autónoma que actúa racionalmente de acuerdo a sus percepciones del exterior y el estado de su conocimiento.

Programa: Implanta una función que mapea las percepciones del agente en un conjunto de acciones. Para diseñar y desarrollar un programa esqueleto de un agente es necesario especificar previamente los elementos constituyentes del mismo tales como la memoria en donde se guardara todas las percepciones que capturan los sensores del agente depende del dominio específico en el que se encuentra, así como las acciones con las cuales responderá a través de los efectores.

Arquitectura: La arquitectura utilizada por el programa esqueleto de un agente le permite ejecutarse. Antes de proceder al diseño de un programa de agente es necesario contar con una idea bastante precisa de las posibles percepciones y acciones que intervendrán, qué metas o medidas de desempeño se supone lleve a cabo el agente, así como del tipo de ambiente en que tal agente operará, para tal efecto, denominamos a la matriz PAMA (percepciones, acciones, metas y ambientes), donde:

Percepciones: Es la secuencia de la información de los distintos estados del mundo exterior.

Acciones: Depende de la secuencia de percepciones que percibe el agente y

son una representación de la decisiones que puede asumir para alcanzar sus metas.

Meta: Es la guía por la cual el agente debe discernir sobre lo que quiere lograr y las acciones que quiere desarrollar en función de las percepciones recibidas y las acciones emprendidas.

Ambiente: Es el dominio donde el agente debe desenvolverse a fin de poder ejercer sus acciones sobre él.

2.6.3. Modelos de implementación de Sistemas Tutores Inteligentes (STI)

Las técnicas convencionales de programación para desarrollar (STI), se basan en el planteo de modelos básicos, que interactúan entre sí y donde cada uno de ellos posee la información necesaria para poder cumplir con su tarea. Esta definición de módulos dentro de un (STI), es útil a la hora de identificar cada una de las partes de este, pero es casi imposible, hasta el momento, que en las implementaciones de los (STI) estos módulos estén definidos completamente como lo indica, categoría que no soporta, ya que siempre existe un solapamiento con el dominio del problema a resolver con todos los demás módulos.

Una de las ventajas principales, según Iglesias, Garijo y González [Iglesias et al., 01] es la popularidad de las metodologías orientadas a objetos, para las que existen muchas metodologías para implementar este paradigma, como es el caso de OMT (Object Modeling Technique), OOSE (Object Oriented Desing) y UML (Unified Modeling Language), entre otras las experiencias adquiridas con la utilización de estas metodologías desde hace por lo menos una década puede dificultar la adopción de las metodologías de agentes por la inercia que se genera en la industria, hasta tanto se disponga de una propia.

2.6.4. Trabajos similares

Comenzando con las primeras implementaciones de los sistemas que serían el modelo de instrucción asistido por computadora (CAI), en la década de los setenta, aparecen implementaciones como el sistema Scholar lleva a la práctica una nueva propuesta de Carbonell [Carbonell, 70]. Esta es la primera implementación práctica donde el sistema recibe un feedback de los alumnos e intenta inferir el estado cognitivo de los mismos.

Luego aparecieron los tutores Why y Sophie de Collins y Brown [Stevens et al., 77]; también muy limitados en su funcionalidad, pero en los diferenciaron los intelligent (CAI), y los (STI), que hasta la época se consideraban sinónimos. El proyecto Why de Collins siguió los lineamientos propuestos por Carbonell, pero, explicando de modo socrático y por exposición indirecta. Todas las aproximaciones a resolver el problema en la década de los setenta eran muy rudimentarias y se encuentran en desuso.

2.6.4.1. El proyecto CircSim

Actualmente hay que destacar el proyecto CircSim, Sistema Tutor Inteligente (STI), creado en conjunto por el departamento de ciencias de la computación de Illinois Institute of Technology y el departamento de fisiología del Rush College of Medicine. Extractor es el más avanzado actualmente y su implementación se utilizan para complementar las clases teóricas sobre problemas cardiovasculares [Kim, 89, 00; Cho, 00; Hume et al., 92].

Las bases de esta construcción, basado en el lenguaje natural para los estudiantes de medicina del primer año que están aprendiendo el tema de control de reflejos de la presión sanguínea. En el CirSim, los estudiantes resuelven pequeños problemas mientras son tutelados en forma socrática una computadora [Woo Woo, 91].

Su funcionamiento básico es el siguiente: se presentó un problema a un alumno,

que no resuelve (si puede) y la respuesta de éste se compara con la del solucionador de problemas. Para continuar en el siguiente paso, el planificador de la lección toma las predicciones del alumno y el modelo del alumno, para generar una conversación y detectar los errores posibles del alumno. Este ciclo continúa hasta obtener el objetivo final del problema planteado. El ciclo está implementado sobre una máquina de estados finitos y los planes están almacenados en una pila.

2.6.4.2. STI orientados a la enseñanza de lenguajes de programación

En el ámbito de la programación se han desarrollado varios tutores y asesores inteligentes. Para apoyar la enseñanza de los lenguajes de programación [Castillo et al., 98]. Se incrementaron y se observa utilizan distintas arquitecturas base, que se detallan a continuación con sus características más sobresalientes.

El primer STI orientado a la enseñanza de programación es Meno [Wolf, 84], están específicamente orientado a la enseñanza de programación en el lenguaje Pascal. Este sistema utiliza una serie de plantillas, que representan las soluciones posibles al problema planteado. Además contiene una bases de datos con los errores más comunes que realizan los estudiantes, para poder reconocer los a través de un análisis del código fuente. De acuerdo a los errores encontrados, se representan al alumno una serie de sugerencias para corregir el programa, indicando de además, cuáles son las causas de su error y qué conceptos aplicó en forma incorrecta o conceptos que desconoce. Por otra parte el tutor Prust de Carbonell puede analizar completamente el código fuente sin necesidad de utilizar plantillas. Que además posee innovación que es el diagnóstico de intenciones [Johnson, 86]. Esto significa esto significa que el asesor es capaz de deducir las metas del usuario y compararlas con el código que analiza, lo que permite detectar errores no triviales en la lógica de programación. Esta característica se extendió luego a otros dominios, fuera del asesor por su gran utilidad.

Otro ejemplo de asesor inteligente, orientado a la enseñanza de lenguajes de programación es el Coach en este caso, se ofrece un ambiente de ayuda interactiva, donde el usuario proporciona una entrada (si se espera sea un programa en Lisp) y en un papel separado, el asesor ofrece información relevante. Posiblemente la innovación más terminante que presenta Selker, es que el modelo del estudiante es adaptativo, incluyendo variables tales como las tasas de cambio en el alumno, de esta forma, el sistema va evolucionando cuando el estudiante aprende y esta manera el mismo recibe un distinto tipo de asesoría según su nivel de conocimiento. Por otro lado, en este estudio se demostró claramente la capacidad del sistema para ser adaptado a otro dominio de conocimiento (en este caso, la enseñanza de los comandos del sistema operativo Unix).

2.7. Metodología

La metodología utilizada en la elaboración de este trabajo de investigación se basa en el método científico, para el diseño y elaboración del prototipo se sigue el método de ingeniería de software orientado a objetos con UML, la construcción del agente se basa en la propuesta del método de desarrollo y construcción de agentes inteligentes basados en la ingeniería de software tradicional aplicado al paradigma de agentes.

2.7.1. Método científico

En la investigación se hace uso del método científico, el cual consta de dos partes que son los métodos teóricos que nos permiten comprender y apreciar los objetos a través del razonamiento y la abstracción para llegar a formalizar el modelo, y los métodos empíricos que consta de un conjunto de procedimientos:

- **La observación** de la realidad para llegar a detectar la raíz del problema.

- **Formulación** de la hipótesis
- **La experimentación** sustentada en un modelo formal mediante el cual se representa el tratamiento adecuado.
- **Verificación** de la hipótesis, tras haber obtenido una serie de resultados alcanzados por el alumno y que son medibles.

2.7.2. Método de ingeniería de software orientado a objetos

La ingeniería de software orientado a objetos sigue los mismos pasos que el enfoque convencional, análisis, diseño, implementación y pruebas.

- Análisis, consiste en la investigación del problema, su propósito es definir todas las clases que son relevantes al problema a resolver, las operaciones y atributos asociados, las relaciones y comportamientos asociados a ellas, dentro del dominio del problema.
- Diseño, transforma el modelo de análisis definiendo los objetos lógicos del software: abstracción, ocultamiento de información, independencia funcional y modularidad que finalmente serán implementados en un lenguaje de programación.
- Implementación y pruebas, se implementan los componentes de diseño, para luego realizar las pruebas en niveles diferentes.

2.8. Sistemas basados en el conocimiento

Los sistemas basados en el conocimiento tratan con problemas complejos en un dominio específico y requieren de un elevado conocimiento del mismo. Se lo define mediante la ecuación:

Sistema basado en el conocimiento = Conocimiento + Razonamiento

Base de conocimientos: Es el elemento que almacena el conocimiento de un dominio que debe estar convenientemente formalizado y estructurado y lo

pone a disposición del motor de inferencia para su posterior tratamiento. Algunos autores manejan el concepto único de base de conocimientos y otros lo dividen en base de datos y base de relaciones.

- Base de datos: contiene los hechos, los datos y las heurísticas puntuales del dominio.
- Base de relaciones: contiene las relaciones que se establecen entre los hechos y las heurísticas del dominio.

Motor de inferencia: También llamado intérprete de reglas, es la unidad lógica con la que se extraen conclusiones de la base de conocimientos y coordinan las acciones que deben realizar todos los componentes del sistema.

Tiene la capacidad de:

- Determinar las acciones que tendrán lugar, el orden en que lo harán y como lo harán entre las diferentes partes del dominio.
- Determina cómo y cuando se procesaran las reglas, y dado el caso también la elección de que reglas deberán procesarse.
- Control del dialogo con el usuario.
- La decisión sobre los mecanismos de procesamiento de reglas, es decir que estrategias de búsqueda se implementaran, es de vital importancia para la efectividad del sistema en su conjunto.

[CAPITULO 3]

ARQUITECTURA DEL SISTEMA TUTOR

3.1. INTRODUCCION

El capítulo anterior nos sirvió para exponer todo el marco teórico que ahora se convierten en las bases para la construcción del prototipo, y así poder llegar a satisfacer los objetivos que nos habíamos propuesto en el primer capítulo.

El objetivo del presente capítulo radica en que se formalizara, después de una serie de procesos, la estructura principal del prototipo del sistema tutor para enseñanza del lenguaje SQL.

El campo de estudio está limitado al apoyo en el proceso enseñanza-aprendizaje, especialmente diseñado para estudiantes de la materia de taller base de datos de la carrera de Informática.

3.2. Arquitectura del sistema tutor

Un sistema tutor enfoca una sesión de enseñanza como un proceso de cooperación entre el tutor y el estudiante, con los objetivos de enseñar y aprender determinados conceptos, un factor importante para decidir y aplicar una estrategia es conocer la materia que se imparte y comprender la forma en que el estudiante la asimila, es decir, construir un modelo de la conceptualización que el estudiante va adquiriendo.

La arquitectura de sistema tutor presenta componentes o módulos cuya interrelación permitirá que el proceso de enseñanza-aprendizaje se lleve a cabo

de la manera más eficaz y eficiente y alcanzar sus objetivos. (Ver Fig. 3.1).

Arquitectura y Componentes

Tienen como principal objetivo impartir la enseñanza de un contenido dado en dominio en la forma más adecuada a las necesidades individuales del estudiante. Estos sistemas se basan en una arquitectura compuesta por tres grandes módulos: el modulo del estudiante, el modulo del tutor, el modulo dominio, [Villareal, et al., 01]. Podría agregarse un cuarto módulo denominado como el modulo de evaluación y, un quinto modulo como es el modulo de interface y el ecosistema propuesto por Cataldi [Cataldi, 04].

- 1. Módulo Estudiante:** se representa el estado inicial del alumno y sus características individuales, las más importantes es el conocimiento individual instantáneo sobre el dominio [Villareal et al., 01], por otro lado Guardia Robles [Guardia Robles, 93] lo define como: “el modelo del estudiante, que refleja cuanto conoce el estudiante sobre el dominio, así como las experiencias cognitivas y de aprendizaje que ha llevado, del cual puede obtenerse un diagnostico.”
- 2. Módulo del dominio:** Posee el conocimiento de la materia formado por reglas de producción, estereotipos, etc. De aquí el modulo tutor obtiene el conocimiento que debe enseñar [Villareal et al., 01]. Definido como “El modelo experto o del dominio, el cual versa sobre la materia o curso que se impartirá” [Guardia Robles, 93].
- 3. Módulo Tutor:** Posee el conocimiento sobre las estrategias y tácticas de enseñanza para poder seleccionarlos en función de las características del estudiante que están almacenadas en el modulo del estudiante [Villareal et al., 01] pero debe ir más allá de la experiencia en el dominio ya que debe ofrecer a cada estudiante un método de enseñanza de acuerdo a sus necesidades y criterios.

4. **Módulo de Evaluación:** Se encarga de realizar una evaluación en general del sistema y generar estadísticas acerca de los estudiantes; pudiendo efectuar el diagnostico evolutivo luego de cada uno de los estados considerados, de este método podría también predecir el comportamiento en los eventos futuros. La evaluación de los estudiantes debe ser constante y durante la carga del proceso, con instancias de autoevaluación. También se deberán generar informes a utilizar para evaluar al sistema como método apto de enseñanza.
5. **Módulo de Interfaz:** Es la interface de la interacción entre el STI y el estudiante, que se encarga de mostrar el material del dominio, “la interfaz, que permite a los usuarios interactuar con el sistema. Se distinguen tres tipos de usuarios: el estudiante, el instructor y el desarrollador del sistema” [Guardia Robles, 93].

Los primeros tres módulos conforman la arquitectura clásica de Carbonell [Carbonell, 70] y también funcional de los STI [Villareal et al., 01]. Fig. 3.1.

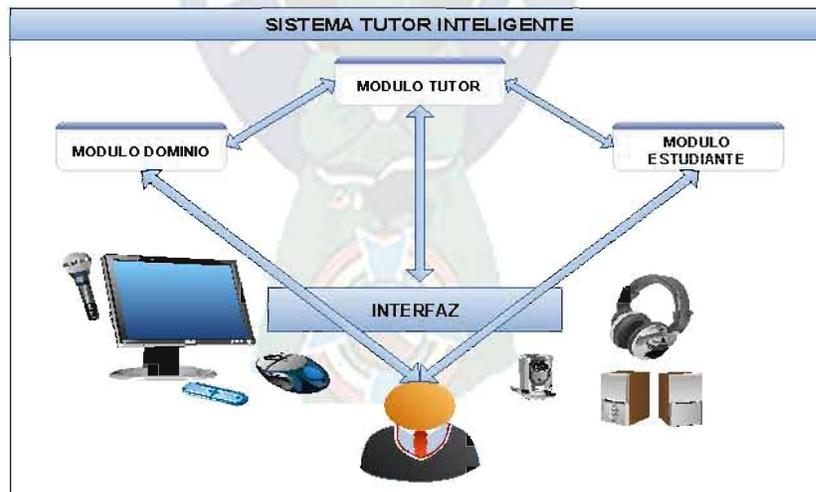


Fig. 3.1: Arquitectura básica de un STI.

3.2.1. El Módulo Estudiante del STI

El módulo del Estudiante tiene por objetivo brindarle al sistema el estado cognitivo instantáneo del estudiante. [Costa et al. 05] plantean los siguientes submódulos básicos.

- a) **Estilos de aprendizaje:** Está compuesto por una base de datos con los estilos de aprendizajes disponibles en el sistema, los métodos de selección de estilos y las características de cada uno de ellos, Un estilo de aprendizaje es la forma de clasificar el comportamiento de un estudiante de acuerdo a la manera en que toma la información, forma las estrategias para aprender, cómo entiende y cómo le gusta analizar la información que está utilizando para acceder a un conocimiento determinado. En otras palabras, es una forma agrupar o clasificar un estudiante de acuerdo a un perfil en relación con la información.
- b) **Estado de conocimientos:** Contiene el mapa de conocimientos obtenido inicialmente a partir del módulo del dominio y que el actualizador de conocimientos irá modificando a través de los resultados obtenidos en las evaluaciones efectuadas por el módulo del tutor quien le enviará dichos resultados procesados. Estos datos también se le proporcionarán al módulo tutor para que éste pueda decidir qué conocimientos presentar y cómo hacerlo, en cada una de las sesiones con el estudiante, basado en los postulados de Coll [Coll, 94] acerca de qué enseñar y cómo hacerlo. La idea es que este módulo contenga la representación del estado de conocimientos instantáneo del alumno, que podría ser modificado tanto en las sesiones posteriores de tutelado con el Sistema Tutor Inteligente.

3.2.2. El Módulo del Dominio

Tiene el objetivo global de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del Sistema Tutor Inteligente (STI).

- a) **Parámetros básicos del sistema:** Los parámetros básicos del funcionamiento del sistema se almacenan en una base de datos, así como también las contraseñas para los profesores, para que éstos puedan acceder a menús particulares, donde se le muestre el desarrollo de las sesiones pedagógicas que posee el sistema, las estadísticas, las proyecciones, etc.

Existen otros parámetros internos, tales como el lenguaje por defecto con el que interactuará el estudiante, las contraseñas de los usuarios y otros datos que se requieran para operar el sistema, pero que no se encuentran almacenados en ninguno de los otros submódulos por lo que entran dentro de esta categoría y por lo tanto se almacenan como un parámetro más para el funcionamiento del sistema.

- b) **Conocimientos:** Son los contenidos que deben cargarse en el sistema, a través de los conceptos, las preguntas, los ejercicios, los problemas y las relaciones: Se piensa que éstos deberían estar disponibles en varios idiomas y con diferentes formas de acceso al conocimiento (directa a cada una de las definiciones, descendiendo por el mapa de conceptos, etc.) para que independientemente del protocolo pedagógico que el planificador de la lección seleccione para desarrollar el concepto seleccionado, se encuentren todos los datos a disposición del generador de contenidos.

En el mapa de conceptos de los conocimientos, se encuentran las relaciones entre los conceptos, sus dependencias, así como los conceptos a explicar para cada tema en cada asignatura, a fin de poder armar un currículo completo. Cuando se coteja este mapa de conocimientos con la estructura de los conocimientos del estudiante (proveniente del módulo del estudiante) se detectan cuáles son los conceptos faltantes, por lo que se puede presentar una sesión pedagógica (o varias) a través del planificador (del módulo del tutor) a fin de brindárselos a fin de actualizar el estado del estudiante.

- c) **Elementos Didácticos:** Son las imágenes, videos, sonidos, es decir material multimedia que se requiere para facilitarle al alumno apropiarse de conocimiento en la sesión pedagógica. También se puede incluir una biblioteca con ejercicios y ejemplos, con las respuestas para su autocomprobación, así como cualquier otro recurso didáctico que permita al alumno acceder a ese conocimiento, es decir desde la idea de nuevas puertas de acceso al mismo.

3.2.3. El Módulo del Tutor del STI

Es quien define y aplica una estrategia pedagógica de enseñanza, contiene los objetivos a ser alcanzados y los planes utilizados para alcanzarlos. Selecciona los problemas, monitorea el desempeño, provee asistencia y seleccionar el material de aprendizaje para el estudiante integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y del dominio a ser enseñado.

- **Protocolos pedagógicos:** Está compuesto por una base de datos de estilos pedagógicos o protocolos pedagógicos disponibles en el sistema, las formas de selección y características de cada uno de los métodos.

Los métodos disponibles deben contener la información necesaria para poder desarrollar una sesión pedagógica con los recursos didácticos necesarios para que el alumno pueda aprovechar cada uno de los estilos pedagógicos. Por ejemplo, una sesión socrática deberá tener almacenados los algoritmos para la resolución de los problemas en el módulo del dominio con sus respectivas preguntas y repreguntas, así el estilo magistral deberá contener las herramientas necesarias para mostrar las diapositivas que se utilicen para mejorar y facilitarle al alumno el acceso a los contenidos.

Los factores que influyen en la selección del protocolo pedagógico estarán dados, entre otros, por el perfil de aprendizaje del alumno (que se obtiene de una manera procesada por parte de módulo del Estudiante) y los conceptos faltantes del alumno en un determinado tema.

- **Planificador de lección:** Dependiendo del protocolo pedagógico seleccionado y del estado de conocimientos del alumno (a partir del módulo del estudiante), se deberán poder establecer los objetivos de la lección (ya sea un tema, un ejercicio, un conjunto de definiciones, etc.). A medida que la lección se lleva a cabo, el generador de contenidos diagramará la lección para alcanzar los objetivos de la misma, los cuales se almacenarán en una pila de objetivos.

Si en la lección aparecen objetivos secundarios, o el sistema determina que existen objetivos previos que deben cumplirse para lograr el objetivo general de la sesión (un ejemplo de esto sería el caso de que el estado de conocimientos entregado por el módulo del estudiante sea impreciso o no refleje el estado cognitivo real del usuario) éstos se agregan a la pila y se intentan satisfacer antes de lograr el objetivo principal de la lección.

A partir de las estructuras de representación interna del conocimiento las salidas se transforman en lenguaje natural por medio de un módulo dedicado a tal fin, facilitando el entendimiento del usuario. El submódulo de lenguaje natural se encarga tanto de generar el lenguaje natural del usuario y obtener los conceptos relevantes que le son preguntados o evaluados.

El submódulo de lenguaje natural se encarga tanto de generar el lenguaje natural para las sesiones, como de procesar el lenguaje natural del usuario y obtener los conceptos relevantes que le son preguntados o evaluados.

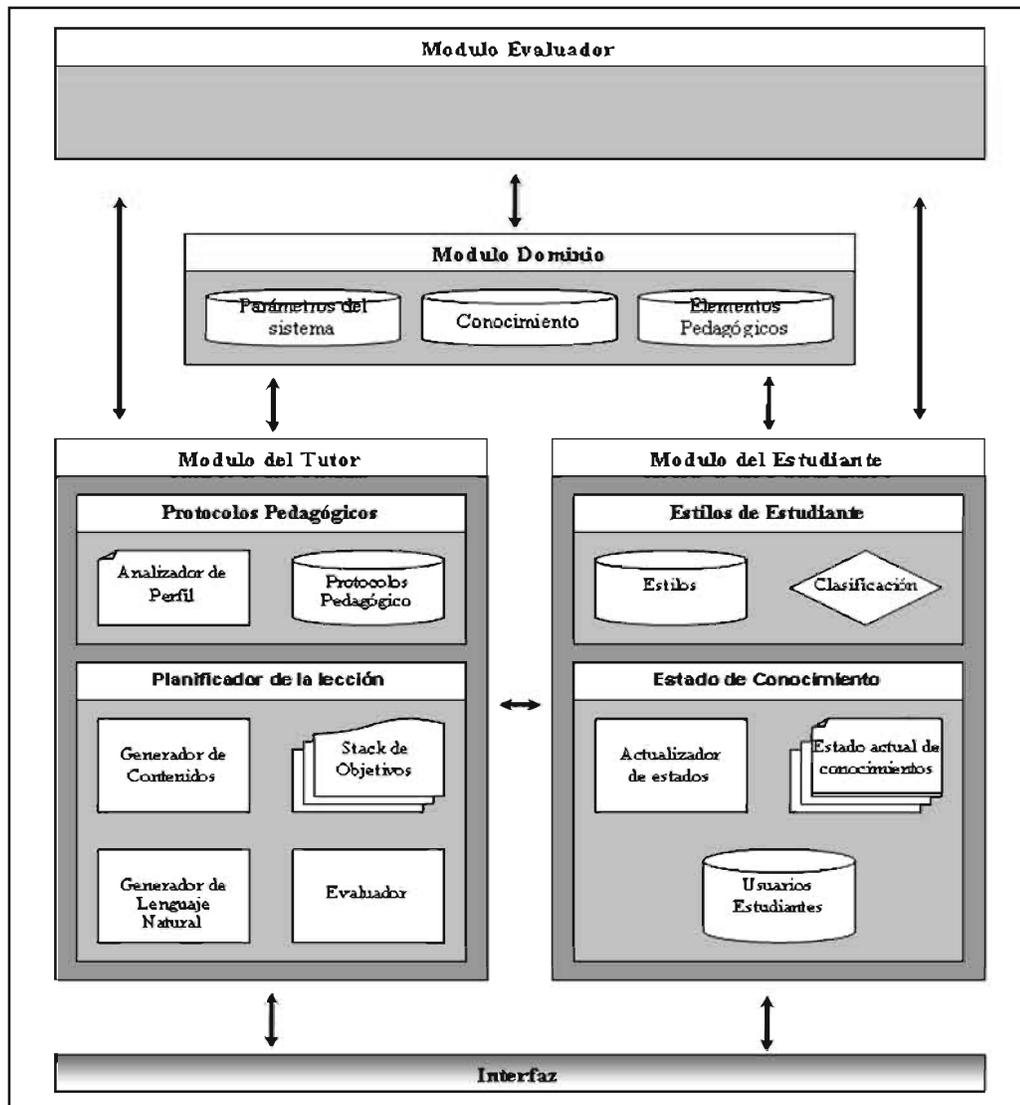


Fig. 3.2: Módulos y submódulos del STI con módulo evaluador adicional. [Fuente: IV Workshop de tecnología informática en educación]

3.3. Análisis de los Submódulos del Módulo Tutor

A continuación se describe cada uno de los submódulos que se proponen para el módulo del tutor. Se definen las interfaces y las tareas específicas de cada uno de ellos, se señalan las mejoras respecto de los desarrollos actuales en los Sistemas Tutores Inteligentes y se los ubica dentro de la estructura global del módulo del tutor.

3.3.1. Submódulo Protocolos Pedagógicos

En la Figura 3.3 se puede ver el esquema general del submódulo de los protocolos pedagógicos, con sus dos componentes básicos. Éste interactúa, con el resto de los submódulo del módulo del tutor y es el que realiza las peticiones de los datos, al módulo del estudiante, para averiguar cuál es el perfil de éste y cuál es su estado de conocimientos.

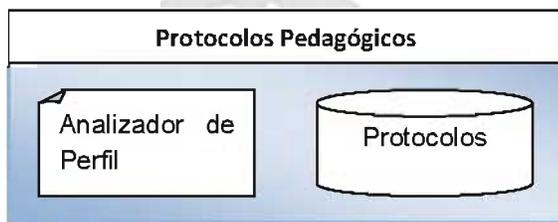


Fig. 3.3 Submódulo de Protocolos Pedagógicos [Fuente: Elaboración propia]

El submódulo de protocolos pedagógicos consta de un analizador de perfil que determina a partir del perfil del estudiante almacenado en el módulo del estudiante, cuál será el mejor protocolo pedagógico para utilizar en la siguiente sesión pedagógica.

El sistema cuenta con una base de protocolos pedagógicos cuyo uso estará subordinado a la disponibilidad de los contenidos en el módulo de conocimientos, pero la lección siempre se puede generar para alguno de los protocolos.

Esto se puede aclarar a través de un ejemplo: cuando el sistema determina que requiere dar una clase del tipo magistral de un tema en particular, pero el módulo de conocimientos no contiene la cantidad suficiente de recursos didácticos como para que la sesión se lleve a cabo. El sistema puede decidir que el material es insuficiente y cambiar a otro protocolo pedagógico, o sino, en el peor de los casos, mostrar solamente las definiciones de los conceptos y las relaciones básicas entre ellos, a través, por ejemplo, de una mapa conceptual como lo define Novak (1984) que es la información mínima que puede poseer el módulo de conocimiento sobre un tema en particular.

A continuación se analizará como el módulo del tutor puede realizar la selección del protocolo pedagógico más adecuado que se ajuste al estilo del estudiante y al objetivo de la sesión pedagógica en particular.

3.3.2. El Analizador de Perfil

El Analizador de Perfil debe encontrar el mejor protocolo pedagógico disponible en el sistema para que el estudiante pueda obtener la sesión pedagógica deseada. Es este caso la eficacia resultante estará influida en gran medida por la precisión en la selección del protocolo seleccionado.

Se utilizarán las planillas de estilos de aprendizaje recreadas de Felder [Felder, 90], que son herramientas para la toma de datos de alta validez y confiabilidad que han demostrado su aptitud a la hora de clasificación de los estilos por su autor y otros investigadores que las utilizaron. La idea de estas planillas es obtener el estilo a través del cual el alumno aprende de un modo más efectivo. Entre las opciones disponibles, se seleccionó esta planilla de estilos de aprendizaje ya que la misma se había aplicado en investigaciones acerca de los estilos de aprendizaje de los estudiantes de Ingeniería Química respondiendo de un modo óptimo. Según Felder [Felder, 90], los estilos se trabajan de a pares y se los puede definir de acuerdo a la:

- **Forma de procesar la información:** Tareas Activas – Tareas de Reflexión.
- **Forma de percibir la información:** Sensorial – Intuitiva (racional).
- **Forma de presentar la información:** Visual – Verbal.
- **Forma del proceso del aprendizaje:** Secuencial – Global.

3.4. Modulo Evaluación

Este modulo contiene el proceso de evaluación que se le aplica a cada estudiante una vez que ha terminado su sesión de aprendizaje de cada tema. Tiene el objetivo, no solo de evaluar el progreso de los estudiantes sino también para orientar y guiar el proceso de aprendizaje de los mismos. Tal

evaluación hace uso del enfoque psicométrico, donde las notas o calificaciones representan una valoración relativa, esta evaluación implica medir el rendimiento del estudiante en su nivel de aprendizaje y será llevada a una escala de calificaciones como se puede observar en la siguiente tabla:

Rendimiento	Resultado de medición ordinal
Sobresaliente	90-100
Distinguido	77-89
Bueno	64-76
Suficiente	51-63
Reprobado	0-50

Tabla 3.1 Escala de evaluación

[Fuente: Escala vigente de calificaciones desde julio de 1972]

Para realizar un aprendizaje, es necesario tener registrada toda la información sobre las evaluaciones realizadas a los alumnos, así como un método de discretización del nivel de conocimiento. Para ello, el responsable docente define sobre cada tema o elemento de la unidad didáctica, la serie de técnicas de evaluación que deberán tomarse como referencia (como cuestionarios, prácticas, ejercicios, etc.) [www.jakarta.apache.org/velocity]. Cada técnica de evaluación tiene asociado un factor reductor (fv) encargado de definir el grado de importancia (comprendido entre 0 y 1). La tasa de evaluación (ecuación [1.1]) aglutina todos los resultados del alumno.

$$EV = \{ev_1, ev_2, ev_3, \dots, ev_n\}, TV = \left(\frac{\sum_{i=1}^n (fv_i * ev_i)}{n} \right) \text{ ec 1.1}$$

Para realizar el cálculo de los parámetros de la variable del nivel de conocimiento (NC) de una unidad didáctica (x), discretizamos las tasas de evaluación de los alumnos con umbrales definidos en cada unidad didáctica. Si utilizamos varios atributos, la discretización es más compleja Wang [Wang, 98]. La variable multivaluada NC_x adquiere los valores $\{x_1=\text{bajo}, x_2=\text{medio}, x_3=\text{alto}\}$, en un vector de parámetros $\theta = \langle \theta_1, \theta_2, \theta_3 \rangle$, La probabilidad de un parámetro θ_k sigue la distribución de Dirichlet [Dirichlet, 97], ecuación [1.2].

$$p\left(\frac{\theta_k}{D}\right) = Dir\left(\frac{\theta_k}{\alpha_i}, \alpha_2, \alpha_3, \dots, \alpha_r\right) \quad \alpha = \sum_{k=1}^r \alpha_k \text{ y } \alpha_k > 0 \quad \text{ec 1.2}$$

Si los datos son completos y existe separabilidad direccional entre los nodos de la red, puede actualizarse de forma independiente cada vector de parámetros. Asumiendo que cada vector sigue la distribución de [Dirichlet, 97] puede obtenerse la distribución de probabilidad condicionada del conjunto de padres, Pa_i , ecuación [1.3] donde N_k es el número de apariciones de cada parámetro en la muestra D en la que $NC_k=x_k$ y $Pa_i=pa_{ij}$.

$$p\left(\frac{\theta_{ij}}{D}\right) = Dir\left(\frac{\theta_{ij}}{\alpha_i} + N_1, \alpha_2 + N_2, \alpha_3 + N_3\right) \quad \text{ec 1.3}$$

Si se produce un nuevo evento en el sistema, como por ejemplo la inserción de nuevos registros de evaluación, (caso $x, N+1$), se actualizan las probabilidades con la ecuación [1.4], simplificada en [1.5]. Para mayor detalle, ver www.webmarcro.org

$$p\left(NC_{x,n+1} = \frac{X_k}{D}\right) = \prod_{i=1}^n \int \theta_{ijk} * Dir\left(\frac{\theta_{ij}}{\alpha_i} + N_1, \alpha_2 + N_2, \alpha_3 + N_3\right) * d\theta_{ij} \quad \text{ec1.4}$$

La tarea de planificación asigna menor tiempo de estudio a aquellas tareas que no supongan mucho esfuerzo al estudiante, sin embargo, refuerza el estudio de aquellas tareas que necesiten más dedicación. Para implementar esta idea, es necesario que cada unidad didáctica tenga definidos los objetivos deseados al finalizar la actividad de aprendizaje. Cada objetivo especifica el conocimiento

adquirido en los distintos elementos de la unidad didáctica. La medida utilizada por el planificador es la probabilidad de la consecución de objetivos a priori (*PCO*). Se define de la siguiente forma:

Variable objetivo: $O_i = \{C, NC\}$ *C* = Conseguido, *NC* = No conseguido

Factor reductor de un objetivo: $f \in [0,1]$

Probabilidad de la consecución de objetivos a priori (*PCO*) en una unidad de conocimiento.

$$PCO = \sum_{i=1}^{i=n} f_i * P(O_i = C) / n \quad \text{ec 1.5}$$

Donde n=numero de objetivos definidos en una unidad didáctica.

3.4.1. Interfaz

Gestiona la comunicación interactiva entre el sistema, el agente y los usuarios, ya que permite a los mismos interactuar con el sistema. Se distinguen tres tipos específicos de usuarios: el Estudiante, el Docente, y el Administrador del sistema.

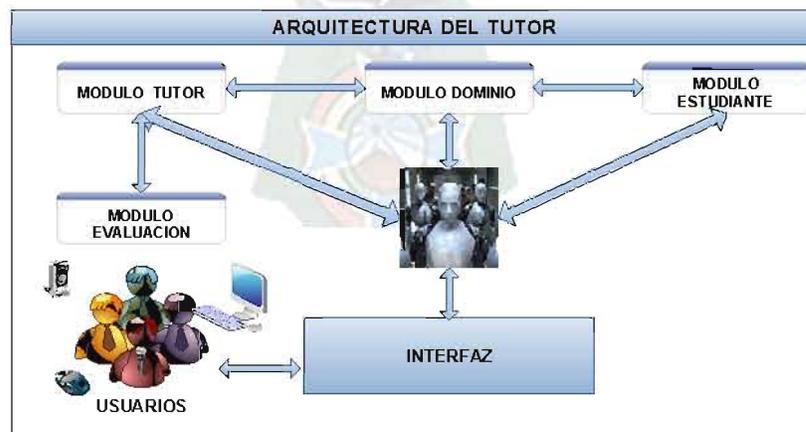


Fig. 3.4 Arquitectura del tutor [Fuente: Elaboración propia]

3.4.2. Interrelación entre componentes

La interactividad de los componentes descritos puede observarse en la siguiente figura 3.5

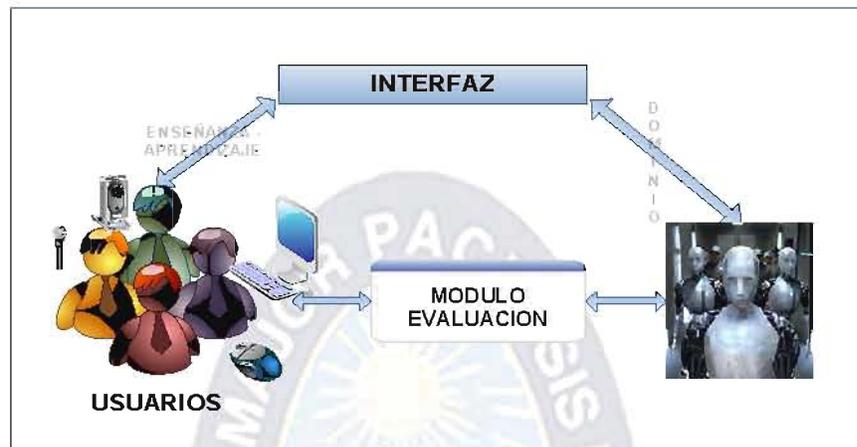


Fig. 3.5 Interrelación entre componentes del tutor

[Fuente: Elaboración Propia]

3.5. Desarrollo orientado a objetos con UML

Si se quiere que el sistema sea eficaz, eficiente y mantenible es necesario que el problema sea analizado y la solución sea cuidadosamente diseñada, tarea que toca hacer en este capítulo. Para tal cometido se sigue el Método de Desarrollo Orientado a Objetos de Craig Larman [Larman, 99] quien propone un proceso que incluye una serie de actividades aplicando los últimos avances de Ingeniería de Software.

Este proceso incluye tres fases al más alto nivel, los cuales son:

- **Planificación y Especificación de requisitos:** Planificación, definición de requisitos, construcción de prototipo, etc.
- **Construcción:** Que tiene las siguientes fases: Análisis, Diseño, Implementación y Prueba.
- **Instalación:** Comprende la puesta en marcha del sistema en el entorno previsto de uso.

Para el desarrollo y construcción del prototipo seguiremos los pasos descritos anteriormente.

3.6. Fase de planificación y especificación de requisitos

En esta fase se determinan los requisitos y se definen los casos de uso de alto nivel.

Descripción de requisitos

El objetivo de esta actividad consiste en identificar que es lo que realmente se necesita. Lo cual se logra mediante la comunicación entre el usuario (cliente) y el desarrollador. Dicha actividad nos lleva a determinar: El propósito, el ámbito, las funciones y los atributos del sistema.

- ❖ **Propósito:** Apoyar a los estudiantes, mejorando su aprendizaje en la materia de base de datos.
- ❖ **Ámbito del sistema:** El ámbito del sistema es el área de base de datos. Se distinguen a los siguientes usuarios:
 - **Los estudiantes:** son los usuarios directos quienes interactúan con el sistema, aprenden para después dar una evaluación.
 - **Los Docentes:** son los que supervisan y guían al estudiante durante el proceso enseñanza-aprendizaje, al final pueden obtener del sistema tutor un historial del seguimiento de cada estudiante que ingreso al sistema. Pueden tomar también el rol del estudiante.
 - **El Administrador del sistema:** tiene la tarea de administrar la seguridad e integridad del sistema y las bases de datos. Es el único actor que puede hacer modificaciones al sistema de acuerdo a necesidades y requerimientos de los demás usuarios.

3.6.1. Requerimientos de usuario

Estudiantes

- Requieren de un sistema tutor sencillo y fácil de manejar, muy amigable.
- Donde el material que se les enseñe este de acuerdo a su nivel de conocimiento.
- Que el sistema cuente con una interfaz interactiva y amigable. (Texto, imagen y sonido).
- Que el entorno con el que tenga que interactuar sea lo más explicativo posible para saber que decisiones debe tomar.

Docente

- El contenido de los temas debe estar supervisado y revisado por el Docente del área.
- La adición o modificación de los contenidos de la materia deben ser sugeridos y/o autorizados por el docente del área.
- La generación de calificaciones de evaluaciones no debe ser ambigua, por lo tanto debe ser automática.
- Tener a su disposición, un informe impreso del historial y seguimiento de los estudiantes (que ingresaron al sistema), para conocer su grado de aprovechamiento.

Administrador del sistema

- El sistema debe ser fácil de instalar, adaptable al entorno y al medio.
- La base de datos debe ser creada bajo estándares de normalización para garantizar la integridad y el almacenamiento correcto de los datos.

Requerimientos del Agente

- Contar con un dominio de conocimientos para que pueda impartir su enseñanza al estudiante o tomar ciertas decisiones.

- Contar con una base de datos del estudiante para que la enseñanza sea más personalizada.

3.6.2. Requerimientos de proceso de datos

- El estudiante puede repetir con la sesión de aprendizaje de cierto tema cuantas veces lo desee y al terminar dicho tema dará una evaluación que tiene una opción a recuperar, si es que no logro los mejores resultados en el primer intento.
- Una vez realizada la evaluación el sistema tendrá un registro y mostrara los resultados obtenidos por el estudiante.

3.6.3. Requerimientos del sistema

Para la elaboración del sistema tutor se hacen necesarios los siguientes requerimientos:

3.6.3.1. Requerimientos de software

- ✓ Lenguaje de programación PHP 5.0 Lenguaje que nos permite implementar el prototipo.
- ✓ Flash 8 Paquete que nos permite darle una nueva imagen a la presentación de las pantallas y gráficos personalizados del tutor.
- ✓ Manejador de Base de datos MySql. Y Apache como complemento de Xamp
- ✓ Procesador de texto Microsoft Word para la redacción de las notas planillas, ect.
- ✓ Edraw Max es una herramienta de diseño de las figuras, para las pantallas graficas

3.6.3.2. Requerimientos de Hardware

Para la elaboración del sistema tutor se requiere de una computadora con las siguientes características:

- ✓ Pentium III o superior
- ✓ Memória RAM de 512 MB o superior
- ✓ Tarjeta de video 64 MB SVGA
- ✓ Monitor de 15" o 17" color
- ✓ Disco Duro de 20 GB o mayor
- ✓ Tarjeta de sonido
- ✓ Teclado
- ✓ Impresora
- ✓ Mouse

3.6.4. Requisitos de interfaz de usuario

- **Interfaz Principal:** Esta interfaz será atractiva y colorida donde se mostraran las opciones principales: "Ingresar"; que permitirá ingresar al sistema tutor.
- **Interfaz de Temas:** Una vez que el estudiante haya ingresado al sistema se le presentara esta interfaz atractiva, donde se encontraran los temas que el estudiante puede seleccionar para su aprendizaje, haciendo clic que lo llevara a la interfaz siguiente.
- **Interfaz de Evaluación:** Esta interfaz le permitirá al estudiante que siguió todo el proceso de aprendizaje de cierto tema, dar un examen elegido el botón Questions "Quiz", El objetivo de esta interfaz es disponer de preguntas abiertas, cerradas, de selección múltiple, que el estudiante debe responder seleccionando correctamente. Al final de este proceso, el sistema se encargara de hacer la evaluación automática para después emitir los resultados, el estudiante tendrá dos oportunidades para evaluarse que será habilitado y supervisado por el docente.

3.6.5. Funciones del sistema

Las Funciones principales que el tutor debe realizar son las siguientes:

NRO.	FUNCIONES
1.	Inscripción: El tutor registrara todos los datos personales de los estudiantes que accedan por primera vez al mismo, para hacerle un Seguimiento personalizado dándole un nombre y un password.
2.	El sistema tutor abarcara todos los temas relacionados al lenguaje SQL de acuerdo a la curricular de la materia.
3.	El sistema tutor utilizara ejemplos de ejecuciones de sentencias SQL durante el proceso de enseñanza-aprendizaje.
4.	Una vez que el estudiante haya terminado cierta lección el sistema tutor le presentara una evaluación que debe rendirla, Dicha evaluación consta de un test con preguntas, de opción múltiple, etc. Cabe señalar que el estudiante solo tiene dos oportunidades para realizar dicha evaluación.
5.	Evaluación automatizada: Se contara con un registro donde se muestre todas las calificaciones que obtuvo y las observaciones correspondientes.
6.	Actualización de los temas y cuestionarios: El Administrador del Sistema podrá realizar actualizaciones tanto de los temas como de los cuestionarios de evaluaciones a sugerencia de los Docentes.
7.	Actualización automática de los datos de los estudiantes que hayan terminado con el proceso de enseñanza-aprendizaje de todos los temas, o abandonen el curso.
8.	El sistema tutor será utilizado como instrumento cognitivo por los estudiantes de la materia de base de datos y docentes como un recurso didáctico.
9.	El sistema tutor ayudara al estudiante a construir sus conocimientos a partir de contenidos de su interés en un entorno colaborativo y tecnológico.

Tabla 3.2 Funciones Principales del Sistema

[Fuente: Elaboración propia]

Atributos del Sistema

Algunos de los principales atributos del sistema son los siguientes:

- ✓ Interfaz: Interfaz fácil y amigable, Interactividad con el usuario, ventanas agradables a la vista del usuario.
- ✓ Tiempo de respuesta: En ingresar a las opciones del sistema es en tiempo de respuesta 5 a 10 segundos como máximo.
- ✓ Plataforma del S.O. Windows XP Sp2, para adelante.

Riesgos del sistema:

Las causas por las que con el sistema no se puedan llegar a satisfacer los objetivos planteados son los siguientes.

- ✓ El escaso equipamiento de hardware y software en el ambiente de instalación.
- ✓ Falta de conocimientos informáticos esenciales.
- ✓ Falta de predisposición e interés del estudiante para aprender.
- ✓ Falta de interés en la carrera para aplicar una nueva forma de aprendizaje, haciendo uso de tecnologías informáticas.

3.6.6. Casos de uso

Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor. Involucra la interacción de actores y el sistema. Un actor representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con estos.

Para tal efecto identificamos a los siguientes actores y el rol que cumplen para con el sistema:

Identificación de actores



- Estudiante
- Docentes
- Administrador del Sistema
- Agente

Rol de los actores

- ✓ Los estudiantes son los actores directos que interactúan con el sistema, aprenden para después dar una evaluación.
- ✓ Los Docentes son los que supervisan y guían al estudiante durante el proceso enseñanza-aprendizaje, al final pueden obtener del sistema tutor un historial del seguimiento de cada estudiante que ingreso al sistema. Pueden tomar también el rol del estudiante.
- ✓ El Administrador del sistema tiene la tarea de administrar la seguridad e integridad del sistema y las bases de datos. Es el único actor que puede hacer modificaciones al sistema de acuerdo a necesidades y requerimientos de los demás usuarios.
- ✓ El agente tiene la tarea de un tutor, es quien guía e instruye al estudiante.

Una vez identificados los actores que intervienen en el sistema, pasamos a identificar los siguientes casos de uso.

Identificación de Casos de Uso:

CASO DE USO 1: Acceso al sistema.

ACTORES: Estudiante, Docente, Administrador del Sistema.

PROPOSITO: El Estudiante accede al sistema para disponer de una herramienta que le ayuda a ampliar sus conocimientos sobre la materia.

El Docente accede al sistema para utilizarlo como un recurso didáctico en su labor educativa como practica.

El Administrador accede al sistema para actualizar los temas y evaluaciones a petición del Docente. También para hacer el mantenimiento respectivo como realizar mejoras.

TIPO: Primario y esencial

CASO DE USO 2. El estudiante realiza su inscripción en el sistema, si es que ingreso por primera vez.

ACTORES: Estudiante, administrador del sistema, docente.

PROPOSITO: El estudiante se inscribe en el sistema introduciendo todos sus datos personales para que el sistema le haga un seguimiento y se tenga un historial de su avance y aprovechamiento.

TIPO: Primario y esencial.

CASO DE USO 3: El estudiante elige el tema que desea aprender.

ACTORES: Estudiante, Docente.

PROPOSITO: Enseñar el tema que el usuario desee aprender. A partir de ello el Docente contara con el seguimiento a cada estudiante.

CASO DE USO 4: El estudiante inicia su proceso de enseñanza-aprendizaje

ACTORES: Estudiante, Docente, Agente.

PROPOSITO: El sistema tutor inicia la sesión de enseñanza del tema elegido por el estudiante.

El Docente sigue el ritmo del proceso de enseñanza para aportar con sus conocimientos y experiencias con el manejo del lenguaje.

El agente del sistema es el guía que colabora al estudiante durante el proceso de enseñanza-aprendizaje.

TIPO: Primario y esencial

CASO DE USO 5: El estudiante decide dar la evaluación correspondiente al tema aprendido para medir el grado de su aprovechamiento.

ACTORES: Estudiante, Agente

PROPOSITO: Evaluar para determinar el grado de aprovechamiento y asimilación del tema, diseñado.

TIPO-. Primario y esencial.

CASO DE USO 6: Evaluación del proceso de enseñanza-aprendizaje

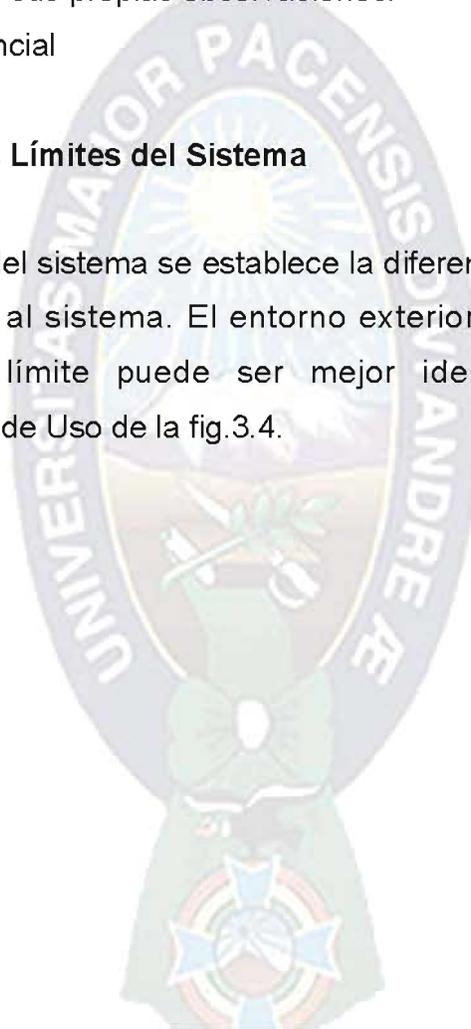
ACTORES: Estudiante, Docente, Agente.

PROPOSITO: Obtener una calificación automática que muestre cuantitativamente los resultados alcanzados por el estudiante. Para que el Docente cuente con dicho reporte y emita sus propias observaciones.

TIPO: Primario y esencial

Identificación de los Límites del Sistema

Al definir los límites del sistema se establece la diferencia entre lo que es interno y lo que es externo al sistema. El entorno exterior se representa mediante los actores, este límite puede ser mejor identificado observando el Diagrama de Casos de Uso de la fig.3.4.



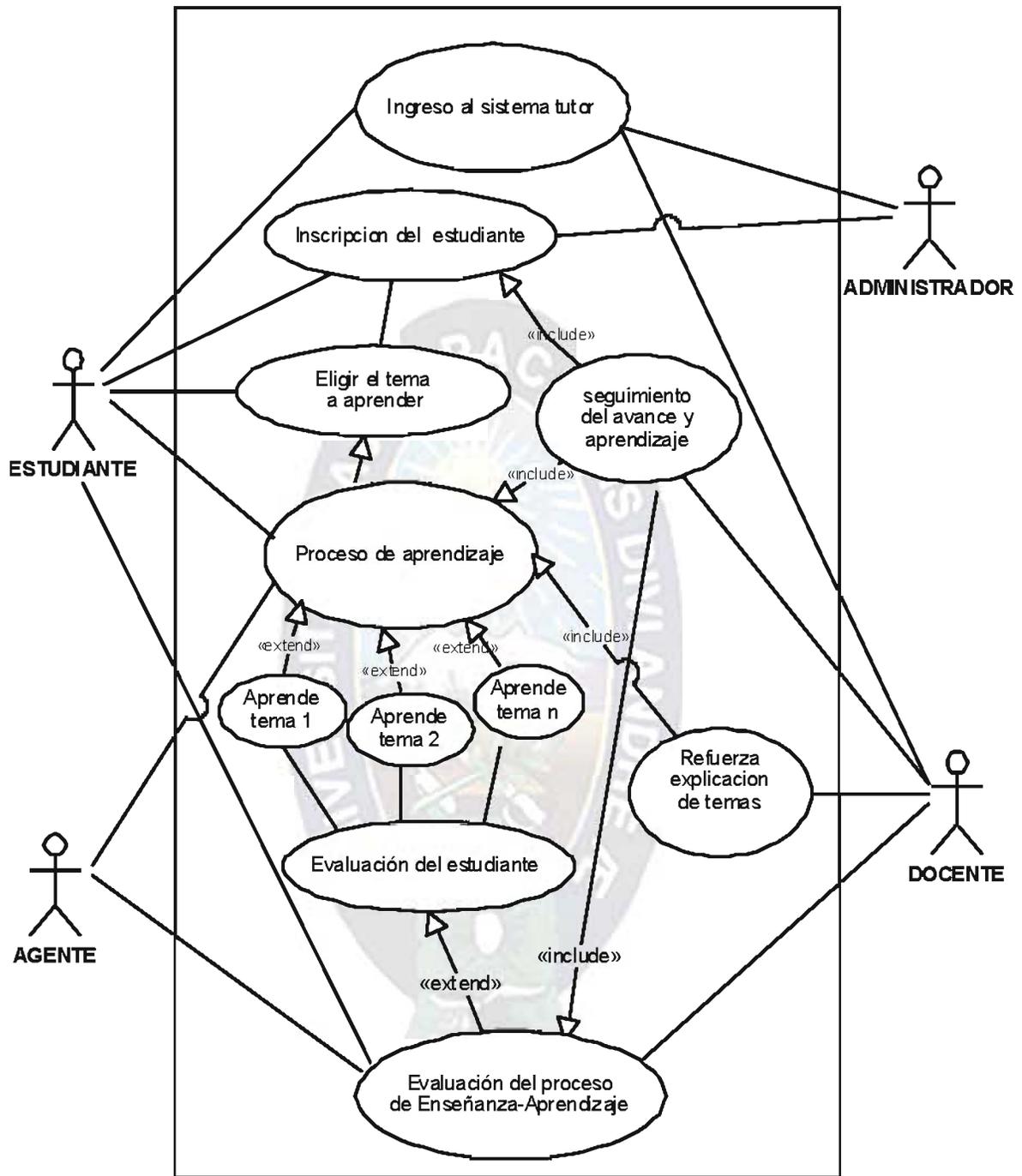


Fig.3.4 DIAGRAMA DE CASOS DE USO
 [FUENTE: Elaboración propia]

Fase de construcción: análisis

En esta fase se verá el análisis del problema a resolver desde la perspectiva de los usuarios y de las entidades externas que van a solicitar servicios al sistema.

Las actividades que esta etapa comprende: la refinación del Diagrama de Casos de Uso ya que pudo haber sido definido en el punto anterior, la definición de los Diagramas de Secuencia del sistema, Diagrama de Estados.

3.6.7. Diagrama de casos de uso

Diagrama de Casos de Uso nos muestra la relación entre los actores y los casos de uso respectivamente. La Fig.3.4 describe la relación que existe entre los actores y los distintos casos de uso identificados con anterioridad.

3.6.8. Diagrama de secuencia

El Diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso. El diagrama de secuencia contiene Detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Una parte de la descripción del comportamiento del sistema un diagrama de secuencia destaca la ordenación temporal de los mensajes. Se construye el Diagrama de Secuencias utilizando cada caso de uso definido anteriormente. Véase fig.3.5.

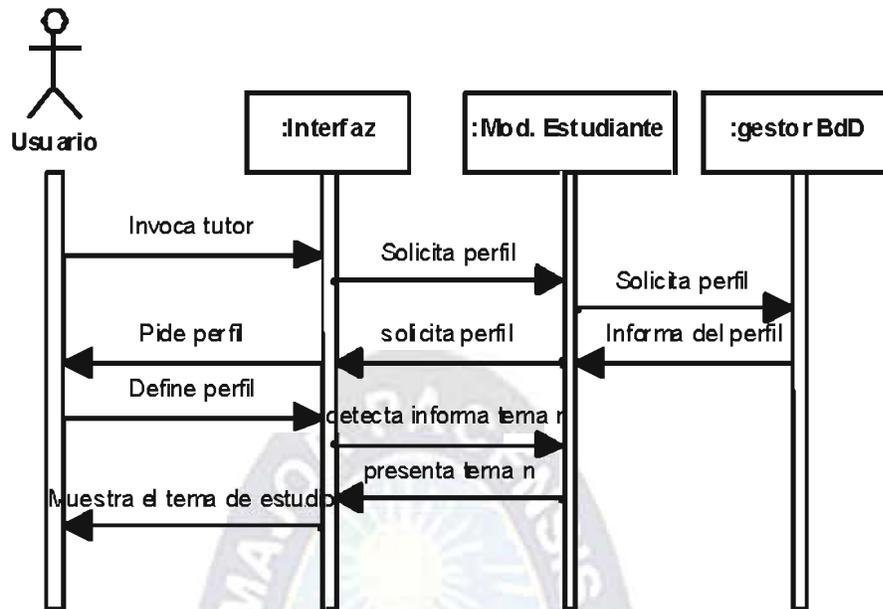


Fig.3.5 Diagrama de Secuencia para el caso de uso: Acceso al Sistema Tutor.

[Fuente: Elaboración Propia]

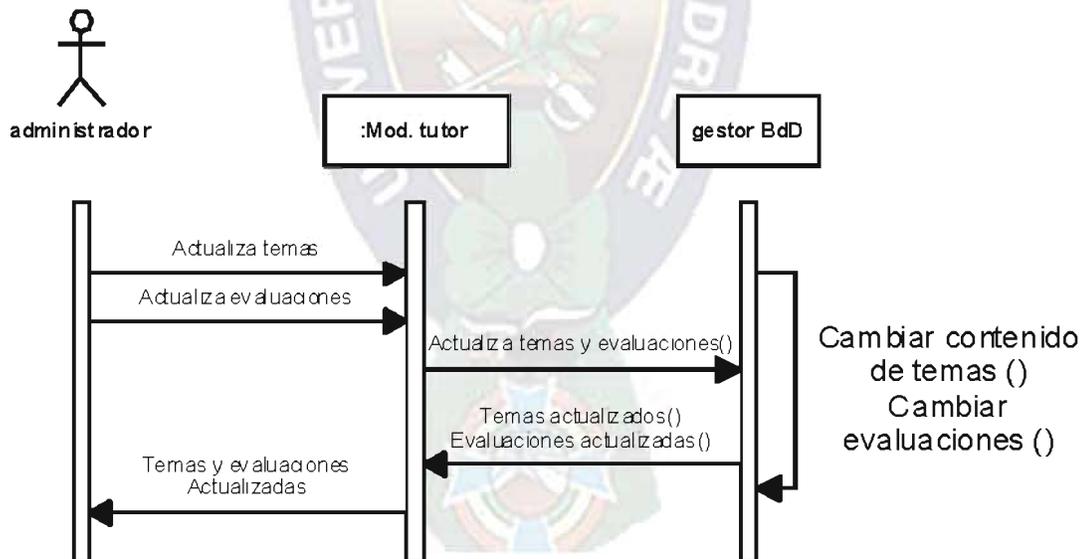


Fig.3.6 Diagrama de Secuencia para el caso de uso: Actualizar temas y evaluaciones.

[Fuente: Elaboración Propia]

3.6.9. Diagrama de estados

Un diagrama de estado se modela para todas las clases que se consideran con un comportamiento dinámico. En él, modelas la secuencia de estado que un objeto de la clase atraviesa durante su vida en respuesta a los estímulos recibidos, junto con sus propias respuestas y acciones.

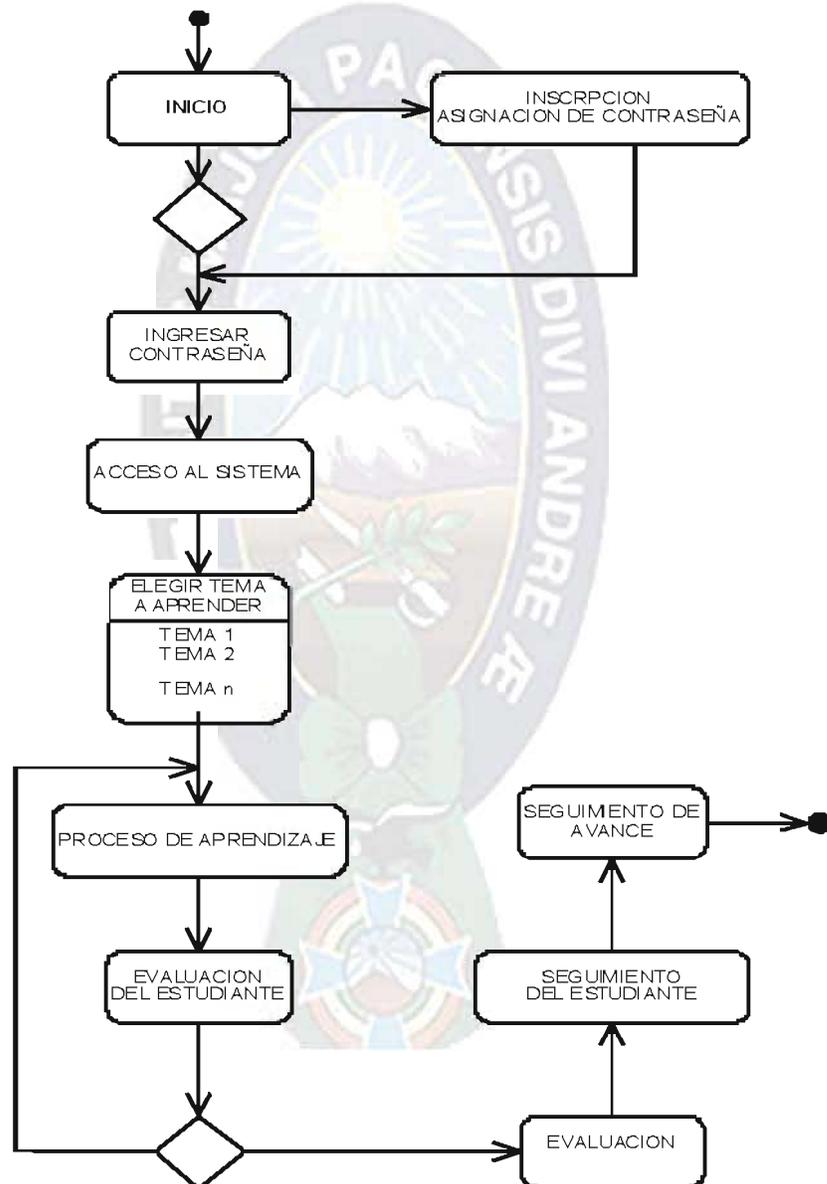


Fig.3.7 Diagrama de estados

[Fuente: Elaboración Propia]

3.7. Fase de construcción: diseño

En la fase de diseño se crea una solución a nivel lógico para satisfacer los requisitos, basándose en el conocimiento reunido en la fase de Análisis.

En esta etapa se definen los Diagramas de Interacción, el Diseño Conceptual y el Diagrama de Clases de Diseño, también se define el esquema de la Base de Datos.

3.7.1. Diagramas de interacción

Los Diagramas de Interacción tanto como los de Estado sirven para modelar la parte Dinámica del sistema. Muestran el intercambio de mensajes entre instancias del modelo de clases. Existen dos tipos de diagramas de interacción: Los Diagramas de Colaboración y los Diagramas de Secuencia vistos anteriormente. De estos dos, se prefiere los primeros por su expresividad (una interacción compleja resultaría muy larga en un Diagrama de Secuencia).

3.7.2. Diagramas de colaboración

El Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos.

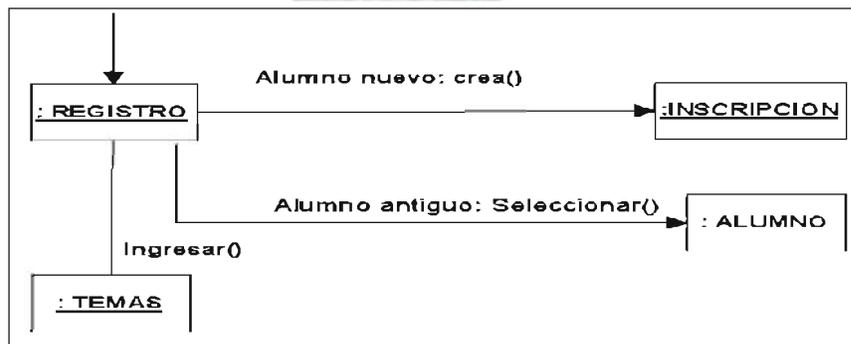
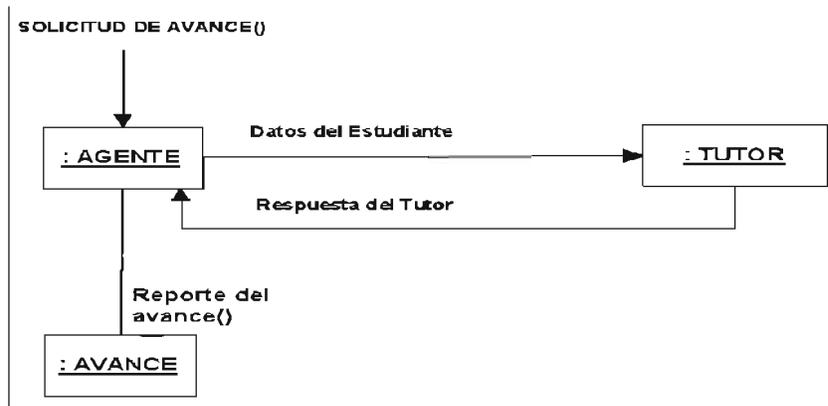


Fig.3.8 Diagrama de Colaboración para el caso: Registro de estudiante.

[Fuente: Elaboración propia]



**Fig.3.9 Diagrama de Colaboración para el caso:
Solicitud de Avance y Seguimiento**

[Fuente: Elaboración propia]

3.7.3. Diagrama de clases de diseño

Son diagramas que modelan la parte estática del sistema, el conjunto de todas las clases usadas en los apartados anteriores, junto con sus relaciones, forman el Diagrama de Clases de Diseño. La conjunción de los diagramas estáticos y dinámicos da como resultado un sistema bien estructurado.

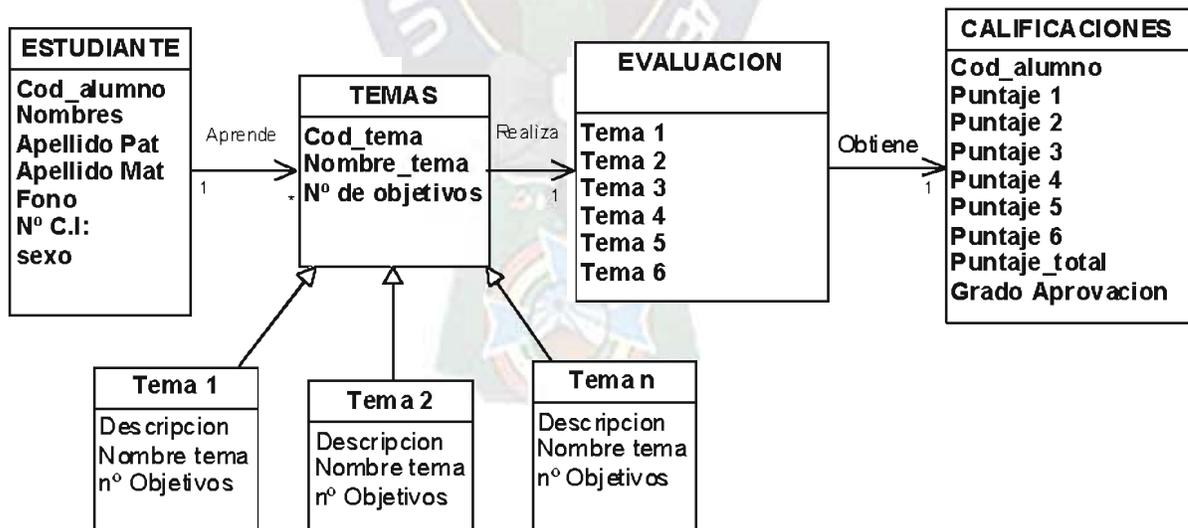


Fig.3.10 Diagrama de Clases principal del Sistema

[Fuente: Elaboración propia]

3.7.4. Diagrama de diseño lógico

Aquí se especifica mediante el uso de clases la arquitectura general del sistema. Para el caso de la implementación existe una clase base que es la encargada de la representación del conocimiento el tutor está constituido directamente sobre el conocimiento. La pedagogía y la curricular son formas de conocimiento, es por eso que heredan todas sus cualidades. Por último existe una clase que hereda todo el conocimiento e interactúa con el estudiante fig. 3.11 muestre el diseño orientado a objetos del tutor.

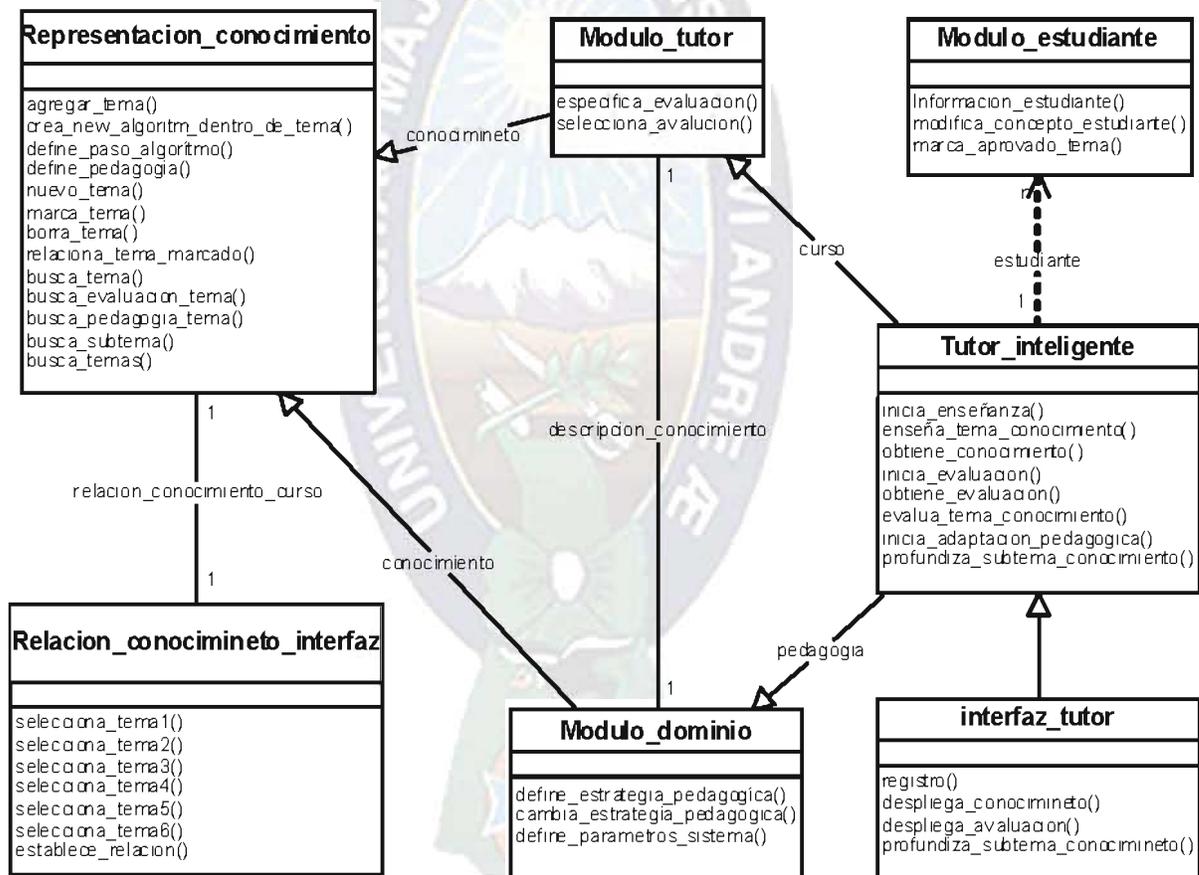


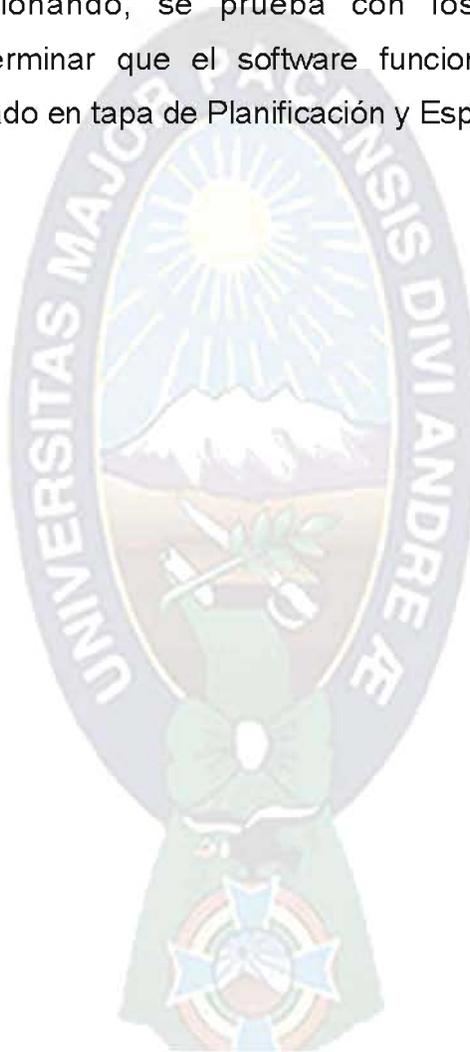
Fig.3.11 Diagrama de diseño lógico del Sistema

[Fuente: Elaboración propia]

3.8. Fase de construcción: implementación y pruebas

Una vez que se definió el Diagrama de Clases de Diseño, se pasa a la fase de implementación, donde se lleva lo especificado al lenguaje de programación para implementar el prototipo.

Una vez obtenido el programa, este se depura y cuando ya se tiene una gran parte sistema funcionando, se prueba con los futuros usuarios del sistema, para determinar que el software funciona correctamente y que satisface lo especificado en etapa de Planificación y Especificación de Requisitos.



[CAPITULO 4]

IMPLEMENTACION DEL PROTOTIPO Y MEDICION DE LA CALIDAD DEL SOFTWARE

4.1. Implementación del prototipo

En este capítulo, se describe la implementación concreta de las ideas, en un sistema computacional que cubra todas las características requeridas. Todo el desarrollo del Sistema Tutor Inteligente (STI) requiere de muchísimo más tiempo, por el cual este trabajo solo aspira a implementarse como prototipo, que permita evaluar los resultados de las técnicas, y herramientas seleccionadas en los capítulos anteriores, se requiriera de un desarrollo incremental y continuo para implementarlo como acabado el Sistema.

Descripción de la interfaz interactiva

Todo sistema debe tener en principio una pantalla inicial, y si es atractiva a la vista del usuario resulta más agradable la interacción entre el sistema y el usuario propiamente dicho. La fig.4.1 nos muestra la pantalla principal del sistema, en ella se pueden observar opciones:



Fig. 4.1 Pantalla Inicial del Sistema

[Fuente: Elaboración Propia]

Desde la interfaz que se observa en la Fig. 4.1 si el estudiante es antiguo puede seleccionar su nombre de la lista e introducir su contraseña que representa el código que se le asigna cuando efectúa su inscripción, “Ingresar”, que permite el acceso al Sistema Tutor una vez comprobada la contraseña tras haberla verificado en la base de datos del estudiante.

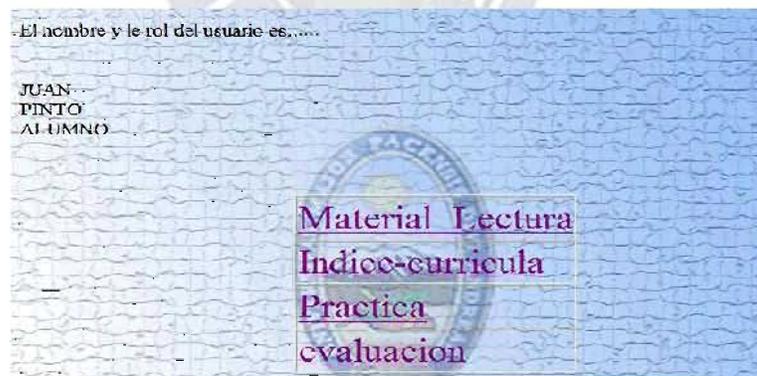


Fig. 4.2 Pantalla de selección de tarea a realizar

[Fuente: Elaboración Propia]

En la siguiente figura puede observarse la pantalla que muestra los temas a enseñar, de los cuales el estudiante puede elegir para su aprendizaje. A partir de esta, la enseñanza se toma personalizada porque esta registra su nombre, también muestra el botón “Avance de Temas”, que proporciona un reporte de todo el proceso de aprendizaje del estudiante.



Fig. 4.3 Índice de la curricular
 [Fuente: Elaboración Propia]

La figura 4.4 muestra el desarrollo de la primera lección, en ella como en las siguientes pueden observarse el nombre del estudiante que está siguiendo el proceso de enseñanza y un botón que dice “mas Información”, que le permite al estudiante transportarse a otra pantalla dinámica que amplía sus conocimientos.

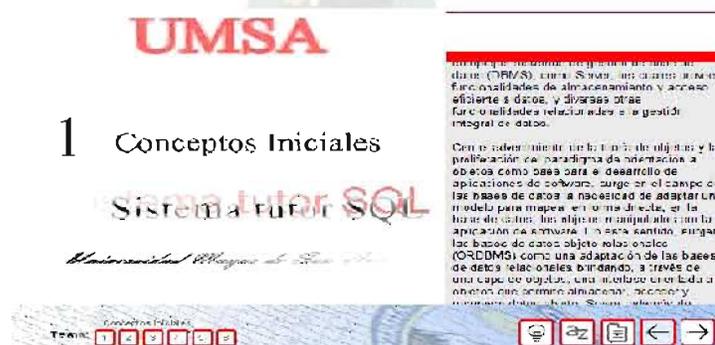


Fig. 4.4 Pantalla del primer tema
 [Fuente: Elaboración Propia]

En esta pantalla al mismo tiempos se pueden observar en la parte superior a los botones que le permitirán al estudiante ubicarse en las distintas partes del desarrollo del tema.



Fig. 4.5 Botones de navegación

[Fuente: Elaboración Propia]

En esta figura se pueden observar las siguientes opciones:

“Sigue”, permite ubicarse al estudiante en una pantalla después de la actual del tema en estudio.

“Atrás”, permite ubicarse al estudiante en una pantalla antes de la actual del tema en estudio.

“Índice” que nos mostrar el índice de temas a cursar, como ayuda.

“Glosario” nos ayudara a entender mejor algunas palabras sobre la sintaxis básica del lenguaje SQL, como también alguna terminología del lenguaje.

La parte de Práctica, es donde el estudiante podrá plasmar código, sentencias SQL, esta parte le permite al estudiante poner en práctica todo lo que va avanzado según vaya avanzando al estudiar la curricular del tutor, podemos practicar con los diferentes ejemplos que se proponen en la misma. En esta parte está instalado el SQLPLUS de Oracle 10g en su versión de libre figura 4.6.

Evaluación "Quiz", **Questions** da la opción al estudiante a realizar su evaluación correspondiente ya que finalizo con su proceso de aprendizaje del tema elegido.

Espacio de Trabajo

Introduzca sentencias SQL, PL/SQL y SQL*Plus.

```
select *
from user_tables
```

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOI_NAME	STATUS	PCI_FREE	PCI_USED	INI_TRANS	N
REGIONS	EXAMPLE			VALID	10			1
COUNTRIES				VALID	0	0		0
LOCATIONS	EXAMPLE			VALID	10			1
DEPARTMENTS	EXAMPLE			VALID	10			1
JOB	EXAMPLE			VALID	10			1
EMPLOYEES	EXAMPLE			VALID	10			1
JOB_HISTORY	EXAMPLE			VALID	10			1

Fig. 4.6 Pantalla consola de sqlplus

y si aun no cree estar seguro de rendirla, tiene la opción a retomar el tema, mediante los botones de "temas" donde puede navegar por los distintos temas.

EL NOMBRE DEL USUARIO, SU ROL, Y SU CODIGO DE REGISTRO ES.....

JUAN
PINTO
ALUMNO
3

PORFAVOR ELIJA EVALUACION:

Mucha suerte!!!!!!!

Fig. 4.7 Selección de evaluación

[Fuente: Elaboración Propia]

En la siguiente figura mostráramos como se evaluara al estudiante, es un evaluación de selección múltiple una vez terminado cliqueamos en verificar (bien) y nos mostrara la calificación obtenida esta será la nota que obtuvo el estudiante.

SISTEMA TUTOR DOL Módulo evaluación · Tema 1

Arrastre las opciones hacia la posición correcta.

Drag & Drop

12 Clasifique los grupos de sentencias SQL que se presentan según su tipo.

Resultados		
	Link Asociado	
✓ 1	1.1.3	
X 2	1.1.3	
✓ 3	1.1.4	
✓ 4	1.1.6	
X 5	1.1.6	
X 6	1.1.6	
✓ 7	1.1.6	
X 8	1.1.11	
X 9	1.1.8	
X 10	1.1.9	
Porcentaje 42%		

Pregunta: 1 2 3 4 5 6 7 8 9 10 11 12

✓ ↶ ↷ ↵ ↶ ↷

Fig. 4.8 Evaluación al estudiante [Fuente: Elaboración Propia]

Un elemento clave de cualquier proceso de Ingeniería de Software es la medición con el objetivo de valorar la calidad del producto de ingeniería o sistema. A diferencia de otras disciplinas, la Ingeniería de Software no está basada en leyes cuantitativas básicas de la Física. En su lugar se intenta obtener un conjunto de medidas indirectas que dan lugar a métricas que proporcionan una indicación de la calidad de algún tipo de representación del software.

Aunque las medidas y métricas del software aun están en debate porque no son absolutas, nos proporcionan una manera sistemática de valorar la calidad basándose en un conjunto de reglas claramente definidas. También nos proporciona una visión interna en el acto en vez de a posteriori, para corregir problemas a tiempo.

4.2. Métricas orientadas a la función:

Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como valor de normalización. Ya que la

funcionalidad no se puede medir directamente, se debe derivar indirectamente otras medidas directas. Estas métricas se desglosaran en la parte de anexos.

4.3. Métrica de mantenimiento

Para aplicar esta métrica de mantenimiento se realiza mediante el método del índice de madurez propuesta por el estándar IEEE [IEEE 94] que proporciona una indicación de la estabilidad del producto de software basada en cambios que ocurren con cada versión del producto. La relación matemática que utiliza es la siguiente:

$$IMS = \frac{[Ma - (Fa + Fc + Fb)]}{Ma} \quad \text{ec.2}$$

Donde:

IMS Índice de madurez del software.

Ma Número de módulos en la versión actual.

Fa Número de módulos en la versión actual que se han añadido.

Fc Número de módulos en la versión anterior que se han cambiado.

Fb Número de módulos en la versión anterior que se han borrado en la versión actual.

Si el valor de IMS se aproxima a 1 significa que el software empieza a estabilizarse.

Entonces haciendo los cálculos correspondientes con la ec.2 se tiene:

$$IMS = [6 - (0+0+0) / 6] = 1$$

Por tanto se concluye que el presente sistema tiene una estabilidad del 100%

4.4. Portabilidad

Se dice que un sistema es portable, si hablando de costos, el costo de transportar y adaptar a un nuevo ambiente es menor al costo de rediseñar el sistema para el mismo ambiente. La relación que permite determinar el grado de portabilidad es la siguiente:

$$GP = 1 - \text{costo de portabilidad} / \text{costo de rediseño}$$

Donde:

GP: Grado de portabilidad.

Portabilidad efectiva: $GP > 0$ Portabilidad perfecta: $GP = 1$

4.5. Adaptabilidad

La adaptabilidad es una componente más de la usabilidad. Si se consigue lograr una buena adaptabilidad se estará optimizando la usabilidad. El sistema es adaptable puesto que la programación realizada en el mismo no es compleja y está sujeta a cualquier cambio o modificación. Al mismo tiempo las estructuras de las tablas son simples, adaptables a un nuevo ambiente.

4.6. Prueba del sistema tutor SQL

Después de haber aplicado parámetros que nos dan una medida cuantitativa acerca de la medición de la calidad del software, realizamos una prueba del tutor para reflejar el grado de satisfacción del usuario, para ello se realizó un cuestionario dirigido a los estudiantes de taller de base de datos.

NOMBRE Y APELLIDOS.....

SEXO.....FECHA.....



Preguntas:

Respuestas:

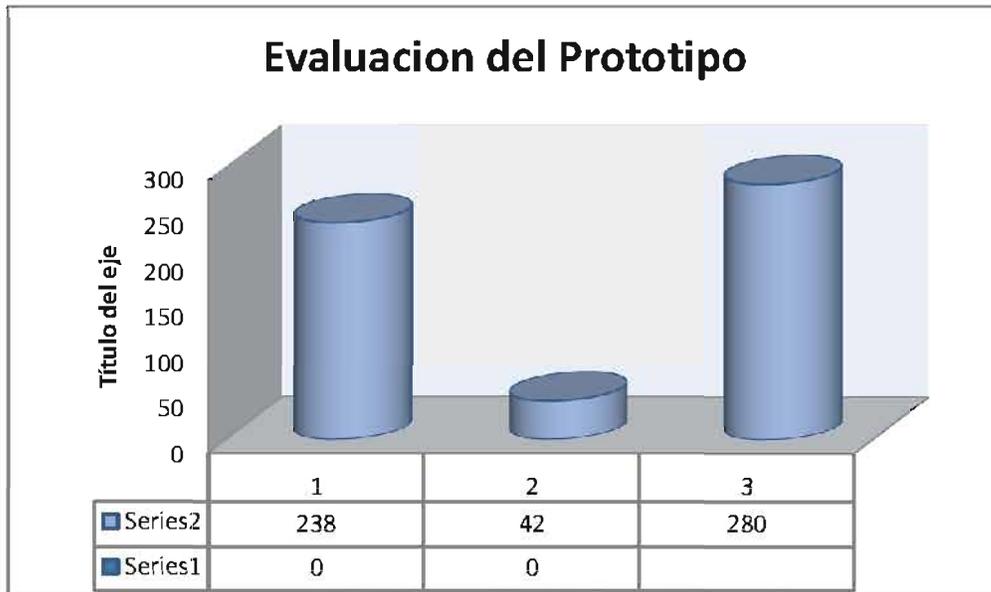
1. ¿Es sencillo de manipular el tutor?		
2. ¿El glosario del tutor es suficiente?		
3. ¿Te gusta los temas presentados?		
4. ¿Está de acuerdo con la presentación en flash?		
5. ¿Te parece interesante los temas expuestos?		
6. ¿Es fácil la entrada de datos?		
7. ¿Te gusto la experiencia con Oracle como lenguaje de consulta y manipulación de datos?		
8. ¿Te parece amigable el tutor?		
9. ¿Te gusta lo que aprendes ponerlo en práctica?		
10. ¿Te gusta estar frente al computador mucho tiempo?		
11. ¿Quisieras que el tutor esté disponible por internet?		
12. ¿Piensas que el tutor es solución a las clases numerosas?		
13. ¿Estás conforme con la evaluación del tutor?		
14. ¿Te gustaría que el tutor sea adaptable para otros lenguajes de programación?		

Conteo de preguntas

Pregunta	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Si	16	15	18	19	15	17	16	16	19	20	17	15	17	18

No 4 5 2 1 5 3 4 4 1 0 3 5 3 2

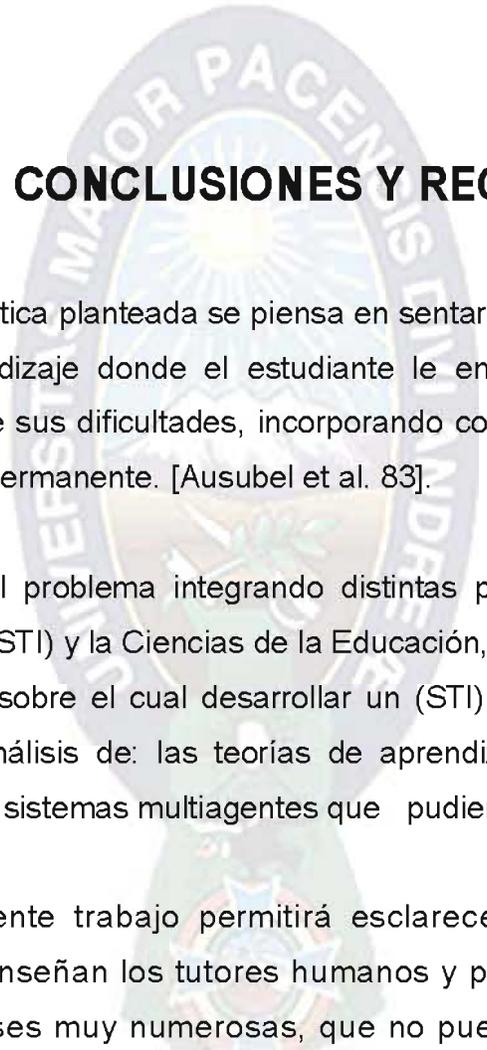
N=20 Si = 238 No =42



Donde N = Es el número de estudiantes encuestados

	Si	No
Total	239	41
Porcentaje	0.85 = 85%	0.15 = 15%

Observando los resultados del cuestionario se determina que el sistema tutor SQL tiene un 85% de aceptación y se comprueba que ayuda la enseñanza, con la hipótesis queda comprobada.



[CAPITULO 5]

CONCLUSIONES Y RECOMENDACIONES

Debido a la problemática planteada se piensa en sentar las bases para un sistema de enseñanza-aprendizaje donde el estudiante le encuentre significado a sus acciones, que supere sus dificultades, incorporando conocimientos nuevos de un modo significativo y permanente. [Ausubel et al. 83].

Se ha investigado el problema integrando distintas perspectivas: los Sistemas Tutores Inteligentes (STI) y la Ciencias de la Educación, a fin de obtener una base teórica consolidada, sobre el cual desarrollar un (STI), se han tenido en cuenta particularmente el análisis de: las teorías de aprendizaje, de enseñanza y las características de los sistemas multiagentes que pudieran aportar a la propuesta.

El aporte del presente trabajo permitirá esclarecer algunos interrogantes respecto de cómo enseñan los tutores humanos y podrían dar solución a los estudiantes de clases muy numerosas, que no pueden acceder a consultar al docente durante el horario habitual de clases. Tomando en cuenta consideraciones señaladas, el objetivo es un sistema que, aprovechando los recursos disponibles en la actualidad, permita disminuir las tareas de los tutores humanos y que a la vez mejore la experiencia de aprendizaje desde la perspectiva del estudiante

Por lo expuesto con anterioridad podremos nos permite afirmar que, la hipótesis junto a los objetivos planteados están cumplidos. El sistema tutor fue diseñado y desarrollado además de probado, sin embargo dependerá mucho del alumno como del profesor el uso adecuado del mismo. El sistema tutor para enseñanza del lenguaje SQL, proporciona una adecuada enseñanza mediante el enfoque cognitivo. El tutor le permita establecer un entorno de aprendizaje amigable, sencillo, fácil en su manipulación, interactivo para el usuario, por lo tanto nos permite obtener un buen grado de efectividad educativa.

Se logro establecer la estructura diseñada introduciendo en la misma el modulo de control y evaluación. Cumpliendo de esta manera con los objetivos, logrando una aceptación considerable por parte de estudiantes, cumpliendo de esta manera con el objetivo.

Existen diferentes experiencias de investigación de agentes inteligentes aplicados a la educación, las cuales apoyan sustancialmente la relación entre los usuarios y los sistemas de computación. Sin embargo, uno de los problemas más grandes en el desarrollo e implementación de agentes educativos es la dificultad y el costo, debido a la multiplicidad y complejidad que estos agentes representan.

El trabajo pretendía demostrar la hipótesis que; basado en teorías de agentes inteligentes de construye un tutor inteligente que es viable para conseguir un proceso de asesoría para que el estudiante aprenda el lenguaje SQL.

Por ende podemos afirmar que el objetivo general que es "Desarrollar un prototipo de sistema tutor para enseñanza del lenguaje SQL, para estudiantes de la materia de taller de base de datos, está cumplido.

La gran ventaja que posee los estudiantes de esta materia, es que se aprovechara como un docente fuera de clases, ya que para ellos fue simplemente aplicar el proceso inverso de aprendizaje que consiste en ir de la práctica a la teoría.

5.1. Estado de la hipótesis

En el planteamiento de la hipótesis que es:

“El sistema tutor para enseñanza del lenguaje SQL, contribuirá a mejorar el nivel de aprendizaje de los estudiantes de la materia de taller de base de datos”.

La hipótesis se valida ya que se ha encontrado un herramienta que es capaz de determinar el estilo de enseñanza partiendo de la curricular que cuenta la materia. Dándole al estudiante tutorizado tanto teórico como práctica.

Después de haber analizado los resultados obtenidos mediante el experimento, podemos decir que la hipótesis planteada, queda demostrada, puesto que los valores obtenidos muestran un notable incremento. Por tanto se comprueba la hipótesis planteada en el primer capítulo.

5.2. Recomendaciones y futuras líneas de investigación

La utilización del sistema tutor debe ser adecuada para su buen funcionamiento. Los agentes pueden habitar en diferentes tecnologías educativas como ser vía web, en sistemas multiusuario, monousuario, etc. Se puede aplicar el uso de agentes de una manera más eficiente lo que se logra con los agentes inteligentes. Se recomienda realizar trabajos que estén orientados al área educativa, ya que es esta donde se percibe muchas necesidades.

Deberían realizarse tutores educativos que abarquen más áreas del conocimiento para facilitar a los estudiantes de herramientas que le ayuden en su formación personal.

Por lo tanto una de las futuras líneas de investigación, para ampliar el desarrollo de este trabajo es la de adaptar las ideas aquí planteadas a otras estructuras similares a la de los Sistemas Tutores Inteligentes (STI), como pueden ser los Asesores Inteligentes o mismos Sistemas Tutores Inteligentes que no posean un modelo de interacción tradicional, de aquí también se desprende que las mismas ideas pueden ser adoptadas a modelos en los que no se aplica el marco de referencia tripartito, planteado por Carbonell, acatando y que distribuyendo los submódulos que desempeñan las tareas del tutor a las nuevas estructuras, que si bien difieren de la estructura clásica deberían seguir cumpliendo las mismas funciones y logrando los mismos objetivos

Otro campo en el que la investigación puede seguir en su curso, es en el de su agregado de nuevos módulos a la estructura clásica. Se mantienen los tres módulos básicos, pero se agrega módulos para ampliar los campos de aplicación del sistema, como puede ser el módulo evaluador o el módulo ambiental (ecosistema), propuesto por varios autores. De esta manera las interfaces que unen a los diferentes modelos añadidos deberían ser definidas para aquí la interacción con el módulo de tutor se realice de forma eficiente y sin la superposición de funcionalidades (solapacion de funciones), que tanto ha demorado al desarrollo del Sistemas Tutores Inteligentes implantables.

Anexo A: Ejemplo del curso del esquema (HR) Recursos Humanos

Los ejemplos utilizados en este tutor, pertenecen una aplicación de recursos humanos (HR), que se puede crear como parte de la base de datos inicial. Para la implementación mediante código. A continuación, se muestran algunas reglas principales de negocio de la aplicación (HR):

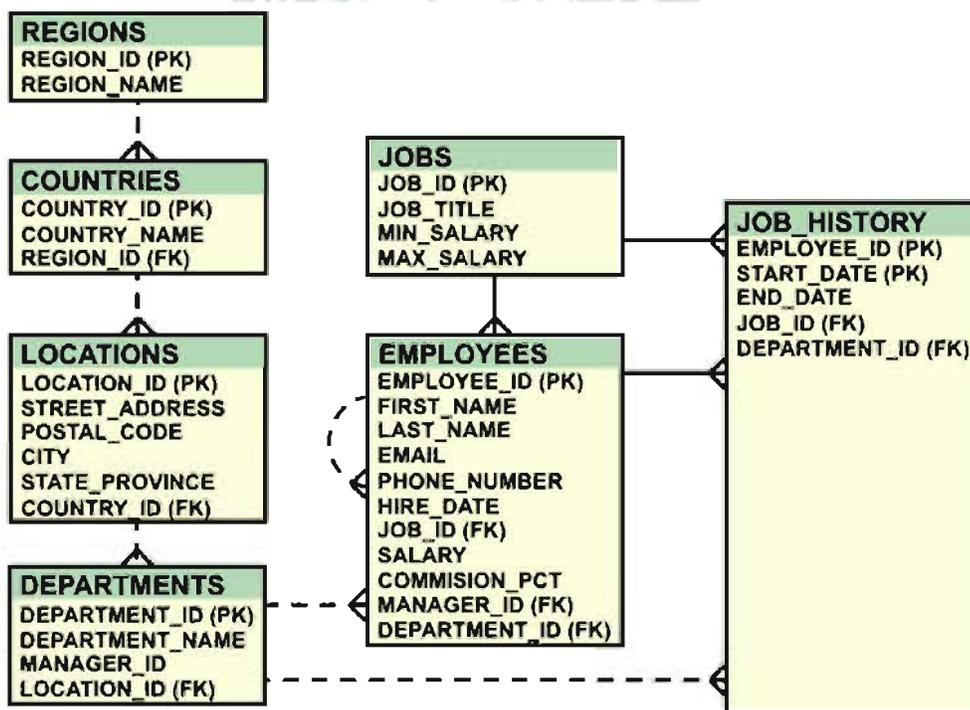


Fig. A.1: Esquema de recursos humanos (HR)

- Cada departamento puede ser el empleador de uno o más empleados.

Cada empleado se puede asignar a un único departamento.

- Cada puesto debe ser para uno o más empleados. Cada empleado debe estar asignado a un único puesto simultáneamente.
- Cuando un empleado cambia su departamento o puesto, un registro de la tabla JOB_HISTORY registra las fechas de inicio y de finalización de las asignaciones pasadas.
- Los registros de JOB_HISTORY se identifican mediante una clave primaria compuesta (PK): las columnas EMPLOYEE_ID y START_DATE.

Notación: PK = clave primaria, FK = clave ajena Las líneas solidas representan las restricciones obligatorias de clave ajena y las líneas discontinuas las opcionales. La tabla EMPLOYEES también tiene una restricción de clave ajena consigo misma esta es una implementación de la regla de negocio: Cada empleado informa directamente a un único supervisor. La clave ajena es opcional porque el empleado de nivel superior no informa a otro empleado.

Un pequeño ejemplo de cómo se crea una de las tablas mencionadas:

```
CREATE TABLE employees
( employee_id NUMBER(6)
, first_name VARCHAR2 (20)
, last_name VARCHAR2 (25)
  CONSTRAINT emp_last_name_nn_demo NOT NULL
, email VARCHAR2 (25)
  CONSTRAINT emp_email_nn_demo NOT NULL
, phone_number VARCHAR2 (20)
, hire_date DATE DEFAULT SYSDATE
  CONSTRAINT emp_hire_date_nn_demo NOT NULL
, job_id VARCHAR2(10)
  CONSTRAINT emp_job_nn_demo NOT NULL
```

En la siguiente figura vemos como de conforma un esquema en nuestro caso (HR), y lo que contiene como son las tablas, los objetos, índices, funciones, ect.

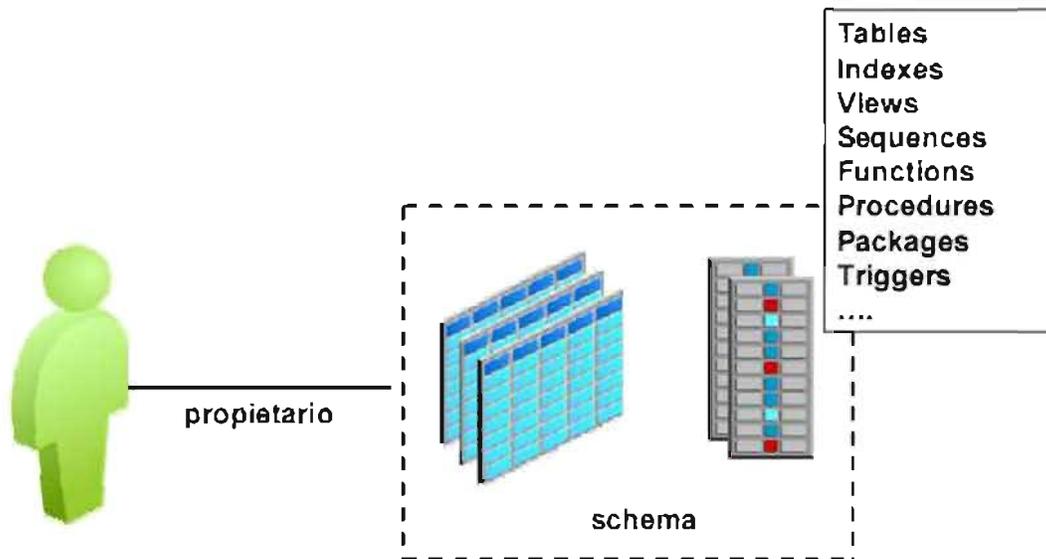


Fig. A.2: Como conforma un Esquema

Anexo b: Métricas orientadas a la función

Punto función

Para aplicar esta métrica al sistema elaborado, inicialmente definiremos las entradas del usuario que llega desde la frontera externa hacia el sistema.

Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deben diferenciar de las peticiones, estas pueden ser el inscribirse, ingreso de datos o control de flujo de la información, preparar un examen, evaluarse. Dichas entradas se llevan a cabo mediante las siguientes interfaces.

- **Pantalla principal**
- **Pantalla de elección de temas**
- **Pantalla de evaluaciones**
- **Pantalla de reportes**

Para las salidas del usuario se cuenta cada salida que proporciona al usuario información orientada a la aplicación. Las salidas del usuario representan los informes, pantallas, mensajes de error.

En nuestro caso tales salidas son: los mensajes de error que emite el sistema por ejemplo, cuando un alumno ingresa una contraseña errónea

o cuando no está registrado, las pantallas que muestran los resultados de exámenes revisados, dar la nota de su evaluación. Estas pueden verse en las siguientes interfaces:

- **Pantalla de mensajes de error**
- **Pantalla que muestra las lecciones**
- **Pantalla de evaluación**
- **Pantalla de notas.**

Para determinar el número de peticiones del usuario: definimos que una petición es una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva.

Se cuenta por separado cada petición, en este caso una petición viene a ser: la consulta de notas del alumno, su avance y seguimiento, lo cual se observa en las siguientes interfaces:

- Pantalla de consulta de notas
- **Pantalla de avance y seguimiento**

Para determinar el número de archivos se cuenta cada archivo maestro lógico, es decir un grupo lógico de datos que pueden ser una parte de una gran base de datos o un archivo independiente. Dicho número determinamos de los siguientes archivos:

- **Archivo Alumno**
- **Archivo Temas**
- **Archivo Avance**
- **Archivo Notas**

El número de Interfaces externas representa todas las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema, en nuestro caso no se cuenta. Una vez recopilados los datos se asocia un valor de complejidad que puede ser simple, mediana o compleja, la determinación de este valor es subjetiva. Todos los datos recopilados los llevamos a la siguiente tabla:

Parámetro de Medición	Cuenta	Factor de Ponderación Medio	Subtotal
Número de entradas del	1	4	4
Número de salidas de usuario	2	5	10
Número de peticiones de	1	4	4
Numero de archivos	2	10	20
Numero de interfaces externas	0	7	=0
Cuenta Total			38

Tabla A.1: Calculo de Punto Función

La cuenta total obtenida se ajusta ahora en función a las características ambientales del sistema, las mismas que se muestran en la siguiente tabla y cada factor debe ser evaluado en una escala de 0 a 5.

Características del sistema	No influyente	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	1	2	3	4	5		
1. ¿Requiere el sistema copia de seguridad y de recuperación fiable?						😊	5
2. ¿Se requiere comunicación de datos?					😊		4
3. ¿Existen funciones de Procesamiento				😊			3
4. ¿Es crítico el rendimiento?			😊				2
5. ¿Se ejecutara el sistema en un entorno operativo existente						😊	5
6. ¿Requiere el sistema entrada de datos interactiva?						😊	5

7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo	😊	5
8. ¿Se actualizan los archivos maestros de forma interactiva?	😊	5
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	😊	1
10. ¿Es complejo el procesamiento interno?	😊	1
11. ¿Se ha diseñado el código para ser reutilizable?	😊	5
12. ¿Están incluidas en el diseño la conversión y la instalación?	😊	4
13. Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	😊	5
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	😊	5

Tabla A.2: Características ambientales del sistema [Pressman, 03]

Entonces se tiene:

$$\sum = fi = 55$$

Cuenta total: es el total de puntos de función sin ajustar.

0.01: Es el error mínimo aceptable de la complejidad.

$\sum = fi$ Es la sumatoria de los factores de complejidad del proceso, que puede variar en el rango de:

$$0 \leq \sum f_i \leq 70 \quad i = 1, \dots, 14$$

Si los factores de complejidad de proceso no influyen en nada $\sum f_i = 0$, los puntos de función sin ajustar serán disminuidos en un 35% y si influyen mucho

$\sum f_i = 70$, este valor será aumentado en un 35%, así: **0.65**: es el pie de corrección.

Entonces reemplazando los valores en la ecuación de punto función se tiene:

$$PF = Cuent_{total} * [0.65 + 0.01 * 6(f_i)] \quad ecc 1$$

Reemplazando se tiene

$$PF = 38 * [0.65 + 0.01 * 55]$$

$$PF = 45.6$$

Comparando los valores de funcionalidad del sistema con el PF máximo:

$$PF = 38 * [0.65 + 0.01 * 70]$$

$$PF = 51.3$$

Con máximos valores de ajuste de complejidad se tiene,

$$Funcionalidad = \frac{45.6}{51.3} = 0.8888 = 88\%$$

Por tanto: concluimos que el sistema tiene una funcionalidad aceptable del 88%

Anexo C: Reglas planteadas para el modulo dominio

Existen reglas que se plantearon dentro del modulo del dominio: que son las gramaticales y las de diagnostico, que están relacionadas y se presentan:

Sección <<Organizations>> código JavaScript embebido en HTML

```
<SCRIPT LANGUAGE=JavaScript>
<!-- var InternetExplorer = navigator.appName.indexOf("Microsoft") != -1;
// Handle all the the FSCommand messages in a Flash movie
function quiz_DoFSCommand(command, args) {
  var quizObj = InternetExplorer ? quiz : document.quiz;
  if (command=="start") {
    window.focus();
    var movie = window.document.quiz;
    movie.SetVariable("actuales", act);
  }
}
function openLink(link1,frame1,link2,frame2,link3,frame3) {
  window.open(link1,frame1);
  window.open(link2,frame2);
  window.open(link3,frame3);
}
// gancho a internet explorer
if (navigator.appName && navigator.appName.indexOf("Microsoft") != -1 &&
  navigator.userAgent.indexOf("Windows") != -1 && navigator.userAgent.indexOf("Windows 3.1") ==
-1) {
  document.write('<SCRIPT LANGUAGE=VBScript> \n');
  document.write('on error resume next \n');
  document.write('Sub quiz_FSCommand(ByVal command, ByVal args)\n');
  document.write(' call quiz_DoFSCommand(command, args)\n');
  document.write('end sub\n');
  document.write('</SCR' + 'IPT> \n');
}
//-->
</SCRIPT>
```

```
<<manifest>>..
<organizations default="tutorSQL">
<title>System Tutor intelligent SQL</title>
<item identifier="intro" identifierref="Rintro" isvisible="true">
  <organization identifier> structure="hierarchical">
// desde aqui controla la validacion del agentito:
<SCRIPT LANGUAGE=JavaScript>
  session_start();
  // if (isset($_SESSION['sw']))
  if($_SESSION['sw']==FALSE)
$nom=$_SESSION['nom'];
$pat=$_SESSION['pat'];
$rol=$_SESSION['rol'];
$id=$_SESSION['id'];
/script>
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Documento sin título</title>
<style type="text/css">
<!-- #Layer1 {
      position: absolute;
      left: 209px;
      top: 41px;
      width: 258px;
      height: 153px;
      z-index: 1;
    }
-->
</style>
<script type="text/JavaScript">
<!--
function MM_jumpMenu(targ, selObj, restore){ //v3.0
  eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
  if (restore) selObj.selectedIndex=0;
}
//-->
</script>
</head>
<body>
<div id="Layer1">
  <table width="200" border="1">
    <tr>
      <td>elija evaluacion </td>
    </tr>
    <tr>
      <td><form id="form1" name="form1" method="post" action="valida_eval.php">
        <p><select name="op_eva" id="op_eva" >
          <option>evaluacion1</option>
          <option>evaluacion2</option>
          <option>evaluacion3</option>
          <option>evaluacion4</option>
          <option>evaluacion5</option>
          <option>evaluacion6</option>
        </select>
        </p>
        <p><label> Ingres
          <input type="submit" name="Submit" value="Enviar" />
          </label>
        </p>
      </td>
    </tr>
  </table>
</div><p>&nbsp;</p>
</body>
</script>
```

BIBLIOGRAFIA

[Ausubel, et al, 90] David Ausubel, Joseph Novak, Helen Hanesian: psicología educativa: un punto de vista cognoscitivo. México: Editorial Trillas, SA de CV, 2da edición, 1990

[Arratia-Sarvia, 03] Arratia D. Saravia R. 2003 Tesis de Grado de Informática Tutores Inteligentes para mejorar el entendimiento de las matemáticas La Paz Bolivia Tesis de grado UMSA.

[Booch, 94] Booch, G.; "Object Oriented Analysis and Design with Applications", the Benjamin Cummings Publishing Company, 1994.

[Brj, 97] Booch, G., Rumbaugh, J., Jacobson, I. Unified Modeling Language. Addison Wesley longman, Inc. Massachusetts, USA, 1997.

[Brooks,97] Brooks, R. A.; "A robust layered control system. For a mobile robot", IEEE Journal of Robotics and Automation, Vol. 2, N° 1, pp. 14-23, 1986.

[Buretta,97] Buretta, M.; "Data Replication tools und Techniques, for magazine Distributed information", Editorial Wiley, 19 97.

[Bruner 90] Bruner J. Actos de significado. Más allá de la revolución cognitiva. Alianza. Madrid. 2002.

[Caeiro, M. et al., 2003] Manuel Caeiro, Fernando Mikic, Luis Anido, Martín Llamas: Análisis de componentes para un Modelo de Descripción de Unidades de Aprendizaje Heterogéneas, En: Tercer Congreso Iberoamericano de Telemática-CITA 2003, Uruguay, octubre 2003.

[Carbonell, 70] Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer assisted instruction. IEEE transaction on Man Machine System. Volumen 11 número 4, p. 190-202.

[Cataldi, Z.; Salgueiro, 05], Lage, F. J. y García-Martínez, R. (2005). Sistemas tutores inteligentes: los estilos del Estudiante para selección del tutorizado. Proceedings of WICC 2005: Workshop de Investigadores en Ciencias de la Computación en CD. 13 y 14 de mayo. Universidad Nacional de Río Cuarto. Córdoba. RED UNCI. Costa, G.; Salgueiro, F. A., Cataldi, Z., García Martínez, R. y Lage, F. J. 2005. Sistemas inteligentes para el modelado del estudiante Proc. GCETE'2005, Global Congress on Engineering and Technology Education CD. Marzo 13-15.

[Castillo et al., 98] Sistemas Expertos y modelos de redes probalísticas. Monografías de la academia española de Ingeniería, Madrid.

[Cullas,01] Cullas, V. J.; "Sistema de apoyo a la gestión de un ambiente de replicación de dalos en un .sistema distribuido de bases de datos", Tesis para optar por el Título de Ingeniero Informático, PUCP-PERU.

[Dirichlet, 97] Geiger D., Heckerman D. A. Characterization of the Dirichlet Distribution Through Global and Local Parameter Independence. The Annals of Statistics, Vol. 25, No. 3, 1997, pp. 1344-1369.

[Felder, 88] Felder R.M.; Silverman L.K. (1988). Learning Styles and Teaching Styles in Engineering Education. Engr. Education, Volumen 78, número 7, p. 674-681.

[Gardner, H. 85] Las Inteligencias Múltiples. Paidós, Barcelona.

[Guardia Robles, 93] Asesores inteligentes para apoyar el proceso de enseñanza-aprendizaje de lenguajes de programación. Tesis de grado Assessor ITESM: Institute technologic de Monterrey.

[Hermans, 96] Hermans, B.; "Intelligent Software Agents On The Internet: An Inventory Gf Currently Offered Funcionality In The Information Society Prediction Of (Near) Future Developments", Tilburg University, The Netherlands, 1996.

[Hume, 96] Hume G., Michael, J; Rovick, A.; Evens, M. (1996), Hinting as a tactic in one-on-one tutoring. Journal of Learning Sciences

[Hernandez, 01] Hernández Sampieri C. metodología de la investigación Mc Graw Hil. México 2001.

[Jacobson, 97] Jacobson, L. Christenson, M. Jonson, and Wergaard, G.; "Object Oriented Software Engineering", A Use Case Driven Approach, ACM Press.

[Maturana, 98] Da Biología á Psicología Artmed Editora Ltda.

[Maes, 91] Maes, P.; "Agents that reduce work and information overload", Communication of the ACM, Vol. 37, N°. 7, pp. ~31-40, 1991.

[Microsoft, 01] Microsoft Documentación; "Libros en pantalla de SQL Server", 2001

[Minsky, 94] Minsky, M.; "A Conversation with Marvin Minsky about Agents", Communication of the ACM. Vol. 37, N°. 7, pp 81-91, 1994.

[Oracle, 99] Oracle Documentation; "Oracle9i Database", Release 2 (8.1.6). A76959-01, 1999.

[Oracle 10g, 05] Administración y análisis de base de datos Cesar Pérez 2005 Alfa Omega Group Editor S.A. Biblioteca UMSA.

[Oracle SQL Reference, 05] Database SQL Reference 10g Release 2 (0.2) Part Number B14200-02 June 2005

[Oracle Administrator's Guide, 05] Oracle® Database Administrator's Guide 10g Release 2 (10.2) B14231-01 June 2005

[Paradela, L., 2001] Luis Paradela: Tesis doctoral: Una Metodología para la gestión del conocimiento. España: Universidad Politécnica de Madrid, 2001.

[Pazos, J., 2000] Juan Pazos Sierra: Esquema de Solución de Problemas. España; Publicación Personal, 2002.

[Pazos, J., 2001] Juan Pazos Sierra: Enseñanza del futuro: a grandes males pequeños remedios. España: Universidad Politécnica de Madrid, 2001.

[Perkins, 95] Perkins D. la escuela inteligente. Gedisa.

[Piaget, 89] Piaget J. la construcción de lo real en el niño. Critica. Grijalbo.

[Platon, 01] la República. Ed Tecnos. Reimpresión 2001.

[Pozo, J., 1996] Juan Ignacio Pozo: Aprendices y maestros. Madrid: Alianza Editorial, 1996.

[Real Academia Española, Real Academia Española: Diccionario de la Lengua 2001] Española. España: Real Academia Española, 2001 (vigésima segunda edición).

[Rumbaugh, 91] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenzen, "Object Oriented Modeling and Design", Prentice-Hall, 1991.

[Russell, 95] Russell, S. J. and Norvig P.; "Artificial Intelligence: A Modern Approach", Englewood Cliffs, NJ: Prentice Hall, 1995.

[Seu Jai et al., 91] Ru-Charn Chang, Jun Li; Evens M. Language Differences in Face-to-Face and Keyboard-to-Keyboard, Tutoring Session. Proceedings of the Cognitive Science Society.

[Shoham, 93] Shoham, Y.; "Agent oriented programming", Artificial Intelligence, Vol. 60, N°. 1, pp. 51-92, March 1993.

[**Stuart, 95**] Stuart J.R. Norvig, Canny, J. Artificial Intelligent a Modern Approach. Prentice Hall, Englewood Cliffs, New Jersey 07632.

[**Woold, 95**] Wooldridge, M. and Jennings N. R.; "Agent theories, Architectures, and Languages: a Szrrvey", in Wooldridge and Jennings Eds., Intelligence Agents, Berlin: Springer-Verlag, Vol. 1, N° 22, 1995.

[**Kim, 89**] Kim, J. H. (1989). CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology. Ph.D. tesis, Illinois Institute of Technology.

[**Van Lehn, 88**] Student Modelling. M. Polson. Foundations of intelligent Tutoring System. Hillsdale. N. J. Lawrence Erlbaum Associates, 55-78

[**Vernet, 05**]. Vernet, F.M.T, Dunin-Keplicz B.M. Treur J. and Verbrugge, "lirfodelling Internal Dynamic Behczviour of 13 Agents". Proceedings of the Third International Workshop on Formal models of Notes Springer Verlag, 1997.

[**Vigotsky, 78**] vigotsky L. Mind in society. The development on higher psychological process Harvard University Press Cambridge M. A.

[**Villareal, 03**] Villareal Goulart R. R.; Giraffa Arquitectura de Sistemas tutores Inteligentes.

[**Wang, 98**] Wang K. y Liu B. Concurrent discretization of multiple attributes. Rim Internacional Conference on AI, pp. 250, 259.

[**Wooldrigde, 99**] Artificial Intelligent and tutoring systems 1999.

[**Woo Woo, 91**] Instructional planning in an Intelligent Tutoring System: Combining global lesson plan whit local discourse control. Degree of Doctor of Philosophy in computer Science in the Graduate School of the Illinois Institute of Technology. Chicago, Illinois December, 1991.