



UNIVERSIDAD DE PINAR DEL RÍO

“Hermanos Saíz Montes de Oca”



Facultad de Informática y Telecomunicaciones

En convenio con la Universidad Técnica de Cotopaxi del Ecuador

Tema:

**Sistema Auto-evaluador de la Solidez del Aprendizaje
(SAESA)**

**Tesis presentada en Opción al Título de
Ingeniero en Informática y Sistemas Computacionales**

Autores:

Darwin Fernando Núñez Collantes

Franklin Rolando Vargas Toaquiza

Tutor:

Dr. C. José Quintín Cuador Gil.

CUBA

Pinar del Río, Junio del 2008

Año 50 de la Revolución

Pinar del Río, Junio de 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de esta tesis y autorizamos al Departamento de Informática de la Universidad de Pinar del Río “Hermanos Saíz Montes de Oca” para que hagan el uso que estimen pertinente con el presente trabajo.

Darwin F. Núñez Collantes.

Autor

Franklin R. Vargas Toaquiza

Autor

Dr. C. José Quintín Cuador Gil

Tutor

Q mis amados padres; por estar apoyándome en cada etapa de mi vida; especialmente a mi madre Gladys Collantes quién con amor infinito estuvo brindándome ternura, confianza y apoyo incondicional en mi formación personal y profesional

A la memoria de mi querido y siempre recordado abuelito Manuel Monge quién me brindó cariño, aliento y apoyo sobretodo en los momentos difíciles.

Hoy simplemente les dedico el fruto de sus concejos, de sus anhelos, de sus esfuerzos y de sus sabias enseñanzas. Muchas gracias...

Darwin

Q Dios por concederme la vida y toda la felicidad del mundo lleno de bendiciones.

Q mis queridos padres María Zara Toaquiza y Manuel Vargas quienes supieron enrumbarme por el camino del bien con su amor, ternura y comprensión, gracias por darme el ejemplo de lucha y persistencia para alcanzar metas en la vida.

Q mis queridas hermanas y hermano: Silvia, Rosa, Jenny y Edison con quienes compartí muchos momentos felices en familia y con el anhelo de que sigan preparándose ya que es la mejor herencia de nuestros padres.

Q toda mi familia, mil gracias por haber estado en todas mis metas y esta es una de ellas.

Franklin

Agradecimiento

Infinitamente a Dios por permitirme no solo ver realizado este gran sueño, sino además por rodearme de personas tan maravillosas que me apoyaron en el alcance de esta meta:

A mis queridos padres Luís Núñez y Gladys Collantes quienes supieron enrumbarme por el camino del bien con su amor, comprensión, y enseñanzas gracias por darme el ejemplo de lucha y persistencia para alcanzar metas en la vida.

A mi tierna hija Ninel quien con su dulzura e inocencia fue mi inspiración y ahínco de todos los días.

A Pilar por su amor, confianza y ánimos durante todo este tiempo.

A mis queridas hermanas Diana y Elenita con quienes compartí muchos momentos felices en familia y con el anhelo de que sigan preparándose ya que es la mejor herencia de nuestros padres.

A mi amigo Miguel que me brindó su apoyo para el logro de este objetivo.

A mis compañeros y amigos con quienes compartí mis alegrías y tristezas durante todo este tiempo.

A mi tutor Dr. C. José Quintín por brindarnos sus conocimientos profesionales para el desarrollo de este trabajo de diploma.

A los profesores Msc. Manuel Jesús López, Msc. Mykel Menéndez por su ayuda profesional al desarrollo de este proyecto.

A la Universidad Técnica de Cotopaxi, por haberme permitido crecer intelectual y profesionalmente.

A la Universidad Pinar del Río " Hermanos Saiz Montes de Oca " Cuba, por haberme abierto las puertas para desarrollar mi diplomado.

A todos aquellos que de un modo u otro contribuyeron en la realización de este trabajo...

Muchas Gracias...

Autores: Darwin Fernando Núñez Collantes

Franklin Rolando Vargas Toaquiza

Centro de Trabajo: Universidad de Pinar del Río “Hnos. Saíz Montes de Oca”.

Correo Electrónico: darwinf-@postgrado.upr.edu.cu

franklin@postgrado.upr.edu.cu

Tutor: Dr. C. José Quintín Cuador Gil

Centro de Trabajo: Universidad de Pinar del Río. Dpto. de Informática.

Correo Electrónico: cuador@info.upr.edu.cu

Durante los últimos tiempos la tecnología y la Informática han venido evolucionando en todo el planeta, a tal punto que las tareas manuales cada día se van automatizando, beneficiando así al hombre en sus tareas diarias, logrando obtener rapidez, eficiencia y tecnificación. Siendo esto una ventaja y ayuda para la sociedad, Cuba acoge los distintos avances informáticos aplicándolos en los diversos campos como en el educativo concretamente en la Universidad de Pinar del Río.

La falta de softwares auto-evaluativos de solidez del aprendizaje que no dispone la UPR no permite determinar los niveles de conocimientos de los estudiantes, por lo que fue objetivo de este trabajo el de “obtener una herramienta evaluadora para que los alumnos de años terminales puedan auto evaluar la solidez de su aprendizaje adquirido durante la carrera.”, consultando para ello con una Base de Datos en que se recoja preguntas de: selección única, selección múltiple y de enlace definidas para dos categorías que son perfiles y herramientas académicas, , permitiéndole al administrador editar, incorporar y eliminar la información para así ir adecuándolos a las necesidades.

SAESA se confeccionó con la herramienta Dreamweaver MX, soportado sobre el lenguaje de programación PHP; empleando MySQL Server como gestor de Base de Datos y el CASE Rational Rose, diseñado sobre el Lenguaje Unificado de Modelado (UML) para caracterizar y modelar el sistema, siendo esta capaz de realizar, de manera eficiente, búsquedas categorizadas que posibilitan el acceso inmediato a la información pedida y viabilizando la inserción de manera organizada de nuevos recursos con la aprobación especializada del personal capacitado. El producto obtenido, SAESA contribuye a incrementar la rapidez y certeza de la toma de decisiones del estudiante al rendir su auto-evaluación, además de permitirle conocer su grado de conocimiento adquiridos durante el transcurso de sus estudios anteriores.

Índice

INTRODUCCIÓN	1
CAPÍTULO I.- BASES PRELIMINARES	4
Introducción	4
I.1.-Fundamentos teóricos de la Investigación	5
I.2.- Diseño de la Investigación.....	9
I.3.- Costo aproximado de la herramienta evaluadora.	12
CAPÍTULO II.- DISEÑO DE SAESA	20
Introducción	20
II.1 Herramientas utilizadas en la Modelación.....	21
II.1.1. - Unified Modeling Language (UML).	21
II.1.2.- La Herramienta CASE “Rational Rose”	24
II.2. Diseño de la Base de Datos.	26
II.3.- Diseño de las interfaces de usuario.	28
II.4.- Plan de Seguridad.	30
II.5.- Arquitectura utilizada.	31
II.6.- La Navegación en SAESA.	33
CAPÍTULO III.- IMPLEMENTACIÓN DE SAESA	37
III.1.- Construcción de la Base de Datos.....	38
III.1.1.- Criterios considerados para elegir a MySQL como gestor.	39
III.1.2.- El Modelo Físico de los Datos.....	41
III.2.- Construcción de las Interfaces de Usuario.....	42
II.2.1. - Personal Hypertext Preprocessor (PHP).	43
II.2.2. - Structured Query Language (SQL).....	44
III.2.3. - El editor WEB Dreamweaver.....	45
III.3.- Implementación del Plan la seguridad.....	48
CONCLUSIONES	51
RECOMENDACIONES	52
ANEXOS	53

Índice



INTRODUCCIÓN

El desarrollo acelerado de la informática y de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC), determina que aún en la situación socioeconómica de países en vías de desarrollo, se disponga de recursos tecnológicos para automatizar la sociedad. El comandante Ernesto Che Guevara en época tan temprana como marzo de 1962 planteo “El mundo camina hacia la era electrónica..... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestro esfuerzos en ese sentido con audacia revolucionaria”.

Hoy nos encontramos, si tenemos en cuenta el desarrollo de la ciencia y la tecnología, en una etapa que bien pudiera caracterizarse como una Revolución de la Información denominándose a nuestra sociedad como “Sociedad de la Información”. Las industrias de la cibernética, telecomunicaciones, el desarrollo de la electrónica y de la microelectrónica constituyen, entre otros, elementos claves en las NTIC. Es una era donde se caracteriza como recurso del poder el conocimiento, la información y por tanto requiere de hombres y de mujeres capaces de procesar la información con uso de la computación y de las NTIC, elemento clave de la supervivencia y desarrollo actual y que distingue la actual sociedad.

Desde el año 1959 el estado revolucionario cubano se ha preocupado por la utilización de la computación en la enseñanza, no solo como objeto de estudio introduciéndola en los Planes de Estudio de todos los niveles de enseñanza sino también como herramienta de trabajo durante el proceso de enseñanza- aprendizaje, en las investigaciones, en la gestión docente, en el control del aprendizaje, etc. Hoy día se dispone de un caudal enorme de software educativo (tutoriales, entrenadores, simuladores, etc.), de sistemas de gestión educacional, de software profesionales, de plataformas interactivas de aprendizaje, dirigidas a solventar diversos problemas. Sin embargo en la arista del control del aprendizaje y más aún en poder controlar el alumno la solidez de su aprendizaje adquirido durante su formación como futuro profesional se ha hecho bastante poco. En esta dirección es dirigida la investigación realizada.

Se tomo como objeto de investigación “el control de la solidez del aprendizaje de los alumnos de la Universidad de Pinar del Río” sobre el cual se trabajará para buscar el



Introducción

medio que permita al alumno jugar el papel protagónico en su realización, empleando una base evaluativa creada por el profesor.

El objetivo del trabajo es el de obtener una herramienta evaluadora para que los alumnos de años terminales pueda auto evaluar la solidez de su aprendizaje adquirido durante la carrera.

El documento presentado sirve de hilo conductor de cómo se llevo a cabo la investigación, así se tiene:

Capitulo II.- Diseño de la Herramienta Evaluadora

Comienza con un epígrafe en que se expresan los fundamentos teóricos en que se baso la investigación, tratando conceptos como los de: aprendizaje, enseñanza, proceso enseñanza-aprendizaje, evaluación, control del aprendizaje, etc. En su segundo epígrafe contiene los elementos tenidos en cuenta en el diseño de la investigación, se justifica el tipo de herramienta evaluadora necesaria obtener, el modelo conceptual de la base evaluativa, objetos necesarios manejar en esta de acuerdo a ¿Qué debe ser controlado? Se concluye con un epígrafe donde se realiza un estimado del costo de la herramienta evaluadora.

Capitulo I.- Diseño de la Investigación

En él se comienza con el epígrafe donde es caracterizado el lenguaje de modelado usado para obtener algunas vistas del producto informático, Unified Modeling Language (UML) y de la herramienta CASE “Rational Rose” utilizada para construir los diagramas correspondientes a esas vistas. En su segundo epígrafe se diseña la Base Evaluadora presentando el Diagrama de Clases Persistentes y el Modelo de Datos obtenido a partir de él. En el tercero se diseña las Interfaces de Usuario que serán usadas por la Herramienta evaluadora, en correspondencia con las funcionalidades que debe tener, se ofrece la Vista de Caso de Uso del producto a través de los Diagramas de Caso de Uso. En el cuarto epígrafe se expone el Plan de Seguridad el cual tiene en cuenta los roles jugados por cada actor ante la herramienta. En el quinto epígrafe se refuerza la validez de la arquitectura seleccionada y se presenta en el Diagrama de Clases Cliente/Servidor como estas estarán distribuidas según la arquitectura tomada. Concluye el capitulo con el epígrafe que muestra como cada actor navegará en la herramienta evaluadora.



Capítulo III.- Construcción de la Herramienta Evaluadora

EL capítulo comienza con el epígrafe de la construcción de la Base Evaluadora, realizándose de antemano una valoración por que se tomo como gestor MYSQL, terminando con el Modelo Físico de los Datos obtenido con este gestor. En su segundo epígrafe se valora las razones de haber tomado a Dreaweaver como editor WEB y el PHP como lenguaje de programación, concluyendo con la ejemplificación de algunas de las ventajas usadas de estas tecnologías WEB.

El método fundamental que se toma para aglutinar todos los métodos empleados fue el dialéctico materialista, puesto que se trabajó sobre la base de sus categorías básicas. Este se manifiesta en diversos momentos, así, al utilizar el método genético se partió de la célula fundamental que eligió nuestra investigación: el problema. En el mismo, se parte de la contradicción que hay entre lo conocido y lo desconocido para mejorar el proceso de control final del aprendizaje.

Por otra parte, para dar solución a la contradicción y constatar su desarrollo fue utilizado el método histórico-lógico, el cual permitió: primero, relevar la historia del uso de la computación para realizar el control final del aprendizaje en específico se su solidez, que constituyó el centro de la investigación; después, de desarrollar de una manera lógica y deductiva los principios básicos y esenciales encontrados.

Para poder desarrollar la investigación se necesitó establecer una exploración que nos sirvió para planificar el sistema de tareas, y los modelos más adecuados para representar la solución al problema con la creación de una herramienta evaluadora automatizada. Estas tareas constituyeron un sistema coherente que posibilitaron el método de ascenso de lo abstracto a lo concreto (método genético).

Es evidente que todos los métodos teóricos mencionados conformaron un sistema que respondieron al objetivo de la investigación.

Con relación a los métodos empíricos debemos plantear el empleo del análisis documental que fue realizado con el propósito de constatar el estado en que se encuentra del uso de la computación para controlar de la solidez del aprendizaje. El criterio de experto por su parte nos permitió valorar la viabilidad de la implementación de la herramienta evaluadora obtenida.



CAPÍTULO I.- BASES PRELIMINARES

Introducción

El capítulo presenta en su primer epígrafe los fundamentos teóricos en que se baso la investigación, se puntualizan los conceptos de: aprendizaje, enseñanza, proceso enseñanza-aprendizaje, evaluación, control del aprendizaje, autocontrol mediante los criterios de varios autores. Se hace énfasis en el control del aprendizaje presentando los tipos de controles y sus funciones y respondiendo la pregunta, ¿cuándo controlar?

En el segundo epígrafe del capítulo se presenta el diseño de la investigación comenzando con la valoración de la necesidad de una herramienta que permita a los alumnos realizar la auto evaluación de la solidez del aprendizaje adquirido durante el estudio de la carrera. Es señalado la necesidad que esta herramienta sea automatizada y construida en un ambiente WEB, presentando los objetos empleados para realizarla mediante su Modelo Conceptual. Se señala que para poder obtener un control válido es necesario garantizar la calidad de la base de datos, recomendando algunos de los aspectos a tener presente por los expertos para lograrlo.

El capítulo concluye con la estimación del costo de construir la herramienta de autocontrol, estimación realizada con el Modelo de Diseño Temprano de COCOMO II, usando como métrica la de los puntos de función.

I.1.-Fundamentos teóricos de la Investigación

Es bien conocida la necesidad de elaborar fundamentos teóricos que justifiquen, guíen y provean significados para desarrollar futuras prácticas. Las teorías marcan los núcleos fundamentales, revelan nuevas formas de conocer y sugieren alternativas. La educación debe basarse en postulados teóricos sólidos, coherentes y rigurosos.

Teniendo en cuenta la problemática antes planteadas y los objetivos propuestos, nos dimos la tarea de analizar los criterios de algunos autores referentes a los aspectos más significativos de la investigación, por ejemplo: enseñanza, aprendizaje, proceso de enseñanza–aprendizaje, evaluación, control del aprendizaje, y autoevaluación.

Ha sido interés en diferentes investigaciones, el perfeccionar el proceso enseñanza aprendizaje en la escuela actual y centros educacionales en sentido general. Es rico el contenido encontrado referente a este concepto y enorme la gama de autores, de las diferentes tendencias pedagógicas, que se han dedicado a la conceptualización del mismo, por ser un concepto medular de la pedagogía.

Particularmente nos interesa analizar este concepto desde la perspectiva del enfoque histórico–cultural, que es el defendido en este trabajo. Para ello, centraremos nuestra atención en tres conceptos: enseñanza, aprendizaje y enseñanza–aprendizaje. Comenzaremos con los criterios del Dr. José Zilberstein Toruncha, que al respecto plantea:

- **Enseñanza:** es el proceso de organización de la actividad cognoscitiva de los escolares, que implica la apropiación por estos de la experiencia histórico–social y la asimilación de la imagen ideal de los objetos, su reflejo o reproducción espiritual, lo que mediatiza toda su actividad y contribuye a su socialización y formación de valores (Zilberstein J., 2000)¹.
- **Aprendizaje:** es un proceso en el que participa activamente el alumno, dirigido por el docente, apropiándose, el primero, de conocimientos, habilidades y capacidades, en comunicación con los otros, en un proceso de socialización que favorece la formación de valores, "es la actividad de asimilación de un

¹ Zilberstein Toruncha José y Silvestre Oramas Margarita. ¿Cómo Hacer más Eficiente el Aprendizaje? Editorial Pueblo y Educación, 2000.

proceso especialmente organizado con ese fin, la enseñanza." (Zilberstein, J, 2000 (1)).

- **Proceso enseñanza–aprendizaje:** es el proceso que está regido por leyes concatenadas (pedagógicas, psicológicas, lógicas, filosóficas, entre otras), que interactúan y se condicionan mutuamente. Estas leyes deben conocerse por los docentes, a los efectos que este se desarrolle como un sistema (Zilberstein, J, 2000 (1)).

Por su parte el Dr. Carlos M. Álvarez de Sayas₂ analiza estos conceptos tomando como concepto superior el **Proceso Docente–Educativo**, analicemos su punto de vista:

- **Proceso Docente-Educativo:** es el proceso mediante el cual se debe alcanzar el objetivo cuando el estudiante se apropia del contenido. El Proceso Docente-Educativo se caracteriza estructuralmente por un conjunto de componentes: problema, objeto, objetivo, contenido, método, medio, forma y evaluación.
- **Enseñanza:** es el proceso donde se guía a la persona para transformar la información.
- **Aprendizaje:** actividad del estudiante para instruirse.
- **Proceso de enseñanza–aprendizaje:** es la actividad en sí del proceso Docente-educativo.

Según estas definiciones de Carlos Álvarez, se establece entre estos conceptos la relación siguiente: La visión de J. Zilberstein es mucho más abierta que la de C. Álvarez pues este último autor restringe la enseñanza y el aprendizaje a la instrucción aunque cuando define proceso de enseñanza–aprendizaje considera las dimensiones: instructiva, educativa y desarrolladora. Por su parte Zilberstein de forma integral armoniza estas tres dimensiones en los conceptos objetos de análisis, es por ello que en este trabajo tomamos partido por J. Zilberstein.

La evaluación.

Por su parte María Eugenia Garriga₃ analiza el concepto de evaluación como:

“La estimación, permanente y sistemática del proceso de aprendizaje, producido en la mente de los evaluados, en sus capacidades, y en sus conductas, con el fin de orientarlo y reorientarlo hasta que lleguen a resolver problemas individuales y sociales, con incidencia además, en el proceso de educación general, para

₂ Álvarez de Sayas Carlos. Hacia una escuela por excelencia. Editorial Academia.1996.

₃ Garriga María Eugenia. Propuesta de un Sistema de Evaluación Alternativa. 2008



favorecer el aprendizaje mediante la toma de decisiones y la intervención oportuna de los alumnos del grupo y su profesor”.

María Eugenia Garriga es su trabajo señala que la evaluación del aprendizaje forma parte del perfeccionamiento del proceso enseñanza–aprendizaje, aunque en el interior de las aulas en ocasiones se mantienen las tendencias tradicionales establecidas a través de la historia. Como es el hecho de la no-coparticipación de los sujetos (alumnos y profesores) en el diseño y elaboración de la evaluación, las funciones de los alumnos están centradas en reproducir y la de los profesores en medir los objetivos pedagógicos. El Centro de Estudios de la Didáctica de la Educación Superior (CEDES) en la Universidad de Pinar del Río, en el cual se realizó el trabajo presentado, hace un llamado para descubrir nuevas vías de evaluación que contribuyan al perfeccionamiento del proceso docente-educativo en esta entidad⁴.

En el ámbito central, el que fue Ministro de Educación Superior en Cuba, Dr. Fernando Vecino Alegret, se cuestionó las concepciones y prácticas vigentes en las universidades respecto a la evaluación del aprendizaje (3)⁵. Otros especialistas también reclaman una actitud de investigación, tanto de los profesores, como de los futuros profesionales en todas las Ciencias. Es por esto, que toda labor educativa realizada en las aulas, puede constituir un soporte conceptual pertinente, útil para el mejoramiento didáctico.

La tarea actual de los docentes como respuesta a esta situación está en crear alternativas, con el supuesto de variar los sistemas de evaluación para que potencien el desarrollo general del alumno e incidan favorablemente en el aprendizaje cualitativo.

Control del aprendizaje.

La académica Dra. Nina F. Talízina⁶ en sus conferencias referente al tema del Control y sus funciones docentes señala aspectos con los que coincidimos como:

“...si no garantizamos este control no podemos hablar en ningún momento de dirección del proceso de asimilación porque todas las dificultades se quedan sin detectar y la ayuda necesaria le ejecutamos inoportunamente”.

⁴ Notario, Ángel. Conferencia Docente-Metodológica. Universidad de Pinar del Río, 1998

⁵ Dr. Fernando Vecino Alegret. Prólogo del libro “Evaluación de los aprendizajes” Miram González 2000:3 En el se señala la necesidad de desarrollar investigaciones que favorezcan los procesos educativos en los centros de Educación Superior en Cuba.

⁶ Talízina Nina. Conferencias sobre “Los Problemas Fundamentales en la Enseñanza en la Educación Superior”, Departamento de Estudios para el Perfeccionamiento de la Educación Superior, 1984.

Los tipos de controles existentes y sus funciones se resumen en la figura I.1.1

Control Preliminar (Función: Comprobar el nivel de preparación y motivacional)

- Comprobar el nivel de conocimiento (Exámenes de ingreso)
- Comprobar el nivel de motivación
- Comprobar el nivel de conocimiento para estudiar un nuevo contenido (evaluaciones parciales)

Control Sistemático (Funciones: Retroalimentación, Motivación, Apoyo, Reforzamiento, etc.)

- Comprobar la comprensión del contenido (retroalimentación)
- Lograr la motivación
- Brindar ayuda (reforzamiento de los contenidos)

Control Final (Función de Registro)

- Comprobar el nivel alcanzado en la enseñanza y los objetivos planteados

Figura I.1.1.- Tipos de Controles.

Teniendo en cuenta el objetivo de nuestra investigación nos centraremos en el Control Final siendo su fin el de comprobar el nivel del aprendizaje con los objetivos planteados alcanzar. Talízina(4) plantea: “.....cuando analizamos el control final, tenemos que concentrarnos en su función de calificación. Para que el control cumpla con esta función debe responder a dos requisitos: **ser válido y seguro.**”

¿Cuándo podemos asegurar que el control es válido?

Al respecto se señala por Talízina que para poder determinar la validez del Control Final es necesario tener en cuenta no solo la evaluación de un conocimiento o una habilidad, sino también las cualidades con que han sido formados, y que están previstos en los objetivos. Pueden estas ser el grado de generalización, el grado de automatización, la rapidez, la independencia, el **grado de solidez**, entre otras. En el caso específico de nuestra investigación la cualidad de interés a evaluar en el Control Final es el grado de solidez de los conocimientos adquiridos, por lo que es necesario que sea realizado no concluida la enseñanza de esos conocimientos, sino al pasar cierto tiempo, solo así este control sería válido.

¿Qué conocimientos y habilidades evaluar?

Talízina planteaba: “.....el control tradicional no es válido. Cuando escuchamos al alumno contestar dos o tres preguntas concluimos que con esta calidad ha asimilado

toda la asignatura, pero no existe fundamento alguno para esta conclusión. Por eso el primer problema que enfrenta la práctica del control, es elevar el grado de la validez de contenido”. En el caso específico de nuestra problemática serán los contenidos y habilidades (aquellas posibles evaluar por el tipo medio) exigidas en el Modelo del profesional.

La autoevaluación (autocontrol).

Citando a Talízina(4) “.....a medida que se va asimilando la actividad y por consiguiente a medida que se vaya asimilando el control, el alumno logra prescindir del **control externo**, y puede ejercer un **autocontrol** internamente.” En la práctica tradicional de nuestra enseñanza vemos que el alumno no se auto controla a pesar de existir los instrumentos que posibilitan al profesor brindarlo.”

¿Cuándo es necesario el control?

Talízina(4) respondió la pregunta diciendo “.....la frecuencia del control va a estar en dependencia de dos indicadores: en primer lugar del éxito que se logre en la ejecución de la tarea y segundo de la necesidad que tenga el alumno de ser sometido al control. Como resultado de la combinación de estos índices, pueden darse cuatro variantes: 1) el alumno trabaja con éxito pero siente la necesidad del control porque no está seguro de sí mismo; 2) el alumno comete errores y tiene la necesidad de controlarse; 3) el alumno realiza el trabajo con éxito, además se siente seguro de su éxito y no hay por tanto la necesidad de controlarlo; 4) el alumno realiza el trabajo de forma errónea y a la vez esta seguro que todo está mal, por tanto, tiene necesidad del control”. [Talízina (4)].

A modo de resumen, debemos dar la posibilidad del auto control en tres casos:

- Cuando no se haya logrado el éxito en la actividad independiente que se sienta o no la necesidad del control.
- Cuando se ha logrado el éxito en la actividad pero se sienta la necesidad del control.
- No se sienta la necesidad pero el control es imprescindible

I.2.- Diseño de la Investigación

Actualmente existen plataformas interactivas de aprendizaje como la usada en la Universidad de Pinar del Río, Moodle, que propician la apropiación “independiente” del conocimiento por el alumno ayudando al profesor realizar sus roles de organizador,

consultor y controlador durante el proceso de enseñanza-aprendizaje, propiciando debates, intercambio entre el profesor y alumno, la participación grupal, y el autocontrol, etc., en la asignatura Sin embargo también es necesario comprobar **la solidez de los conocimientos adquiridos** sobre la base del modelo de actuación del profesional, por ejemplo del uso del idioma extranjero, de la computación y las Nuevas Tecnologías para la Comunicación (NTC), de la actuación como profesionales de un perfil en exámenes integradores de años terminales y estatales, etc., necesitando en estos casos de disponer de un mecanismo auto evaluador que además de brindarle el resultado al alumno de la evaluación le ofrezca como resolver las dificultades. Esto hizo que nos planteáramos, sin descartar los demás medios usados para el control del aprendizaje, brindar otro medio dirigido este a realizar la autoevaluación de la solidez de los conocimientos adquiridos durante el estudio de la carrera universitaria. El tipo de control a realizar por la herramienta será Final pero con la característica de evaluar la solidez del aprendizaje.

La herramienta propuesta, teniendo en cuenta a quien se dirige (alumnos de años superiores de la carrera) y en qué momento será empleada (fuera de la actividad docente planificada), obliga a que sea automatizada, así flexibiliza la realización del autocontrol y en un ambiente WEB para hacerlo más accesible.

En la actividad de auto evaluar la solidez del aprendizaje son empleados los conceptos mostrados en el Modelo Conceptual, figura 1.2.1, donde aparece la relaciones entre estos conceptos.

- **Situación Problemática.-** Problema de la profesión necesario dar solución con los conocimientos adquiridos del cual se conoce: su nombre, descripción, y el perfil o herramienta profesional a que pertenece.
- **Perfil Profesional.-** Los correspondientes a las carreras estudiadas, siendo de interés su nombre y el Modelo de Actuación (los conocimientos y habilidades) del profesional.
- **Herramienta Profesional.-** Son aquellas que auxilian al profesional en el desempeño de sus funciones., conociendo de ellas su nombre y caracterización.
- **Preguntas.-** De ellas se conoce su tipo (Enlace, Selección única, selección múltiple), cantidad de posibilidades de selección o de enlace, respuestas correctas de selección o enlace.

- **Reforzamiento.-** Explicación de las actividades a realizar para poder dar respuesta adecuada a la pregunta.
- **Preguntas de Selección Múltiple.-** Son preguntas no excluyentes usadas al dar una respuesta, conociendo su texto y si participa en o no en la respuesta.
- **Preguntas de Selección Única.-** Son preguntas excluyentes usada para dar respuesta, conociendo su texto y si es o no la respuesta.
- **Preguntas de enlace.-** Son preguntas que son necesarias enlazar con otra al dar la respuesta, conociendo de ellas su texto y la pregunta con la cual se enlaza.

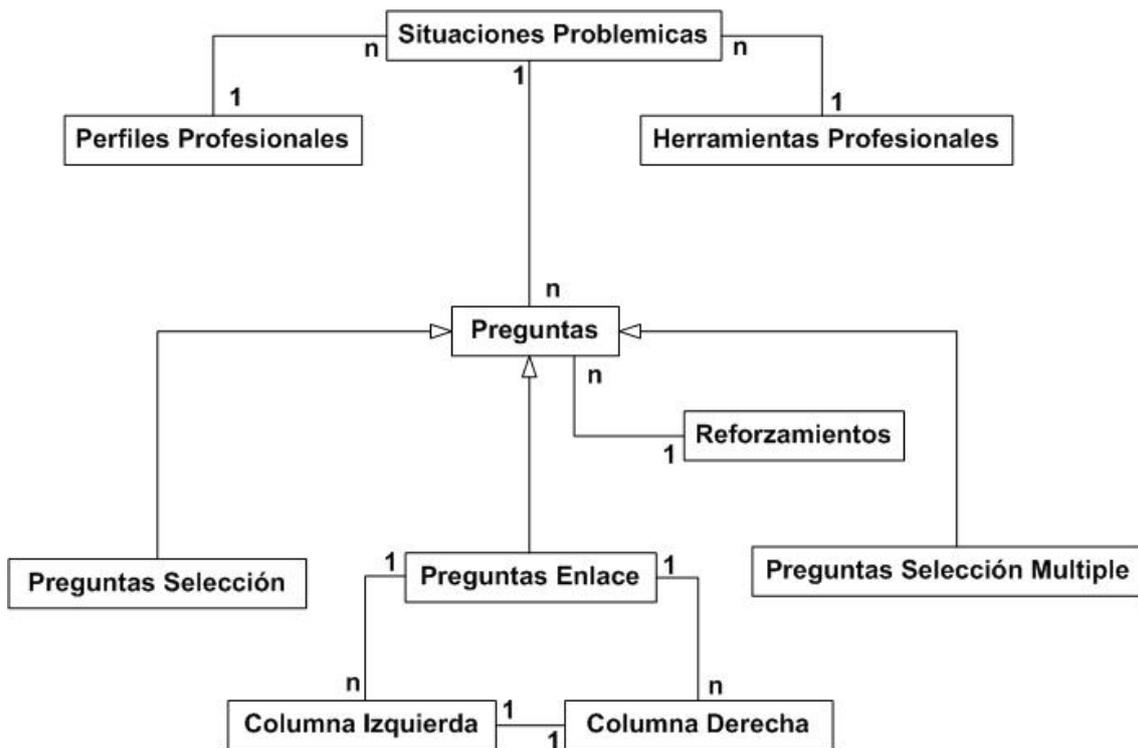


Figura I.2.1.- Modelo Conceptual

Recomendaciones a tener presente al construir la base de evaluación

El medio propuesto construir para autoevaluar la solidez del aprendizaje de los alumnos resulta efectivo en la medida que el control realizado sea válido, pero esta validez depende esencialmente de la calidad de las preguntas incorporadas, Talízina(4) puntualizaba en su conferencias que no todas las PC son capaces de evaluar o percibir una respuesta construida por el alumno, por ello se usa con mayor frecuencia el método de selección de respuesta, advirtiéndolo los peligros que este método entraña, en este caso el alumno no elabora la respuesta, sino que tiene que localizarla dentro de un sistema de respuesta, reconocer es mucho más fácil que elaborar diciendo: “.....la elaboración de un sistema de respuesta de selección exige el cumplimiento de una serie de condiciones. Esto significa que este método no siempre sea tan fácil de utilizar”, agregando a ello que no siempre contamos con la preparación adecuada y experiencia del docente.

Como puede apreciarse en el Modelo presentado se eligió el método de selección de respuesta. Al construir la base de evaluación es necesario que el experto tenga en cuenta las recomendaciones siguientes:

- Toda pregunta debe dar respuesta a una situación problemática a resolver del perfil profesional.
- En las preguntas de selección la ubicación de las respuestas correctas debe ser aleatoria, evitando con ello que aparezcan situadas en un lugar fijo.
- Para decidir cuál de las respuestas es correcta, el alumno tenga que realizar una acción que le permita llegar a una conclusión.
- Tener en cuenta las invariantes, así se evaluará lo esencial.

I.3.- Costo aproximado de la herramienta evaluadora.

Se realizó previamente una búsqueda de herramientas evaluadoras afines a la que se deseaba construir no encontrándose ninguna capaz de dar solución a la problemática. Esto provocó la necesidad de diseñar e implementar la herramienta evaluadora.

Antes de llevar a cabo la tarea de construir la nueva herramienta se realizó una valoración aproximada de su costo y tiempo de desarrollo con uso del Modelo de Diseño Temprano de COCOMO II (**C**onstructive **C**ost **M**odel), usando como métrica la de los puntos de función y realizando los cálculos del Modelo mediante el software USC-COCOMO II desarrollado por University of Southern California Center of Software

Engineering. Se comparó el costo obtenido con los beneficios que se obtendrían con el empleo herramienta decidiéndose acometer la tarea de su construcción.

Al aplicar el modelo utilizando como métrica la de los puntos de función, se determino el número de líneas de código necesarias para implementar un punto de función (entrada, salida o petición) usando la tabla de Reconciliación de las Métricas, en la cual aparece por lenguaje de programación el número promedio de líneas código utilizadas para implementar un punto de función. [BOH 2000]⁷. La creación de los puntos de función se realizó usando como entorno de desarrollo Dreamweaver y como lenguajes de Programación PHP y SQL Los Puntos de Función considerandos fueron:

- **Entradas externas (EI):** Entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.

Nombre	Cantidad de Ficheros	Cantidad de Elementos de Datos	Complejidad
Gestionar Usuarios	2	3	Baja
Gestionar Perfiles	1	2	Baja
Gestionar Situación Problemica	3	3	Baja
Gestionar Herramientas	1	2	Baja
Gestionar Refuerzo	1	3	Baja
Gestionar Preguntas	3	6	Baja
Gestionar Preguntas de Enlace	2	3	Baja

Tabla I.3.1 - Entradas Externas.

- **Salidas externas (EO):** Salida que proporciona al usuario información orientada de la aplicación, informes, pantallas, mensajes de error, etc.

Nombre	Cantidad de Ficheros	Cantidad de Elementos de Datos	Complejidad
Pregunta	3	4	Baja
Refuerzo	1	1	Baja
Solución	3	4	Baja

Tabla I.3.2 - Salidas Externas

- **Peticiones externas (EQ):** Es una entrada interactiva que resulta de la generación de un tipo de respuesta en forma de salida interactiva.

⁷ Bohem, B.W. "SW Const Estimation with COCOMO II", Prentice Hall, 2000.

Nombre	Cantidad de Ficheros	Cantidad de Elementos de Datos	Complejidad
Autenticación	1	3	Baja

Tabla I.3.3 - Peticiones Externas

- **Ficheros lógicos internos (ILF):** Archivo, maestro lógico, agrupación lógica de datos que puede ser una parte de una gran base de datos.

Nombre de Tabla	Cantidad de Registros	Cantidad de Elementos de Datos	Complejidad
Usuarios	50	3	Baja
Perfiles	10	2	Baja
Herramientas	20	2	Baja
Situaciones	100	3	Baja
Preguntas	500	6	Baja
Preguntas Enlace	300	3	Baja
Refuerzos	500	6	Baja

Tabla I.3.4 - Ficheros Lógicos Internos

Según los datos anteriores y utilizando “USC-COCOMO II” se obtuvo: **1,870 LDC** (líneas de código) para poder construir **85 PF** (puntos de función), usando el lenguaje USR1 (PHP, SQL)) como muestra la Figura I.3.1.

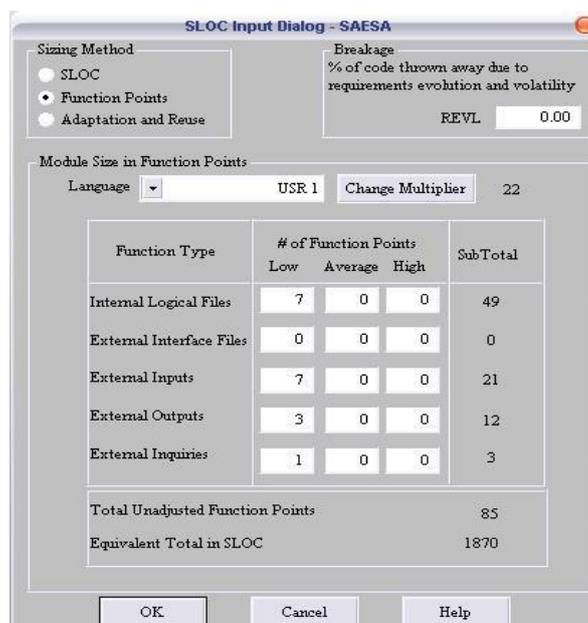


Figura I.3.1 - Puntos de Función y Líneas de Código

Los valores considerados de los Multiplicadores del Esfuerzo (EM) para el Modelo de Diseño Temprano fueron:

Factor	Valor	Justificación
RCPX	Normal	Base de Datos con Normal grado de complejidad.
RUSE	Bajo	El nivel de reutilizabilidad es nulo.
PDIF	Bajo	El tiempo y la memoria estimada son de baja complejidad.
PERS	Normal	La capacidad del personal de desarrollo es normal.
PREX	Normal	Los especialistas tienen experiencia normal en el uso de las tecnologías empleadas.
FCIL	Alta	Se utilizaron herramientas de alto nivel de desarrollo como el CASE Rational Rose
SCED	Normal	Los requerimientos de cumplimiento de cronograma son normales.

Tabla I.3.5 - Valores de los Multiplicadores del Esfuerzo

Estos datos fueron introducidos en la herramienta “USC-COCOMO II”, Figura I.3.2

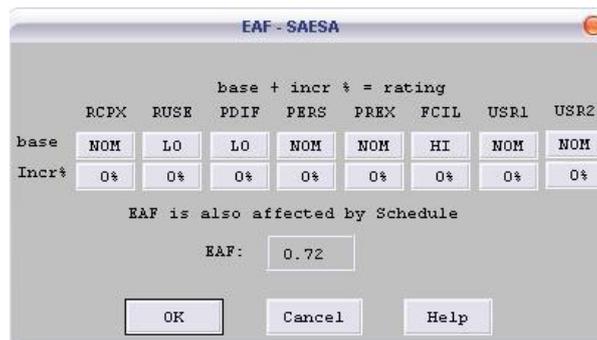


Figura I.3.2 - Valores de Multiplicadores de Esfuerzo.

Los valores considerados de los Factores de escala (SF) fueron:

Factor	Valor	Justificación
PREC	Normal	A pesar de no tenerse experiencia en la realización de software de este tipo este no requiere de aspectos muy novedosos.
FLEX	Normal	La flexibilidad en cuanto a los requerimientos exigidos es normal
RESL	Muy Alto	Existen posibilidades de resolver la mayoría de los riesgos que impone la plataforma.
TEAM	Normal	Existe una cooperación Normal entre los miembros del Team.
PMAT	Muy Bajo	Se encuentra en el nivel 1.

Tabla I.3.6- Valores de los Factores de Escala

A continuación se ilustran la entrada de estos valores en la herramienta “USC-COCOMO II”, Figura I.3.3.

Figura I.3.3 - Factores de Escala

Se asumió como salario promedio \$200 mensuales obteniéndose los resultados mostrados en la Figura I.3.4.

Project Name: SAESA

Scale Factor Schedule

Development Model: Early Design

X	Module Name	Module Size	LABOR Rate (\$/month)	ERF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	SAESA	F:1870	200.00	0.72	USR 1	5.9	4.2	443.6	843.09	0.5	0.7	0.0

Total Lines of Code:	1870	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
		Optimistic	2.8	5.1	662.1	564.87	0.3	0.6	
		Most Likely	4.2	5.8	443.6	843.09	0.5	0.7	0.0
		Pessimistic	6.3	6.6	295.7	1264.63	0.7	1.0	

Figura I.3.4 - Estimación del Costo de “SAESA”

De dónde se obtiene:

Estimados	Esfuerzo (DM)	Tiempo (TDev)	Costo
Optimista	2.0	5.1	564.87
Valor Esperado	4.2	5.0	843.09
Pesimista	6.3	6.6	1,264.53

Tabla I.3.7 - Resultados Parciales



CAPÍTULO I

El valor de cada indicador se obtuvo mediante una media ponderada de los valores dados: **[Valor Optimista + 4 X (Valor Esperado) + Valor Pesimista] / 6**, obteniendo así:

Esfuerzo (DM):

$$DM = (2.0 + 4 * 4.2 + 6.3) / 6 = \mathbf{4.18 \text{ Hombres/Mes}}$$

Tiempo (TDev):

$$TDev = (5.1 + 4 * 5.0 + 6.6) / 6 = \mathbf{5.28 \text{ Meses}}$$

Cantidad de hombres (CH):

$$CH = DM / TDev$$

$$CH = 4.18 / 5.28$$

$$CH = \mathbf{0.79 \text{ Hombres}}$$

Costo de la Fuerza de Trabajo (CFT):

$$CFT = (564.87 + 4 * 843.09 + 1,264.53) / 6 = \mathbf{\$ 866.96}$$

Agregándole a este el Costo de los Medios Técnicos, compuesto este por los costos de depreciación, de mantenimiento y de gasto de energía, y el Costo en Gasto en Materiales.

Costo de los Medios Técnicos (CMT):

$$CMT = CDEP + CMTO + CE$$

Donde:

CDEP: Costo por depreciación (no fue considerado)

CMTO: Costo de mantenimiento de equipo (no fue considerado)

CE: Costo por concepto de energía

$$CE = HTM * CTE * CKW$$

Donde:

HTM: Horas de tiempo de máquina necesarias para el proyecto (700 horas)

CTE: Consumo total de energía (0.608 Kw/h (Estimado))

CKW: Costo Kw/h (\$0.09 hasta 100, \$ 0.20 de 101 a 300, \$0.30 más de 300)

$$KW = HTM * CEN = 700 * 0.608 = \mathbf{425.6}$$

$$CKW = (100 * 0.09) + (200 * 0.20) + (125.6 * 0.30) = \mathbf{\$ 86.68}$$



$$CE= \$ 86.68$$

$$CMT= 0 + 0 \quad 86.68$$

$$CMT= \$ 86.68$$

Cálculo del Costo de Materiales (CMAT):

El costo de los materiales se consideró el 5 % de los costos de los medios técnicos.

$$CMAT= 0.05 * CMT$$

$$CMAT= 0.05 * 86.68$$

$$CMAT= \$ 4.33$$

Cálculos de los Costos Directos (CD):

$$CD= CFT + CMT + CMAT$$

$$CD= 866.96 + 86.68 + 4.33$$

$$CD= \$ 957.97$$

Costo Total del Proyecto (CTP):

$$CTP= CD + 0.1 * CFT$$

$$CTP = 957.97 + 86.69$$

$$CTP = 1,044.66$$

Recursos Humanos:

Tres personas para el análisis, diseño y desarrollo del sistema:

- Tutor: Dr. José Quintín Cuador Gil
- Autores: Darwin Fernando Núñez Collantes
Franklin Rolando Vargas Toaquiza

Recursos Técnicos:

Hardware	Software
Procesador Intel(R) Core 2 Duo (TM) 1.8 Ghz	Sistema Operativo Windows Xp
Memoria: 512 Mb	MySQL.
Disco Duro: 80 Ghz	USC-Cocomo II
Unidad de Respaldo: CD- ROM	PHP-Dreamweaver
Monitor: Res SVGA (1024 X 768) píxeles	Rational Rose



CAPÍTULO I

Teniendo en cuenta que la herramienta es producto de un trabajo de diploma, constituye un ahorro para la Universidad de Pinar del Río al no tener que rembolsar su costo. Valorando los beneficios que la introducción del producto les proporcionaría, los cuales se señalan más adelante se toma la decisión de construirlo.

Beneficios Tangibles:

- Permite realizar a los alumnos de años terminales de las carreras un autocontrol de la solidez de su aprendizaje.

Beneficios Intangibles:

- Aumentar en el alumno la autoestima y seguridad de poder enfrentar las tareas relacionadas con su profesión

Es factible el empleo de la herramienta para auto controlar por el alumno su estado del aprendizaje en una asignatura, solo basta con cambiar la base evaluativa, aunque para este caso recomendamos el uso de la plataforma Moodle por permitir más flexibilidad al profesor para confeccionar los temarios.

CAPÍTULO II.- DISEÑO DE SAESA

Introducción

En este capítulo se ilustra el diseño de la Base de Datos, de las Interfaces de Usuario, de la seguridad, y de la navegación del herramienta propuesta. Este diseño se realizó sobre la base de un lenguaje que permitió construir diferentes vistas del producto (compuestas por diagramas) ayudando a establecer el diálogo Cliente-Desarrollador, usándose el CASE “Rational Rose” para construir los diagramas utilizados, en el primer epígrafe del capítulo son caracterizados el lenguaje y la herramienta CASE.

En su segundo epígrafe se analiza el diseño de la Base de Datos. Se exponen los conceptos (entidad, atributo, relación, llave primaria, llave extranjera, etc.) de Bases de Datos Relacionales utilizados en el Modelo de Datos obtenido a partir del Diagrama de Clases Persistentes.

El tercer epígrafe comienza presentando la captura de los requerimientos funcionales a cumplir por la herramienta evaluadora. Se hace uso de artefactos del Lenguaje Unificado de Modelado (UML, siglas del inglés) como actores, Casos de Usos y Diagramas de Casos de Uso para expresar de forma gráfica y más comprensible estas funcionalidades, se muestra el diseño de las Interfaces-Usuarios de los Caso de Uso principales de la herramienta, que permite brindan los actores de la herramienta las funcionalidades.

El cuarto epígrafe se refiere a como se concibió el Plan de seguridad de la Herramienta evaluadora, teniendo en cuenta los roles de los usuarios que la explotaran.

El quinto epígrafe del capítulo justifica la arquitectura empleada para desarrollar el producto, partiendo de cómo este será explotado por sus usuarios, mostrando el diagrama de clases de la arquitectura usada.

Al tratar el diseño de la navegación del producto, en su sexto epígrafe, se exponen como se tuvo en cuenta su correspondencia con los Casos de Uso (funcionalidades) previamente identificados.

II.1 Herramientas utilizadas en la Modelación.

Para poder realizar el diseño de un producto es necesario saber a quien se dirige y las necesidades que este tendrá del mismo, esto se logra en el diálogo entre el productor (en nuestro caso desarrollador) y el cliente. Esta comunicación se facilita al ser expresada de forma gráfica, mediante diagramas que muestren diferentes vistas del producto a construir (software). El lenguaje usado para construir las vistas fue el Lenguaje Unificado de Modelado (UML, siglas del inglés) y la herramienta empleada para crear los diagramas que componen esas vistas fue la herramienta CASE "Rational Rose", ambos se caracterizan a continuación.

II.1.1. - Unified Modeling Language (UML).

Tomando como base lo expresado en **OMG**⁸, **RUP**⁹, **REF MANUAL**¹⁰ y por **Schmuller**¹¹ se tiene que este lenguaje nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores, los padres de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero de 1997, este lenguaje ha sido utilizado con éxito en sistemas construidos para toda clase de industrias alrededor del mundo: hospitales, bancos, comunicaciones, aeronáutica, finanzas, etc.

El modelado sirve no solamente para los grandes sistemas, aún en aplicaciones de pequeño tamaño, como la muestra, se obtienen beneficios del modelado para ganar en comprensión, sin embargo es un hecho que entre más grande y más complejo es el sistema, más importante es el papel que juega el modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad".

Los principales beneficios de UML, según las referencias citadas son:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (no sólo de software) utilizando conceptos orientados a objetos.

⁸ Object Management Group, OMG Unified Modeling Language Specification V1.5, 2003.

⁹ Jacobson Ivan, Boock Grady, Rumbaugh James, El Proceso Unificado de Desarrollo de Software, Addison Wesley, Madrid 2000.

¹⁰ Jacobson Ivan, Boock Grady, Rumbaugh James, The Unified Modeling Language, Reference Manual, Addison Wesley, 1999.

¹¹ Schmuller Joseph, Aprendiendo UML en 24 horas, Prentice Hall.

- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Usar un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control del proyecto.
- Alta reutilización y minimización de costos.

En estas referencias se destaca que UML es un lenguaje para hacer modelos y es independiente de la metodología seguida de análisis y diseño para construir el software. Existen diferencias importantes entre un método y un lenguaje de modelado. Un *método* es una manera explícita de estructurar el pensamiento y las acciones de cada individuo, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que un lenguaje de modelado carece de estas instrucciones. El lenguaje contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Por las ventajas antes referidas al diseñar la Base de Datos e Interfaces de Usuario de SAESA se emplearon algunos de los artefactos del Lenguaje de Modelado Unificado, construidos con la herramienta CASE (**C**omputer **A**ssisted **S**oftware **E**ngineering) Rational Rose; esto facilitó el desarrollo de un proceso cooperativo entre todas las partes que intervinieron en la construcción del producto al poder disponer de diferentes vistas del mismo (vista de Casos de Uso, vista Lógica, vista de Componentes), compartiendo un mismo modelo a lo largo de todo el ciclo de vida del proyecto.

Según los autores de este lenguaje las vistas permiten mostrar diferentes aspectos del producto modelado. Una vista no es una gráfica, pero sí una abstracción que consiste en un número de diagramas y todos esos diagramas juntos muestran una "fotografía" completa del producto. Las vistas también ligan el lenguaje de modelado a los métodos o procesos elegidos para el desarrollo. Las diferentes vistas que oferta UML son:

- **Vista Use-Case:** Una vista que muestra la funcionalidad del sistema como la perciben los actores externos.
- **Vista Lógica:** Muestra cómo se diseña la funcionalidad dentro del sistema, en términos de la estructura estática y la conducta dinámica del sistema.
- **Vista de Componentes:** Muestra la organización de los componentes de código.

- **Vista Concurrente:** Muestra la concurrencia en el sistema, direccionando los problemas con la comunicación y sincronización que están presentes en un sistema concurrente.
- **Vista de Distribución:** muestra la distribución del sistema en la arquitectura física con computadoras y dispositivos llamados *nodos*.

En el diseño del producto obtenido se utilizaron específicamente vistas Use-Case para representar las funcionalidades a realizar por el sistema y los usuarios interesados en ellas empleándose para ello los conceptos de:

Actores: elementos que interactúan con la aplicación ya sea un humano, un software o hardware, con el interés de emplear alguna funcionalidad de este.

Bertami¹² plantea que generalmente, los actores son encontrados en la problemática mediante entrevistas con clientes y expertos del dominio. Las preguntas siguientes pueden usarse para ayudar a identificar a los actores de un sistema:

- ¿Quién está interesado en un cierto requisito?
- ¿En qué organización el sistema es usado?
- ¿Quién se beneficiará del uso del sistema?
- ¿Quién proporcionará al sistema la información, usará esta información, y quitará esta información?
- ¿Quién apoyará y mantendrá el sistema?
- ¿El sistema usa un recurso externo?
- ¿Una persona juega varios papeles o roles diferentes ante el sistema?
- ¿Varias personas juegan el mismo papel?
- ¿El sistema actúa recíprocamente con un sistema heredado?

Casos de Usos: agrupación de fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores.

Bertami señala que el empleo de las preguntas siguientes puede ayudar a identificar los Casos de Uso del sistema:

- ¿Cuáles roles juegan cada actor del Sistema?
- ¿Cualquier actor creará, guardará, cambiará, quitará, o leerá la información en el sistema?
- ¿Qué casos de uso apoyarán y mantendrán el sistema?
- ¿Qué caso de uso creará, guardará, cambiará, quitará, o leerá esta

¹² Bertamí Hernández K, "Software para el Marketing Forestal", Proyecto de Diploma presentado en opción al título de Ingeniero en Informática, Pinar del Río, 2007

información?

- ¿Cualquier actor necesitará información del sistema sobre cambios súbitos y externos?
- ¿Cualquier actor necesita ser informado sobre ciertas ocurrencias en el sistema?
- ¿Pueden todos los requisitos funcionales realizarse por los casos de uso?

Casos de Uso Abstractos son aquellos que facilitan la comprensión del Caso de Uso Base o es una extensión del comportamiento de este, pudiendo ser de dos tipos:

- **Casos de Uso Incluidos:** Los mismos se ejecutan al ejecutarse el Caso de Uso Base. Se justifica su empleo cuando:
 - ❖ Su contenido puede ser rehusado en otros Casos de Uso.
 - ❖ Simplifica la comprensión del Caso de Uso Base.
- **Casos de Uso Extendido:** Estos no necesariamente se ejecutan al ejecutarse el Caso de Uso Base. Se justifica su empleo cuando:
 - ❖ Existe una extensión del comportamiento del Caso de Uso Base.
 - ❖ Existen comportamientos del Caso de Uso Base que se ejecutan solo bajo determinadas condiciones.

II.1.2.- La Herramienta CASE “Rational Rose”

Arocha¹³, nos dice que “... Las Herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering), tienen su propia historia la que trataremos de describir de la manera más sintética. En la década de los setenta el proyecto ISDOS desarrolló un lenguaje llamado "Problem Statement Language" (PSL) para la descripción de los problemas de usuarios y las necesidades de solución de un sistema de información en un diccionario computarizado. Problem Statement Analyzer (PSA) era un producto asociado que analizaba la relación de problemas y necesidades. Pero la primera herramienta CASE como hoy conocemos para PC fue "Excelerator" en 1984. Actualmente la oferta de herramientas CASE es muy amplia entre muchas otras están: **Rational Rose**, Power Designer y MSVisio. Entre sus principales objetivos se encuentran:

- Aumentar la productividad de las áreas de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la calidad del software desarrollado.
- Reducir tiempos y costos de desarrollo y mantenimiento del software.
- Mejorar la gestión y dominio sobre el proyecto en cuanto a su planificación, ejecución y control.

¹³ Arocha, Aportela, Anairis, “Centro Virtual de Recursos Digitales del CECES”, tesis de diploma presentada en opción al título de Ingeniero en Informática, Pinar del Río, 2007

- Mejorar el archivo de datos (enciclopedia) de conocimientos y sus facilidades de uso, reduciendo la dependencia de analistas y programadores.

Fin del la cita.

Al caracterizar el CASE “Rational Rose” usando lo expresado por **Boggs**¹⁴ se resume que este es una herramienta de modelación visual que soporta de forma completa todas las especificaciones de UML permitiendo establecer una trazabilidad real entre lo modelo en el diseño y el código ejecutable; facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero comparten un mismo modelo a lo largo de todo el ciclo de vida del proyecto.

Una de las grandes ventajas de Rational Rose es el uso del Lenguaje Unificado de Modelado (UML, siglas en inglés), proporcionando a los arquitectos y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

La Corporación Rational ofrece una metodología para la construcción de un software, Proceso Unificado de Desarrollo de Software, desde la etapa de ingeniería de requerimientos hasta la de pruebas. Para cada una de estas etapas la Rational oferta una herramienta de ayuda en la administración de los proyectos, en el caso específico de **Rose** es la herramienta ofertada por la Rational para la etapa de análisis y diseño del producto.

Rose genera código fuente en distintos lenguajes de programación, tales como Java y C++, a partir de un diseño en UML y proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. Permitiendo además generar la base de datos en dependencia del gestor de base datos a utilizar.

II.2. Diseño de la Base de Datos.

En el capítulo precedente al caracterizar la actividad de auto control de la solidez del aprendizaje se realizó su Modelo Conceptual, Figura I.2.1 del capítulo I, tomando este modelo de base se construyó en con el CASE el Diagrama de Clases Persistente, figura II.2.1, del cual se generó el Modelo de Datos, figura II.2.2 en este proceso se empleó siguientes conceptos de Base de Datos Relacionales revisados en **López¹⁵, y Date¹⁶**

Entidad: Elemento, objeto, suceso o concepto del cual es de interés recoger información estando bien diferenciados entre sí, que poseen propiedades y entre los cuales se establecen relaciones. Las entidades consideradas pueden ser vistas en el Modelo de Datos, figura II.2.2.

Entidad Débil: Aquella entidad que su existencia depende de la existencia de otra entidad.

Elemento de Datos o Atributo: Propiedad o característica de interés que describe a una entidad o relación. Los elementos de datos o atributos de las entidades se pueden observar en el Modelo de Datos, figura II.2.2, en este aparecen debajo del nombre de la entidad.

Relación: Correspondencia o asociación entre dos o más entidades.

Los posibles tipos de relaciones que puede haber entre dos Entidades son:

- **Relaciones 1-1:** Cuando una instancia de una de las Entidades está asociada con una y solo una instancia de la otra y viceversa.
- **Relaciones 1-n:** Cuando una instancia de una Entidad 1 le corresponde muchas de Entidad 2, pero una instancia de Entidad 2 sólo puede estar asociada con una instancia de Entidad 1.
- **Relaciones n-n:** Cuando una instancia de una de las Entidades le corresponde muchas instancias de la otra y viceversa.

En el Modelo de Datos, figura II.2.2 aparecen relaciones de los tres tipos. Las relaciones del tipo n-n constituye en el Modelo Físico de los Datos una tabla, la cual tendrá como llave la unión de las llaves de cada una de las Entidades implicadas en la relación, pudiendo tener o no atributos propios la relación los que se convertirían en campos de la tabla. En el Modelo de Datos solo aparecen estas si poseen atributos propios o si se relaciona con otras entidades.

Llave Primaria: Atributo o conjunto de atributos de la entidad que permite identificar de forma única a cada elemento de la misma, esto hace que no pueda existir dos elementos en una Entidad con igual valor de la llave primaria, a su vez esta no puede tener un valor nulo. En el Modelo de Datos estas son señaladas con el prefijo **PK** (Primary Key).

Llave Foránea: Atributo o conjunto de atributos de la Entidad que son llave de otra Entidad con la cual se encuentra relacionada. Estas en el Modelo de Datos se señalan con el prefijo **FK**

Atributo mandatorio: Aquel que no puede tener valor nulo, de hecho todas las Llaves Primarias son mandatorias.

Dominio: Conjunto de valores en los que pueden tomar valor un atributo.

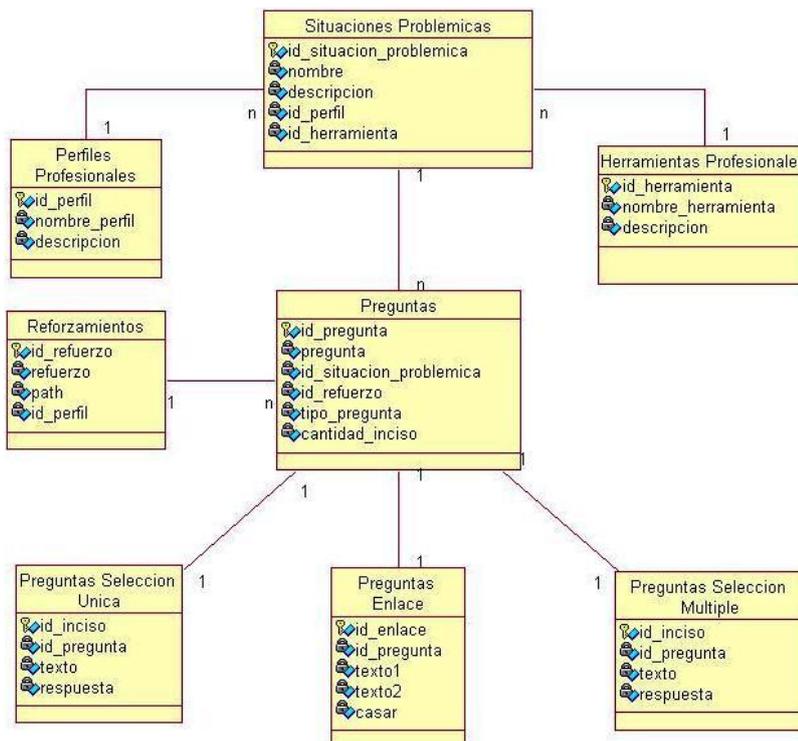


Figura II.2.1 Diagrama de Clases Persistente de SAESA

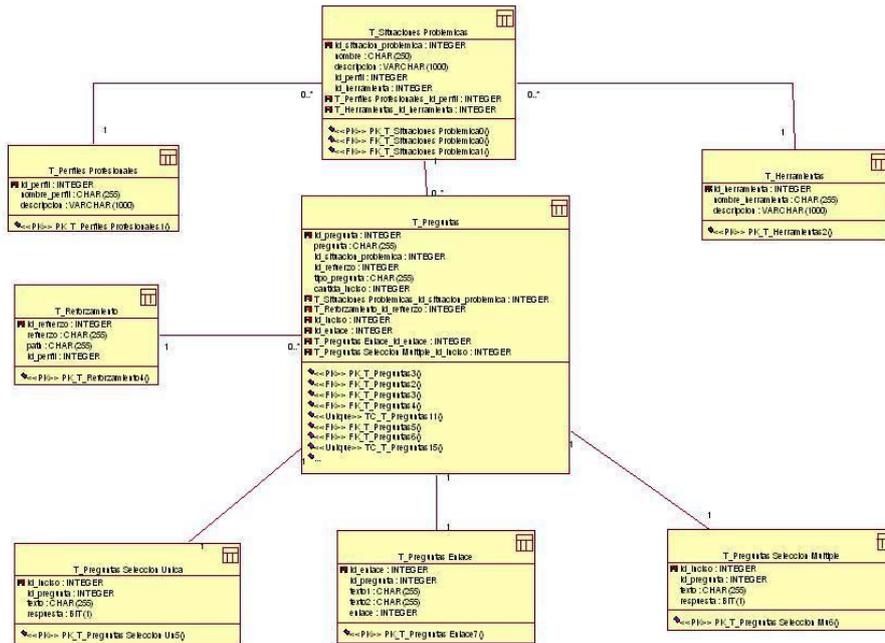


Figura II.2.2 Modelo de Datos de SAESA

II.3.- Diseño de las interfaces de usuario.

Para el diseño de las interfaces de usuario es importante del conocimiento de los elementos de datos que deben estar presentes según las funcionalidades que debe garantizar el producto a sus actores. La captura de estos requerimientos se realizó mediante entrevistas a los expertos (jefes de carrera y profesores experimentados).

Requerimientos Funcionales:

- R1.- Autenticación de los actores que administraran cada perfil y Herramienta.
- R2.-Autenticación del actor que administrara el registro de usuarios y de perfiles/herramientas
- R3.- Gestionar Datos del perfil inscrito
- R4.- Gestionar Datos de Situaciones Problematicas de un perfil
- R5.- Gestionar Datos de las Preguntas correspondientes a un perfil
- R6.- Gestionar documentos de reforzamientos para la pregunta
- R7.- Permitir auto evaluarse.
- R8.- Comparar respuesta con patrón de la pregunta
- R9.- Consultar reforzamiento.

Los actores de SAESAL, beneficiarios de las funciones brindadas por la herramienta auto evaluadora serán:

Actor	Rol
Alumno	Interesado en autoevaluar la solidez del aprendizaje adquirido durante el proceso de enseñanza-aprendizaje realizado en una carrera.
Administrador de Perfil o de Herramienta	Interesado en actualizar la base de evaluación de un perfil (Descripción del perfil, situaciones problemicas, preguntas, documentos de reforzamiento).
Administrador	Interesado en mantener actualizado el registro de usuarios e inscribir perfiles y herramientas.

Tabla II.3.1.- Roles jugados por cada actor de SAESA

A continuación se muestra con la vista de Caso de Uso las funcionalidades del producto relacionadas con sus actores.

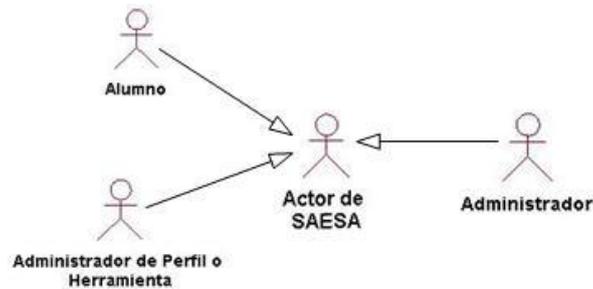


Figura II.3.1.- Actores de SAESA.

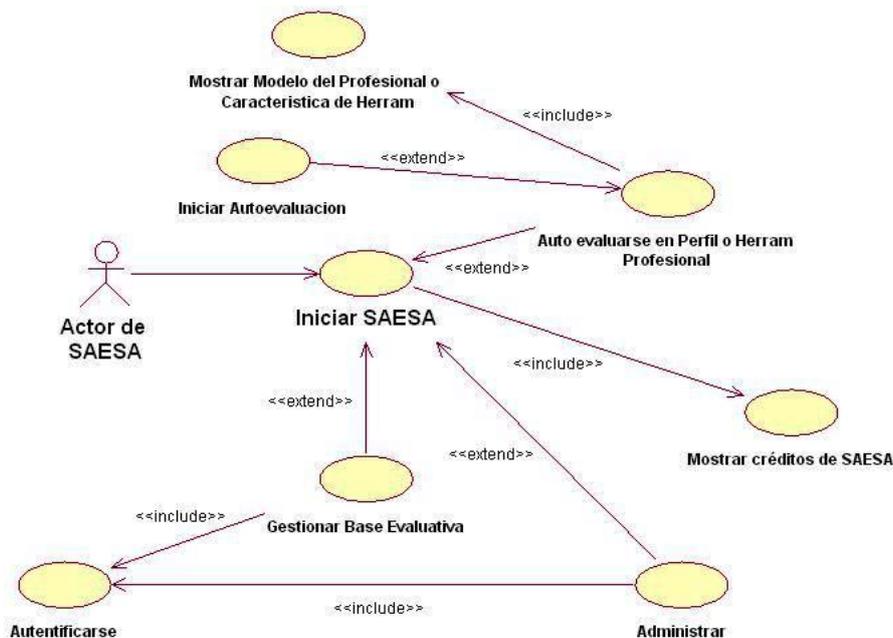


Figura II.3.2.- Diagrama de Caso de Uso de SAESA.

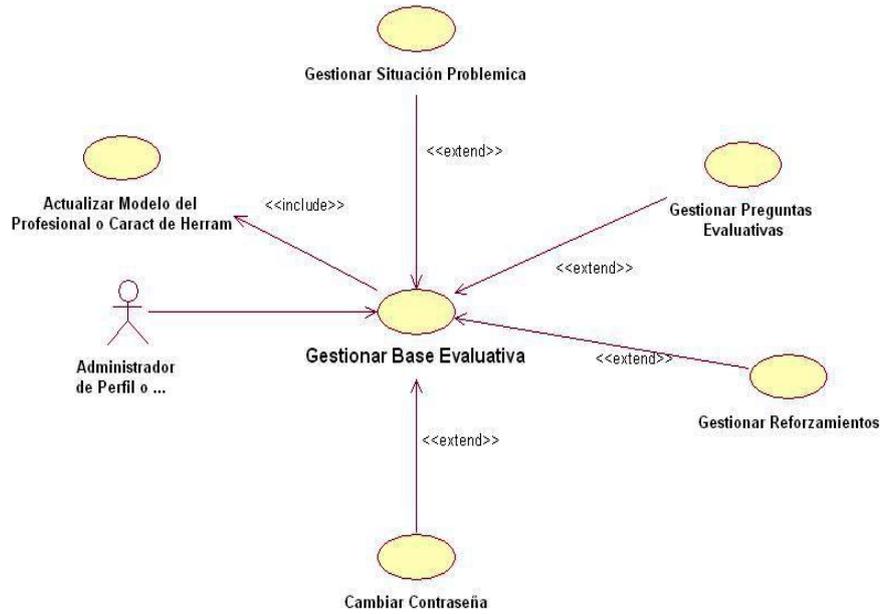


Figura II.3.3.- Diagrama Caso de Uso: “Gestionar Base Evaluativa”

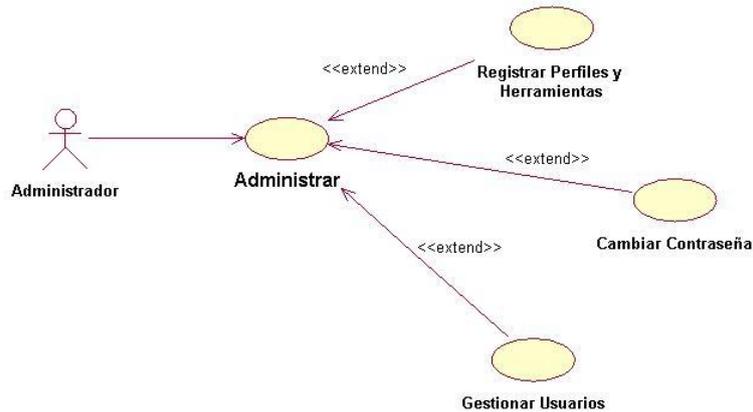


Figura II.3.4.- Diagrama Caso de Uso: “Administrar”

II.4.- Plan de Seguridad.

El Plan de seguridad se estableció siguiendo la siguiente estrategia.

- El acceso a la Base Evaluativa sólo será posible desde la aplicación, esta al conectarse enviara la contraseña a la Base de Datos. Al cerrar la aplicación la Base quedará cerrada.
- Se limita la gestión de los datos de la Base Evaluativa según el rol de cada actor, figura II.4.1

ACTOR**Funcionalidades permitidas****Alumno**

Evaluar solidez de su conocimientos en Perfil Profesional
Evaluar solidez de su conocimientos en Herram Profesional

**Administrador de Perfil o
Herramienta**

Editar Modelo del Profesional (Administrador Perfil)
Editar caracterización de la herramienta (Admin Herramienta)
Mantener actualizado registro de Situaciones Problemicas
Mantener actualizado registro de Preguntas
Mantener actualizado registro de Reforzamientos

**Administrador**

Registrar Perfil Profesional o Herramienta
Mantener actualizado registro de Usuarios

Figura II.4.1.- Plan de Seguridad de SAESA**II.5.- Arquitectura utilizada.**

Para tomar una decisión sobre que arquitectura escoger es necesario tener en cuenta los roles jugados por los actores. SAESA será usado por el alumno de años terminales con objetivo de auto evaluar la solidez de su aprendizaje resolviendo situaciones relativas al perfil profesional de la carrera que estudia, luego es necesario que la Base Evaluativa usada este ubicada en un lugar que permita su acceso desde diferentes PC apoyándose en la estructura de red existente en la Universidad de Pinar del Río, lugar donde se introducirá la herramienta evaluadora. Por otro lado es necesario facilitar la actualización de la Base Evaluativa por cada uno de los administradores de perfiles desde diversas PC de este entorno de red. Por las razones señaladas SAESA necesitará hacer uso de una arquitectura cliente/servidor, vea en las figuras de la II.5.1 a la II.5.4 lo que irá a lado del cliente y del servidor respectivamente. La decisión tomada sobre la arquitectura a emplear repercutirá en la elección del gestor de Base de Datos y del Lenguaje de Programación a usar, aspectos tratados en el próximo capítulo.

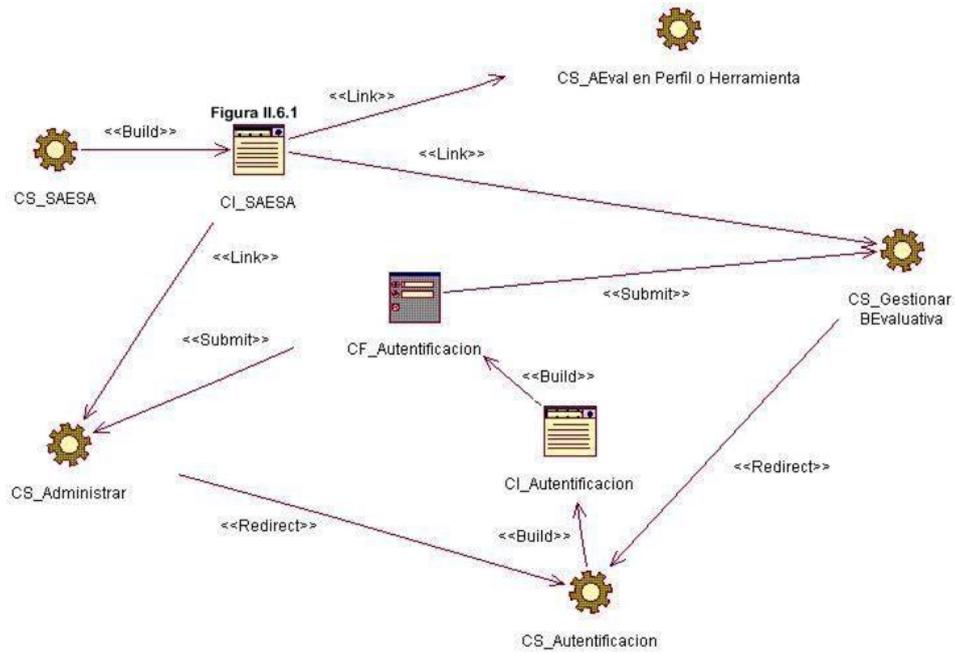


Figura II.5.1.- Diagrama de Clases de SAESA.

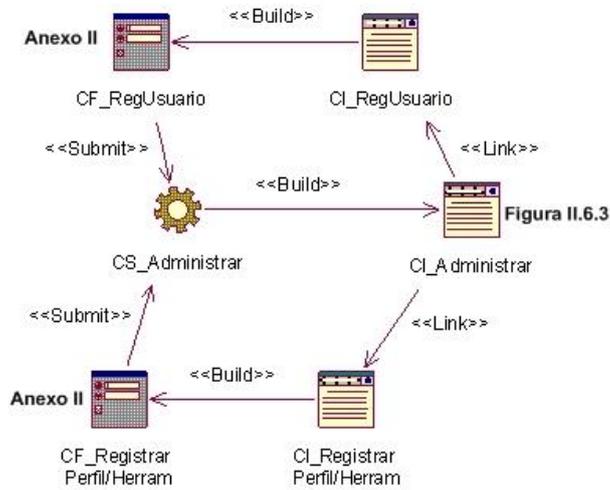


Figura II.5.2.- Diagrama de Clases "Administrar"

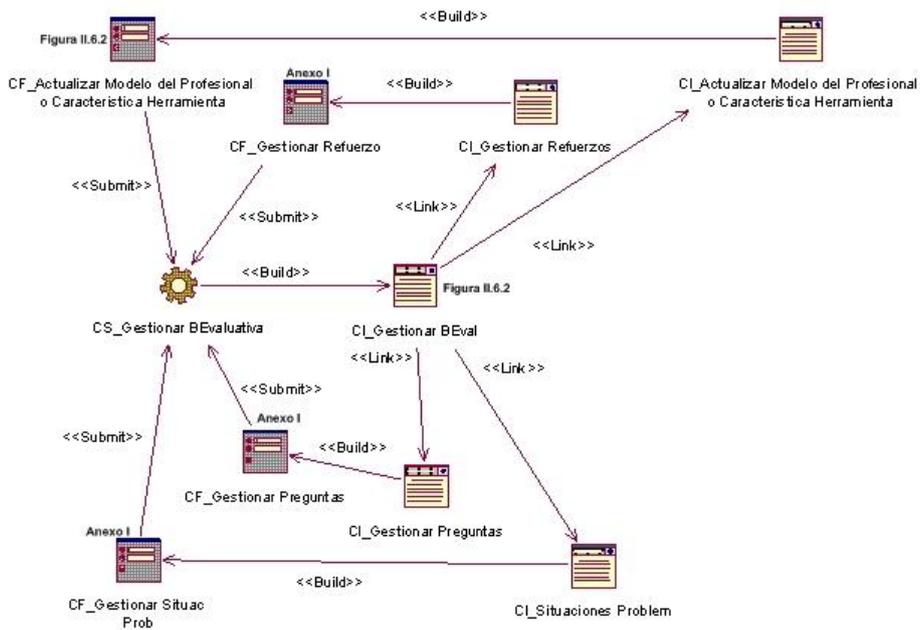


Figura II.5.3.- Diagrama de Clases “Gestionar Base Evaluativa”

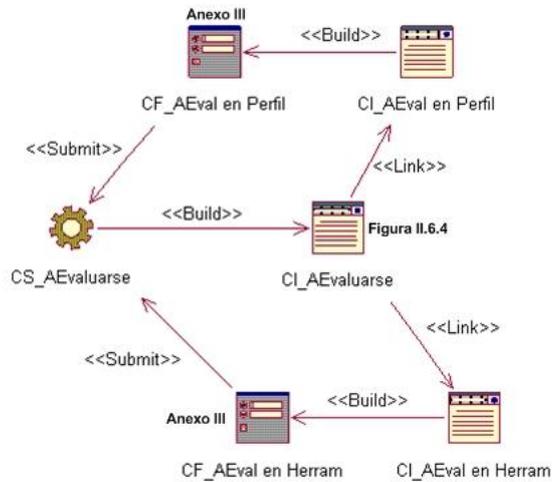


Figura II.5.4.- Diagrama de Clases “Auto Evaluarse”

II.6.- La Navegación en SAESA.

Al diseñar la navegación se tuvieron en cuenta los estándares de diseño preestablecidos para las aplicaciones WEB. Es importante señalar que las funcionalidades brindadas en la interfaz principal se corresponde con los Casos de Uso de SAESA, es decir, en esta interfaz el alumno podrá realizar el autocontrol de la solidez del aprendizaje en el perfil profesional o herramienta de la profesión que

deseo, respondiendo de esta forma la herramienta evaluativa a su razón de ser. Además en esta interfaz se brinda la posibilidad de autenticarse para así poder gestionar la base evaluativa. En la figura II.6.1 se muestra la interfaz inicial de la herramienta evaluadora, compare las funcionalidades en ella brindadas con la figura II.3.2, Diagrama de Caso de Uso de SAESA.

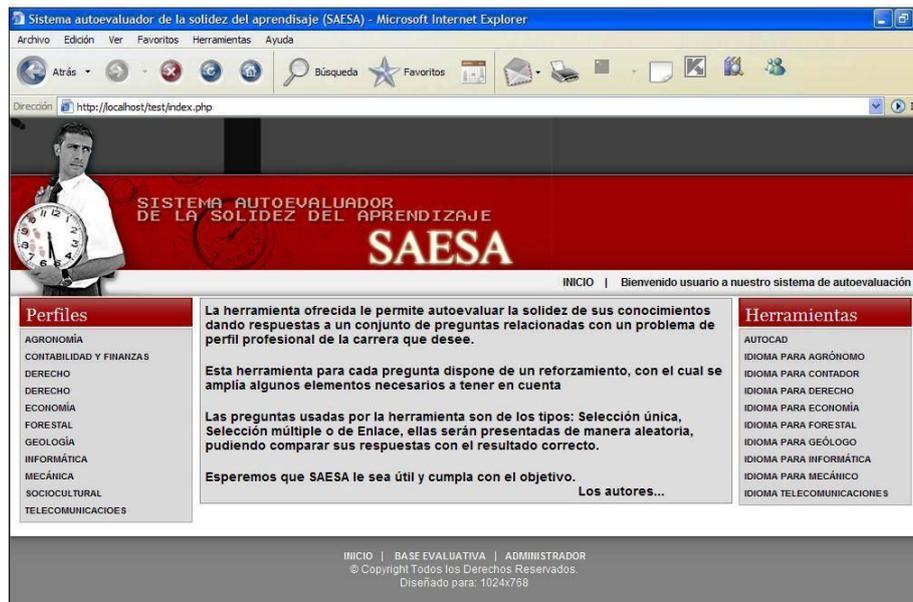


Figura II.6.1.- Interfaz Inicial de SAESA.

La interfaz para la gestión de la Base Evaluativa, figura II.6.2 se corresponde en este caso con el Diagrama de Casos de Uso “Gestión de la Base Evaluativa”, figura II.3.3.

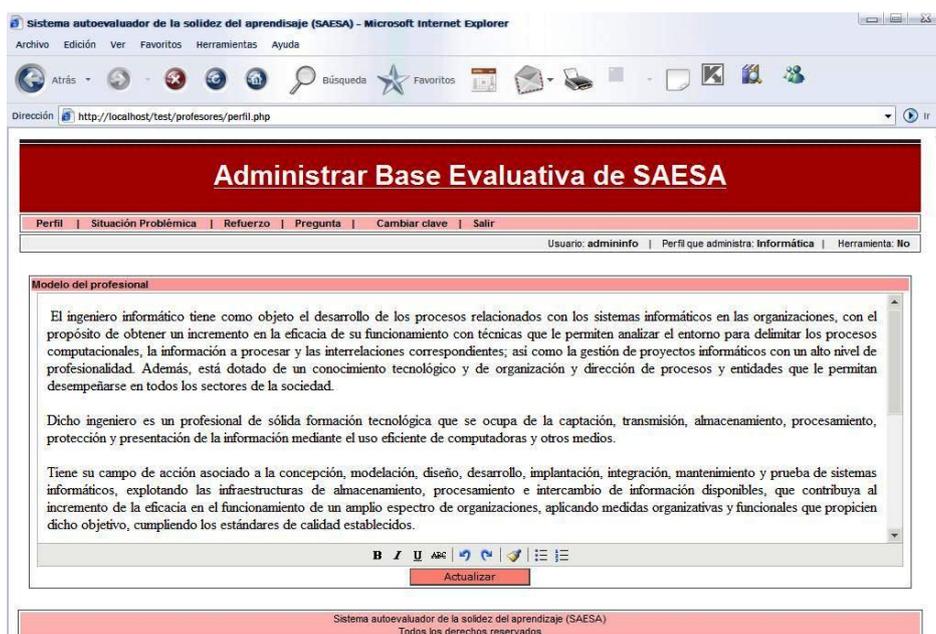


Figura II.6.2.- Interfaz “Gestión de la Base Evaluativa” de SAESA

La interfaz de Administrar, figura II.6.3 se corresponde con el Diagrama de Casos de Uso “Administrar”, figura II.3.4.

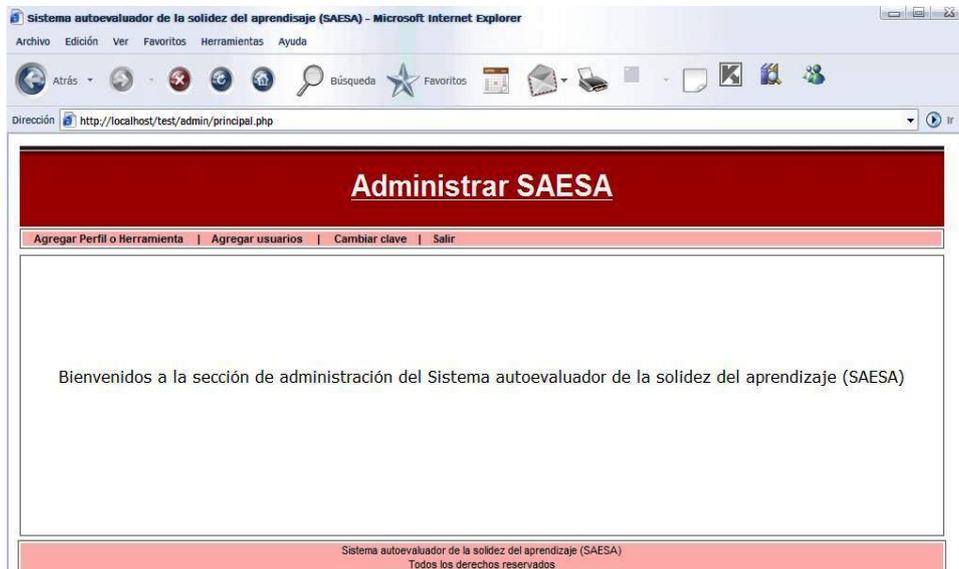


Figura II.6.3.- Interfaz “Administrar”

La interfaz de Auto Evaluarse, figura II.6.4 se corresponde con el Casos de Uso Extendido “Auto evaluarse en Perfil o Herramienta”, figura II.3.2.

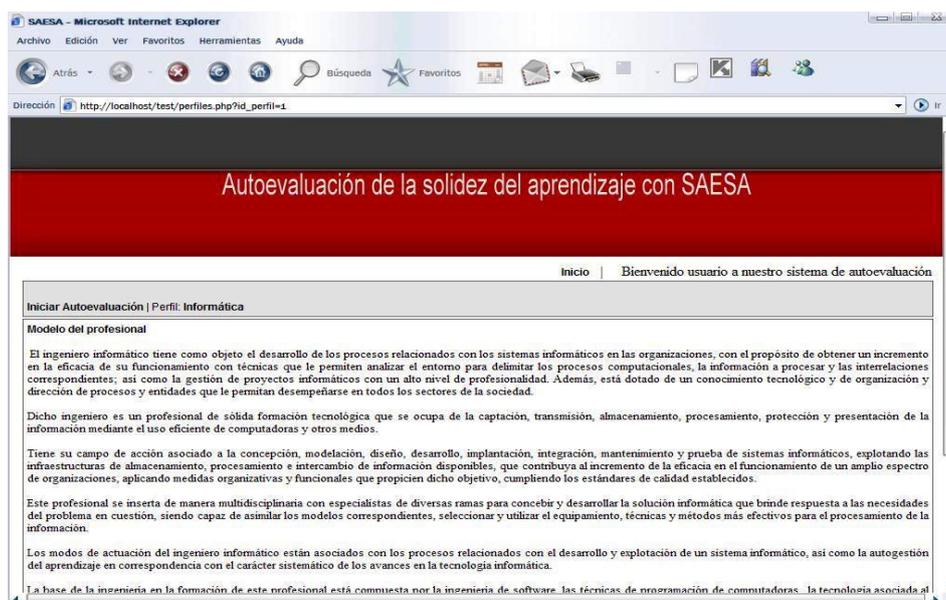


Figura II.6.4.- Interfaz AutoEvaluarse

La figura II.6.5 muestra el esquema de navegación en la herramienta evaluadora SAESA.

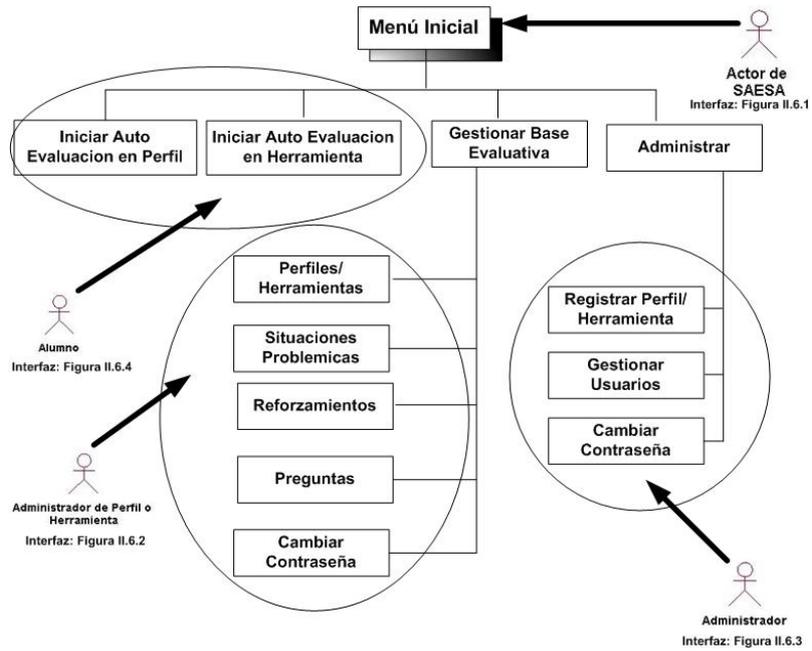


Figura II.6.4.- Navegación de los actores en SAESA.

CAPÍTULO III.- IMPLEMENTACIÓN DE SAESA**Introducción**

El capítulo aborda la construcción de la Base de Datos y de su seguridad empleando el gestor MySQL y el de las Interfaces de Usuario con uso del editor WEB Dreamweaver y del lenguaje de programación PHP

En su primer epígrafe se hace una caracterización del gestor usado para implementar el Modelo Datos, figura II.2.2 del capítulo precedente, finalizando el mismo con la presentación del Modelo Físico de los Datos obtenidos en MySQL.

El segundo epígrafe comienza caracterizando el editor WEB Dreamweaver y del lenguaje de programación PHP, usados para implementar las interfaces usuarios y reglas del negocio de SAESA, especificando algunas de las facilidades utilizadas de estas herramientas.

El capítulo concluye con el epígrafe donde se ilustra como se implemento el Plan de Seguridad de SAESA con uso del gestor MySQL y del lenguaje PHP.

III.1.- Construcción de la Base de Datos.

Con el objetivo de facilitar las tareas de administración de los datos y acelerar todo el proceso y desarrollo de la aplicación se hace necesario una selección correcta del Sistema Gestor de Base de Datos a emplear, una acertada decisión facilita el proceso de administración, reduce el tiempo de implementación del sistema y mejora su rendimiento global.

Korth¹⁷ define a un Sistema Gestor de Base de Datos (SGBD) como un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Su propósito general es el de manejar de manera clara, sencilla y ordenada el conjunto de datos de la base. Según **Martínez**¹⁸ las principales cualidades que inciden en la decisión de qué SGBD utilizar son:

- ❑ **Tamaño:** Que facilite el intercambio de información con memoria secundaria, la búsqueda rápida, etc., cuando el volumen de información aumente, no es el caso de la base manejada por SAESAL.
- ❑ **Concurrencia:** Poseer un mecanismo de control sobre la información cuando sobre ella estén interactuando varias personas o programas de forma concurrente, por ejemplo en arquitecturas Cliente / Servidor, como es el caso de la herramienta evaluadora construida.
- ❑ **Recuperación e Integridad:** Que permita proteger la información de estados inestables provocados por fallos de energía, de la propia aplicación o algún otro tipo de fallo, siempre dejando la información en un estado consistente.
- ❑ **Distribución:** Que posibilite que la información esté almacenada en diferentes lugares.
- ❑ **Seguridad:** Que permite restringir el acceso a la información a usuarios no autorizados.
- ❑ **Administración:** Que permita a los usuarios y administradores de Bases de Datos examinar, controlar y ajustar el comportamiento del sistema.

A continuación abordaremos las principales ventajas de un gestor de Base de Datos de tipo Cliente/Servidor, por tener la necesidad que los datos de la Base Evaluativa utilizada por SAESA sean administrados y consultados desde diferentes lugares.

¹⁷ Korth H.F.; Silberschatz A. "Database Systems Concepts". McGraw-Hill, 1986.

¹⁸ Martínez Prieto, A. B. "Introducción a los SGBDOO". Universidad de Oviedo, Mayo de 2004. URL: <http://www.di002.edv.uniovi.es>

- Todos los elementos de datos están almacenados en una ubicación central donde todos los usuarios pueden trabajar con ellos. No se almacenan copias separadas del elemento en cada cliente, lo que a los clientes tener las actualizaciones realizadas a los datos.
- Las reglas de empresa y de seguridad se pueden definir una sola vez en el servidor para todos los usuarios.
- Los servidores de Base de Datos relacionales optimizan el tráfico de la red al devolver solo los datos que la aplicación necesita.
- Los costos de hardware pueden ser minimizados. Como los datos no se guardan en cada cliente, los clientes no tienen que dedicar espacio del disco a guardar los datos. Tampoco necesitan la capacidad de procesamiento para manejar los datos desde su PC, por otra parte el servidor no necesitará dedicar tiempo para mostrar los datos.
- El servidor puede ser configurado para optimizar las capacidades de entrada/salida del disco necesitado para recuperar los datos mientras que los clientes pueden ser configurados para optimizar el formato en que los datos recuperados del servidor serán mostrados.
- El servidor puede ser situado en un lugar relativamente seguro y equipado con dispositivos de respaldo energético, lo cual es más económico que proteger a cada cliente.
- Las tareas de mantenimiento como la salva y restauración de los datos son simplificadas porque se pueden ser realizadas en el servidor central.

La herramienta evaluadora SAESA usa una Base de Datos centralizada en un Servidor MySQL que permite a la aplicación cliente acceder y gestionar de forma rápida y eficiente la información. A continuación se exponen algunas de las características del gestor usado que refuerzan su elección para implementar el Modelo de Datos examinado en el capítulo precedente.

III.1.1.- Criterios considerados para elegir a MySQL como gestor.

MySQL es un SGBD con interfaz SQL que inicialmente buscó una compatibilidad con la API de mSQL. Es el servidor de base de datos “Open Source” más utilizado en todo el mundo, se puede adquirir gratis en Internet y no es necesario pagar licencia por su explotación. Se utiliza mucho en la creación de aplicaciones Web porque es muy rápido, confiable, y fácil de usar. Sus principales características han sido la velocidad, la robustez y además de ser multiplataforma. Al igual que Oracle, está soportado por

la gran mayoría de los sistemas operativos tales como: Solarix, Linux, Windows, Mac OS X Server, etc.

MySQL es un SGBD que requiere de una herramienta auxiliar para la manipulación gráfica de los diferentes componentes que integran la Base de Datos. La herramienta utilizada al construir la Base Evaluadora de SAESA fue MySQL phpMyAdmin, en la figura III.1.1.1 se muestra el entorno de esta herramienta que facilito la construcción de la Base de Datos de SAESA



Figura III.1.1.1.- Entorno de la herramienta phpMyAdmin.

La Tabla III.1.1.1 contiene los tipos de datos soportados en MySQL

Bit	Date	TinyText	Set
TinyInt	Date Time	Text	Geometry
SmallInt	Time Stamp	MediumText	Point
MediumInt	Time	LongText	LineString
Int	Year	TinyBlob	Polygon
BigInt	Char	Blob	MultiPoint
Float	VarChar	MediumBlob	MultiLineString
Double	Binary	LongBlob	MultiPolygon
Decimal	VarBinary	Enum	GeometryCollection

Tabla III.1.1.1. – Tipos de Datos soportados en MySQL



Figura III.1.1.2.- Tipos de datos de la tabla Preguntas.

III.1.2.- El Modelo Físico de los Datos

Al implementar el Modelo de Datos usando este gestor se realizaron algunas modificaciones dirigidas a ganar en eficiencia al manipular los registros. Estas fueron:

- Se unió en la tabla “Perfiles”, la propia entidad “Perfiles” y la entidad “Herramientas”, para lo cual le fue agregada a la primera el atributo booleano Herramienta que permite distinguir si un registro se trata de un perfil o de una herramienta.
- Las entidades “Preguntas de Selección Única” y “Preguntas de Selección Múltiple” fueron unidas en la tabla “Incisos Selección”

En la figura II.1.2.1 aparece la Base de Datos de SAESA implementada con uso de MYSQL.

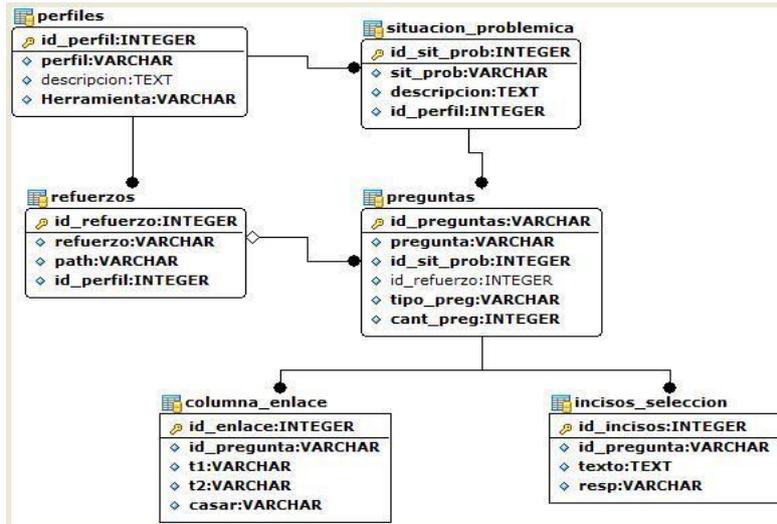


Figura III.1.2.1.- Modelo Físico de Datos de SAESA.

El Modelo mostrado se le agrega la tabla “User”, la que contiene los administradores de Perfiles y de Herramientas con sus contraseñas y la del usuario Administrador, información utilizada en la autenticación para limitar la navegabilidad según rol jugado en SAESA del autenticado.

III.2.- Construcción de las Interfaces de Usuario.

La elección de un Lenguaje y entorno Programación adecuado para implementar las interfaces de usuario de un software es una tarea que requiere especial atención, hay que asegurar que la elección sea la más apropiada para dar cumplimiento a los requerimientos funcionales, teniendo en cuenta el tipo de arquitectura usada y la comunicación de este lenguaje con el gestor de Base de Datos elegido.

SAESA un sistema de arquitectura cliente/servidor que gestiona una Base de Datos, implementada en MySQL cuyos objetivos están centrados en lograr una interfaz amigable para el usuario y ser capaz de cumplir con sus requerimientos y dar respuesta en el menor tiempo posible. Por sus potencialidades en el trabajo con Bases de Datos con MySQL el lenguaje idóneo a usar es el PHP combinando con el Lenguaje SQL para realizar las operaciones necesarias con los datos, usando como editor WEB el Dreamweaver. A continuación se caracterizan cada uno de estas tecnologías.

II.2.1. - Personal Hypertext Preprocessor (PHP).

Resumiendo lo señalado por **Gallego**¹⁹ PHP es un lenguaje de programación pensado en la web de forma tal que resulta ideal para la creación de páginas dinámicas. PHP es la versión libre del sistema equivalente de Microsoft ASP. Este lenguaje es encapsulado dentro de los documentos HTML. De forma que se pueden introducir instrucciones PHP dentro de las páginas. Gracias a esto el diseñador gráfico de la web puede trabajar de forma independiente al programador PHP, es interpretado por el Servidor Web Apache generando un fichero HTML con el resultado de sustituir las secuencias de instrucciones PHP por su salida. Por lo tanto una web dinámica con PHP contiene una serie de documentos PHP que el servidor apache interpreta proporcionando al cliente documentos HTML con el resultado de las ordenes PHP.

En este lenguaje la programación es realizada del lado del servidor integrado una gran cantidad de plataformas, nos permite programar aplicaciones asociadas al servidor de Web, aumentando la funcionalidad de dicho servidor y convirtiéndolo en un sistema de desarrollo de aplicaciones cliente/servidor mucho más completo. La mayoría de sus sintaxis está basada en C, Java y Perl. El principal objetivo del lenguaje es permitir a los desarrolladores de aplicaciones basadas en Web escribir páginas que se generan de forma dinámica de una forma sencilla y rápida. Esta tecnología es “Open Source” y tiene una gran integración con el servidor de Base de Datos MySQL, resaltando por:

- Ser muy sencillo de aprender.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan por dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Puede ser incrustado código PHP con etiquetas HTML.
- Posee un excelente soporte de acceso a Base de Datos.
- La comprobación de que parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar chequear que no se reciban solicitudes adulteradas.
- Viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Soporta hacer todo lo que se pueda transmitir por vía HTTP.

¹⁹ Gallego Vázquez, J. A., “Desarrollo Web con PHP y MySQL”, Ediciones Anaya Multimedia, 2003.

Sus desventajas son:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al ser mezclar con sentencias HTML
- La orientación a objetos es aún muy deficiente para aplicaciones grandes.

II.2.2. - Structured Query Language (SQL).

Todos los SGBD, incluyendo el MySQL basan su funcionamiento en un estándar de lenguaje de Bases de Datos SQL que será abordado a continuación:

El SQL es un lenguaje de alto nivel, no procedural, normalizado que permite la consulta y actualización de los datos de Base de Datos Relacionales. Actualmente se ha convertido en un estándar de lenguaje de Bases de Datos y la mayoría de los sistemas de Bases de Datos lo soportan, desde sistemas para ordenadores personales, hasta los dirigidos a grandes ordenadores. Por supuesto, a partir del estándar cada sistema ha desarrollado su propio lenguaje SQL que puede variar de un sistema a otro, pero con cambios que no suponen ninguna complicación para alguien que conozca un SQL concreto.

El SQL permite realizar consultas a la Base de Datos; además de poder realiza funciones de definición, control y gestión de la Base de Datos e incluye una interfaz que permite el acceso y la manipulación de la Base de Datos a usuarios finales. Las sentencias SQL se clasifican según su finalidad dando origen a tres sublenguajes:

- ***El Lenguaje de Definición de Datos (Data Description Language)***, incluye órdenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre éstas. (Es el que más varía de un sistema a otro).
- ***El Lenguaje de Control de Datos (Data Control Language)***, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la comparación de datos por parte de usuarios concurrentes, asegurando que no interfieran unos con otros.
- ***El Lenguaje de Manipulación de Datos (Data Manipulation Language)***, nos permite recuperar los datos almacenados en la Base de Datos incluyendo órdenes

que permite al usuario actualizar la Base de Datos añadiendo nuevos datos, suprimiendo datos antiguos o modificando datos previamente almacenados.

En la figura III.2.2.1, se ilustra el uso del lenguaje SQL para realizar consultas a los datos situados en el servidor MySQL.

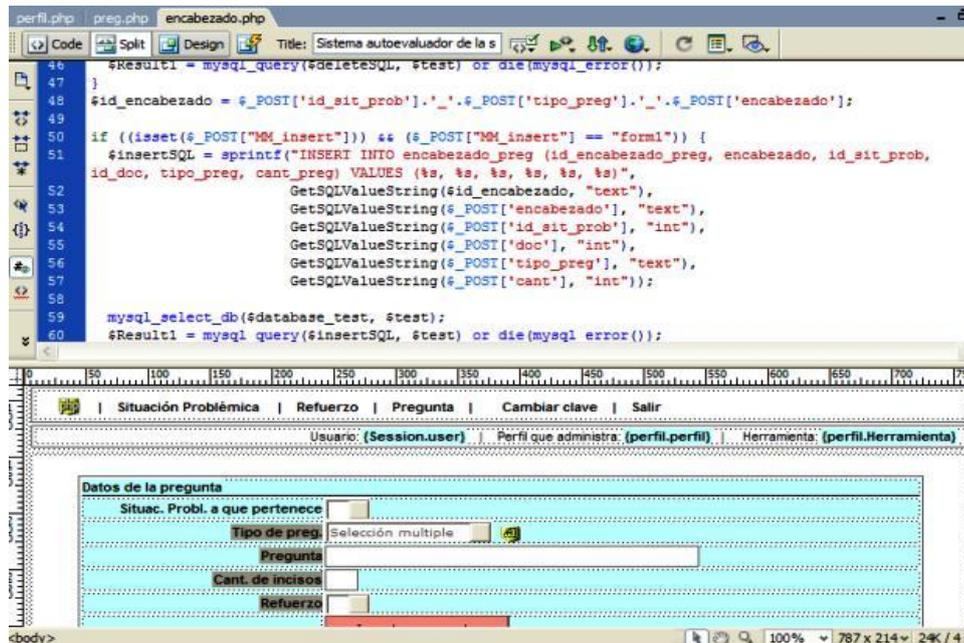


Figura II.2.2.1.- Uso del Lenguaje SQL para consultar los datos

III.2.3. - El editor WEB Dreamweaver.

Resumiendo expresado por Yanover²⁰ Dreamweaver es una herramienta para la creación de páginas y sitios web, que ofrece elementos capaces de controlar los vínculos de un sitio web, además de poder integrarse con publicación dinámica y soluciones de comercio electrónico. En Dreamweaver aparece, como novedad, la elección de una modalidad de programación, lista formada por ASP.Net, PHP, ColdFusion, y HTML, luego presenta otra selección: el ambiente de trabajo, donde encontramos las opciones, ya conocidas de anteriores versiones, WYSIWYG que consiste en diseñar una página web sin necesidad de escribir ningún código, la opción de trabajar con el código, y por último la posibilidad de ver ambas ventanas de desarrollo a la vez. Domina los lenguajes de programación ASP, CSS, PHP, SQL, JSP, y XML. El potencial del software en cuanto a la capacidad de programar bajo los

²⁰ Yanover, D. A. "Dreamweaver vs. FrontPage".2006, URL: <http://www.VitaminaWEB.com>



lenguajes que acabamos de citar es de lo más amplio, permitiendo la creación de aplicaciones y diseños web avanzados. Uno de los puntos de mayor énfasis en Dreamweaver es el soporte y las características de desarrollo en Cascading Style Sheet (cascada de hoja de estilo), haciendo posible creaciones con más facilidad y precisión, aplicando herramientas capaces de inspeccionar el código escrito. Algunas de sus mayores virtudes son:

- **Compatibilidad:** Además del diseño que pueda realizarse con esta herramienta, los plug-ins de Flash, Shockwave, Real Media y todos los compatibles con Netscape pueden controlarse en la página de Dreamweaver con el botón de inicio y detener.
- **Control:** Existe la herramienta site map con la cual es posible realizar el diseño y organización del sitio, ofrece una vista global del mismo con sus vínculos correspondientes. Cuando hay un cambio este es reflejado automáticamente en el mapa de sitio.
- **Búsqueda automática:** La búsqueda y modificación de acciones es de manera automática, como en Word, incluso es posible cambiar los colores del fondo de todo el sitio, o los atributos de ciertas tareas.
- **Trabajo en equipo:** los miembros de un equipo de trabajo pueden editar directamente alguna página sin romper con el diseño, al poder asegurarse regiones que no cambien su diseño y dejar otras para cambiar el contenido del texto pero sin modificar el diseño.

Resumiendo las principales características de este editor WEB que lo hacen una herramienta muy potente son:

- Poder obtener el control total sobre el código fuente, gracias a la Split View (Vista dividida) que permite observar el código y el diseño simultáneamente.
- Poder identificar fácilmente palabras claves y secuencias de comandos (scripts) en el código. El editor de texto integrado incluye coloreado del código ASP, PHP y JSP, sangrías automáticas y números de línea.
- Se maximiza la productividad con Server Behaviors (Comportamientos de servidor). Esta innovación, crea el formato y las secuencias de comandos del servidor, que se necesitan para las aplicaciones Web comunes, como actualizaciones e inserciones a las Bases de Datos.
- Cuenta con elementos de edición de tablas con lo que se pueden seleccionar de manera rápida celdas, renglones, columnas o una combinación de éstas.

- Los colores no están restringidos, ya que se pueden personalizar e incluso copiar un color de una gráfica y salvarlo en la paleta de colores de Dreamweaver.

A continuación ilustremos del uso de algunas de las potenciales brindadas por estas tecnologías WEB utilizadas en el desarrollo de la herramienta evaluadora SAESA.

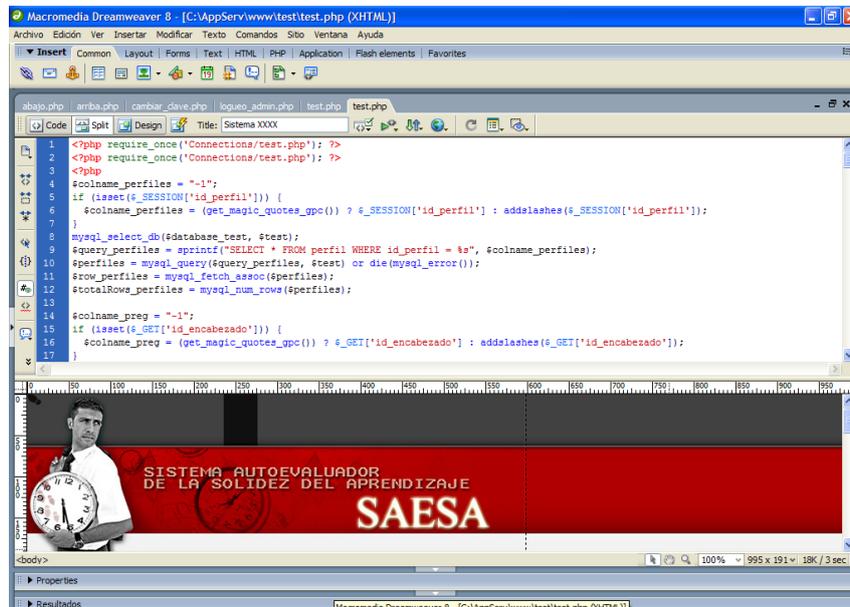


Figura III.2.1- Uso de Split View de Dreamweaver (Vista simultanea de código/diseño).

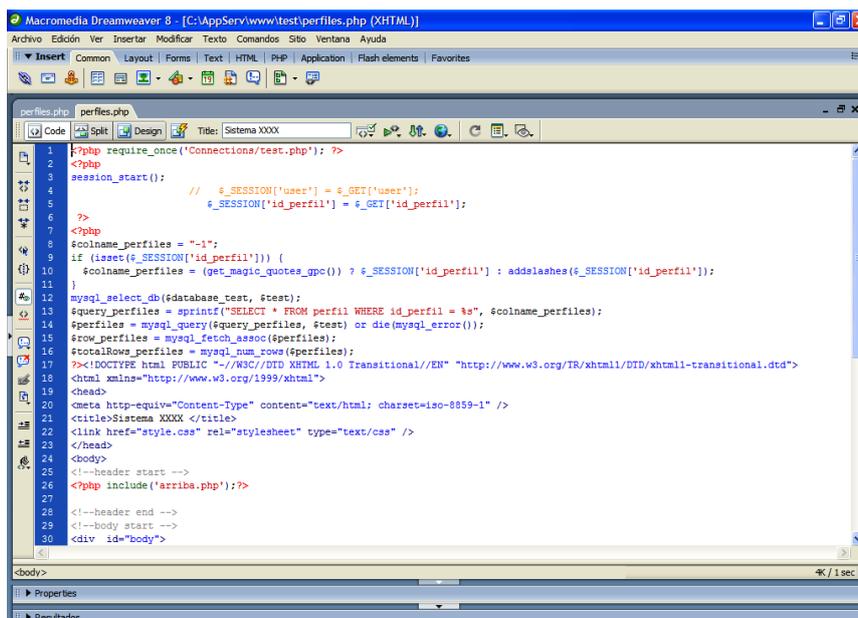
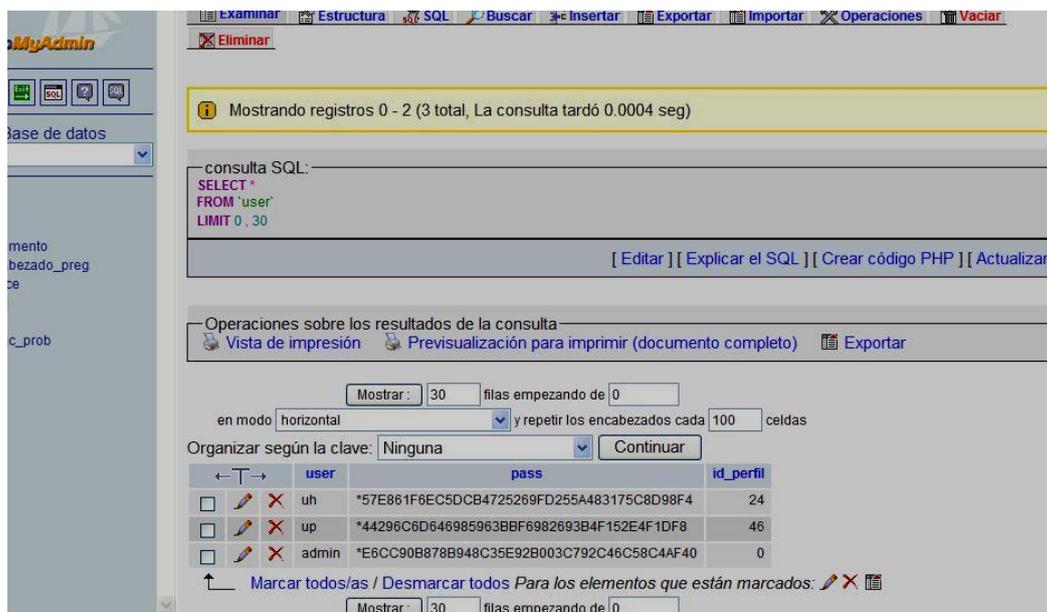


Figura III.2.2- Uso del coloreado del código en Dreamweaver, según el lenguaje

De manera general la implementación del SAESA estuvo enmarcada en el uso de tecnologías Open Source, independientemente del editor Web utilizado, la mayor utilidad aprovechada en cuanto a las tecnologías utilizadas se refiere a la fácil integración entre el gestor MySQL y el lenguaje PHP. La posibilidad de vincular código HTML, PHP, JSP y de realizar consultas SQL de manera directa sobre Dreamweaver dinamizó de manera espectacular el trabajo de implementación y codificación, la creación de plantillas de trabajo, el establecimiento y delimitación de regiones marcándolas como editables y no editables, así como la característica Cascading Style Sheet que facilita el diseño y rediseño en el transcurso de la ejecución del sitio, posibilitando hacer modificaciones a componentes y que estas modificaciones sean reflejadas en todas aquellas páginas que lo contengan.

III.3.- Implementación del Plan la seguridad

Resumiendo el Plan de Seguridad de SAESA presentado en el capítulo II consiste por un lado solo permitir el acceso a la Base de Datos sólo a través de la aplicación, lo que cual es necesario tenerla protegida con usuarios ya definidos desde el propio gestor de los cuales solo el desarrollador conoce su contraseña y por el otro lado limitar las funcionalidades a utilizar según el rol del usuario. Las figuras de la III.3.1 hasta la III.3.4 dan una idea de la implementación de este plan.



Mostrando registros 0 - 2 (3 total, La consulta tardó 0.0004 seg)

```
consulta SQL:
SELECT *
FROM `user`
LIMIT 0, 30
```

Operaciones sobre los resultados de la consulta

Vista de impresión Previsualización para imprimir (documento completo) Exportar

Mostrar: 30 filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna Continuar

	user	pass	id_perfil
<input type="checkbox"/>	uh	*57E861F6EC5DCB4725269FD255A483175C8D98F4	24
<input type="checkbox"/>	up	*44296C6D646985963BBF6982693B4F152E4F1DF8	46
<input type="checkbox"/>	admin	*E6CC90B878B948C35E92B003C792C46C58C4AF40	0

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Mostrar: 30 filas empezando de 0

Figura III.3.1- Editando usuarios y contraseñas con la herramienta phpMyAdmin.

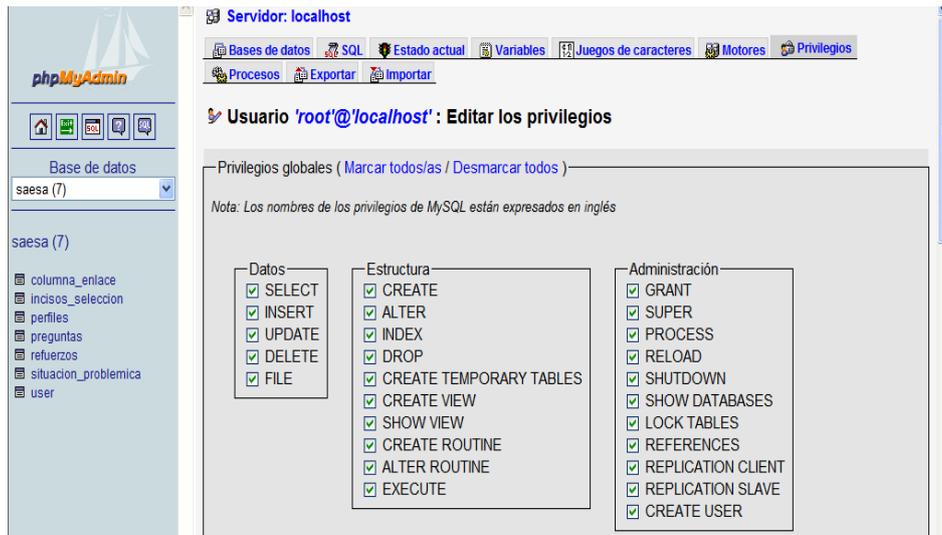


Figura III.3.2.- Privilegios a nivel de objetos de la Base de Datos.

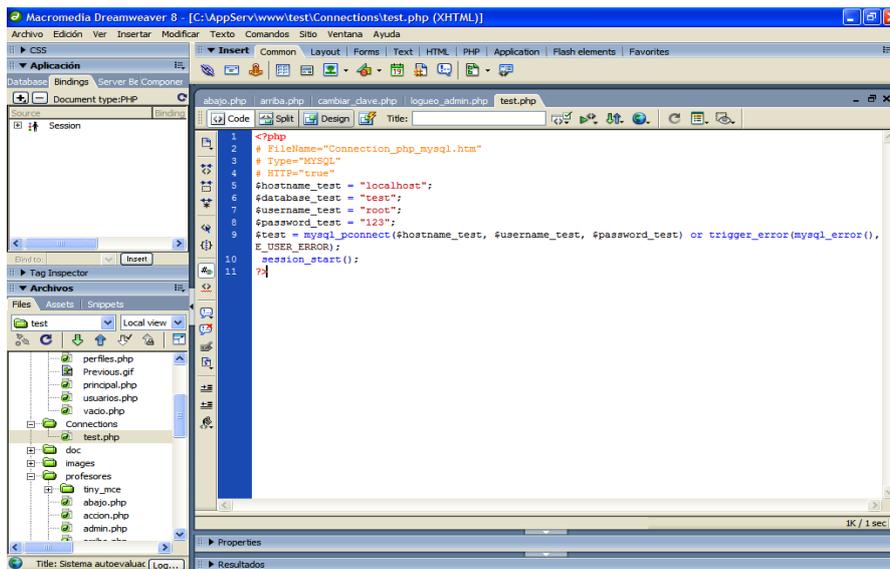


Figura III.3.3- Conectando Usuario a la Base de Datos desde SAESA

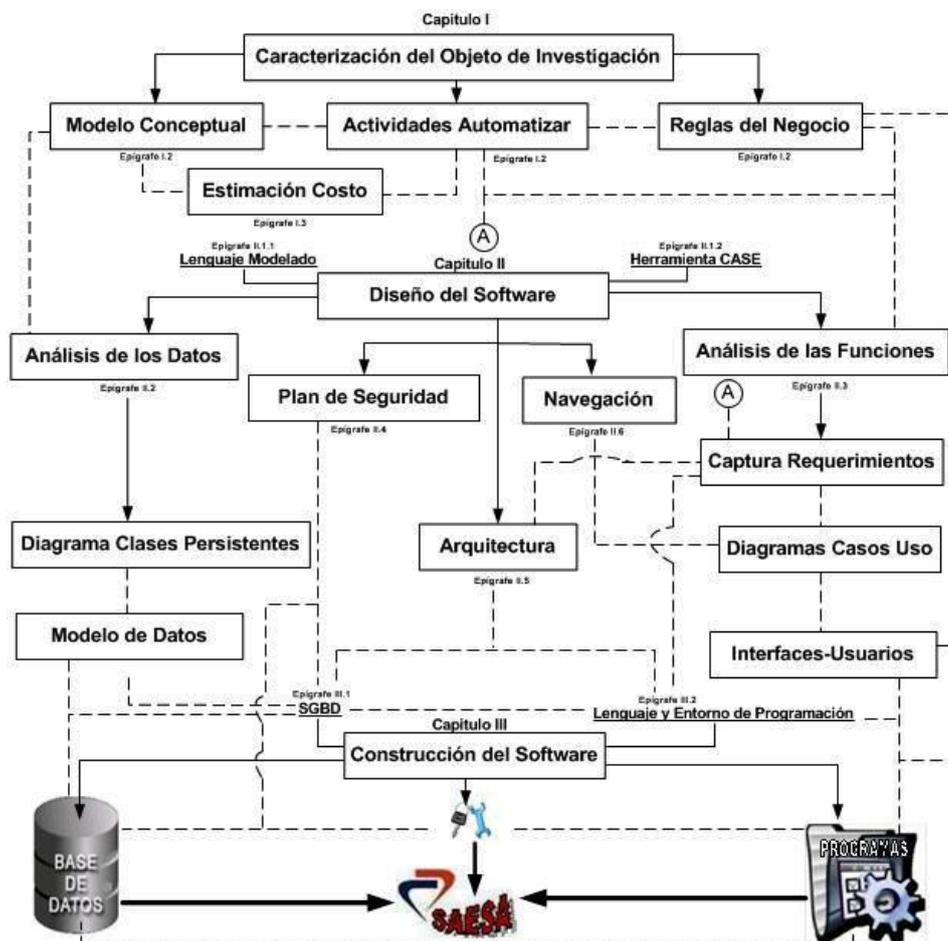
```
if ($accion=='logueo')
{
    mysql_select_db($database_test, $test);
    $query_logueo = "SELECT * FROM user WHERE (user = '". $_GET['user'] ."' and pass=PASSWORD('".
$_GET['pass'] ."'));";
    $logueo = mysql_query($query_logueo, $test);
    $row_logueo = mysql_fetch_assoc($logueo);
    if ($row_logueo['user']!='')
    {
        session_start();
        $_SESSION['user'] = $_GET['user'];
        $_SESSION['id_perfil'] = $row_logueo['id_perfil'];
        session_register("aqui");
        header ("Location:perfil.php");
    }
    else { header ("Location:logueo.php?paso=no"); }
    mysql_free_result($logueo);
}
```

Figura III.3.4.- Sección de Autenticación del usuario que posibilitara limitar su navegación en SAESA según sea su rol.

CONCLUSIONES

- Se construyó la herramienta evaluadora **SAESA** que permite al alumno autoevaluar la solidez de sus conocimientos de la profesión que estudia o en una herramienta usada en la misma.
- Se construyó la estructura de la Base Evaluativa necesitada por la herramienta evaluadora.

La lógica del proceso de investigación realizado hasta obtener la herramienta se muestra a continuación.



“Lógica del Proceso de Investigación”

RECOMENDACIONES

- Llenar la Base Evaluativa por los expertos, administradores de perfiles o de herramientas.
- Al incorporar las preguntas tener en cuenta que el alumno para poder decidir cuál de las respuestas son correctas tenga que realizar una acción que le permita llegar a una conclusión. (Regla no controlada por la herramienta)
- Tener en cuenta las invariantes, así se evaluará lo esencial. (Regla no controlada por la herramienta).
- Introducir la herramienta evaluadora para el perfil de Ingeniero Informático y de Telecomunicaciones (carreras estudiadas en la Facultad en que se realizó la investigación), tomando como herramientas de la profesión: idioma para Informáticos, idioma para Telecomunicadores, herramienta CASE “racional Rose”, herramientas utilizadas para el diseño de circuitos electrónicos.
- Darle mantenimiento a SAESA para solucionar los problemas que ocurran durante su explotación por los alumnos de las carreras seleccionadas.
- Generalizar el uso de la herramienta evaluadora, teniendo en cuenta la experiencia adquirida durante su explotación, a las restantes carreras de la Universidad de Pinar del Río.

ANEXOS

Anexo I.- Interfaces para la Gestión de la Base Evaluativa de SAESA.

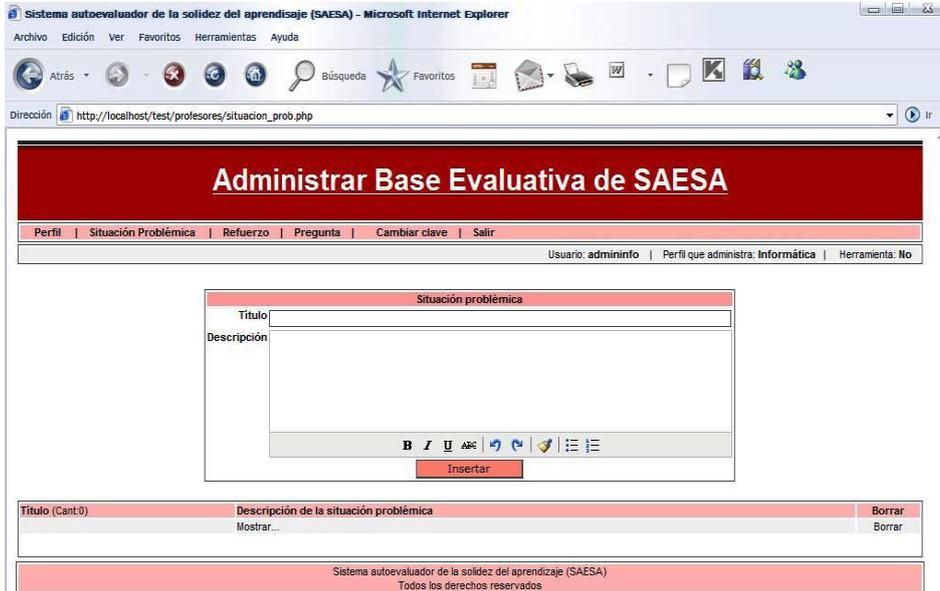


Figura A1.1.- Interfaz “Gestionar Situación Problemática”.

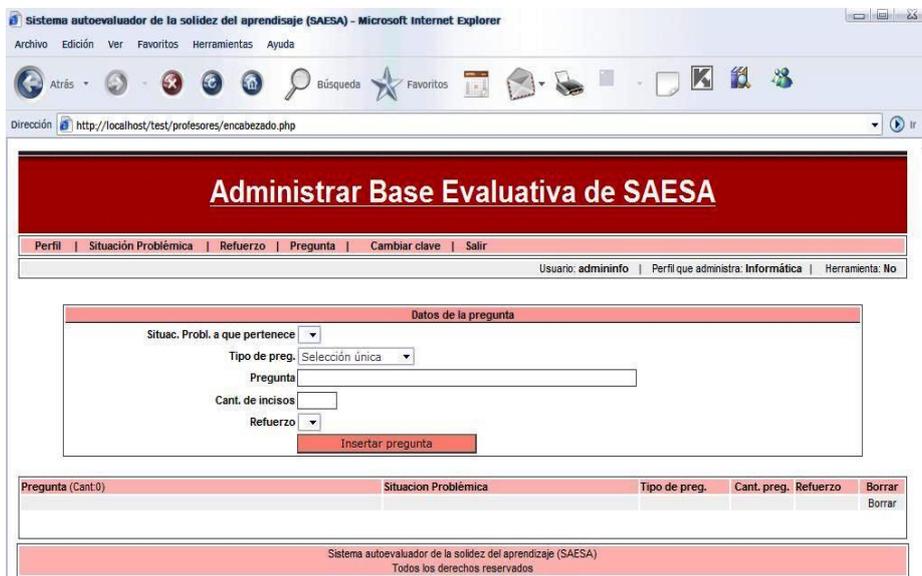


Figura A1.2.- Interfaz “Gestionar Pregunta”.

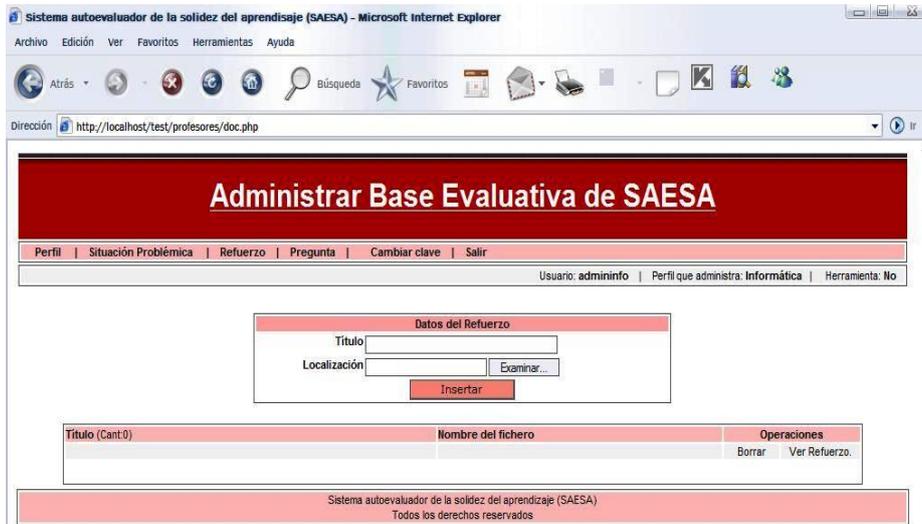


Figura A1.3.- Interfaz “Gestionar Refuerzo”

Anexo II.- Interfaces para la Gestión Administrativa.



Figura A2.1.- Interfaz “Agregar Perfil o Herramienta”



Figura A2.2.- Interfaz “Registrar Usuario Administrativos de Perfiles o de Herramientas”