

006.3011

MAR

S

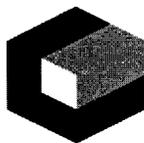
TD 0045-04-01

TD-0045-04-01

INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA

FACULTAD DE INGENIERÍA INDUSTRIAL

ESPECIALIDAD INFORMÁTICA



SICA

**Sistema Inteligente de Certificación Académica
Módulo Aplicación y Calificación de Exámenes**



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
INFORMÁTICO**

Autor:

Yulkeidi Martínez Espinosa

Tutor:

Ing. Licet Gutiérrez Mompí

Asesora:

Dra. Sc. Dámasa Martínez Martínez

Ciudad de La Habana, Junio de 2004.

Resumen

En la actualidad, la labor de los profesores a la hora de la calificación de los exámenes y las preguntas escritas (las cuales se realizan semanalmente) se hace difícil pues deben dedicarle mucho tiempo para esta tarea. Y cuando se debe medir la calidad de los estudiantes dado un criterio, el proceso se hace muy lento y trabajoso. Por ejemplo, si necesitamos escoger los mejores estudiantes para un proyecto determinado, tendríamos seguramente que recurrir al criterio de sus profesores, o a los resultados en las pruebas (que también es un número considerable); ambas variantes consumen mucho tiempo y recursos. Este proceso se hará cada vez más engorroso a medida que aumente el número de estudiantes y proyectos en los que estos deberán trabajar.

Con el propósito de realizar un sistema que garantice y facilite los procesos de evaluación y calificación se crea el Sistema Inteligente de Certificación Académica (SICA), en Universidad de las Ciencias Informática (UCI), de fácil acceso para estudiantes y profesores.

El Sistema Inteligente de Certificación Académica permitirá agilizar los procesos de confección, aplicación y revisión de los exámenes; evaluar a los estudiantes sin la intervención continua de los profesores; certificarlos académicamente en el conocimiento de una aplicación o técnica determinada y; la elaboración y calificación automática de pruebas para la auto evaluación de estudiantes.

Con ayuda del Lenguaje Unificado de Modelación (UML: Unified Modeling Language) y su herramienta por excelencia el Rational Rose, quienes se encargan de crear destrezas indispensables para crear sistemas de software bien diseñados, robustos y de fácil mantenimiento; modelamos las etapas de análisis y diseño del Sistema Inteligente de Certificación Académica (SICA), con vistas a aplicar tecnologías y lenguajes de programación orientado a objetos.

La ejecución e implantación de este proyecto tendrá un impacto social importante porque la nueva tendencia de la nuestra Universidad de las Ciencias Informáticas (UCI) y la educación en nuestro país; es avanzar cada vez más en el camino de la docencia no presencial, y borrar el mito de que la aplicación de exámenes tradicionales.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica.....	4
1.1 Introducción.....	4
1.2 Sobre la certificación.....	4
1.3 Evaluación y automatización. Las nuevas tecnologías en el área de Testing.....	5
1.4 Algo sobre los TAIs (Tests Adaptativos Informatizados)	6
1.4.1 Ventajas que poseen los TAIs o frente a los Test Estáticos.	9
1.4.2 Elementos básicos de los TAIs.....	9
1.4.2.1 Banco de ítems.	9
1.4.2.2 La selección de los ítems.....	10
1.4.2.3 Estimación del nivel de rasgo.....	11
1.4.2.4 Criterio de parada.	11
1.5 Ampliación de los TAIs: algunos modelos.....	11
1.5.1 TAIs con ítems politómicos	11
1.6 Condiciones de aplicación.....	12
1.6.1 Construcción, mantenimiento y renovación del banco de ítems	12
1.6.2 TAIs fáciles	13
1.6.3 Revisión	14
1.7 Reseña sobre los sistemas de certificación existentes	15
1.7.1 BrainBench (http://www.brainbench.com)	15
1.7.2 ExamsOnLine (http://www.examsonline.com)	15
1.7.3 NCSAcademy (http://www.ncsacademy.com).....	16
1.7.4 Análisis de la factibilidad de la alternativa.....	17
1.8 Tecnologías.....	17
1.8.1 ASP (Active Server Pages)	17
1.8.2 ASP.NET (Active Server Pages).....	17
1.8.3 ASP versus ASP.NET	18
1.8.4 Modelo Cliente-Servidor.....	19
1.8.4.1 Servidor	19
1.8.4.2 Cliente	20
1.9 Lenguajes	20
1.9.1 C#.....	20
1.9.2 SQL (Structured query language).....	20
1.9.3 UML (Unified Modeling Language) y rational rose	21
1.10 Gestor de Base de datos	22
1.10.1 SQL Server 2000.....	22
1.11 Metodologías y Herramientas	22
1.11.1 RUP (Rational Unified Process).....	22
1.11.2 Rational Rose	23
1.11.3 Visual Studio .NET	23
1.11.4 Embarcadero ERStudio 5.5	24
1.12 Conclusiones.....	24
Capítulo 2. Estudio Preliminar	26
2.1 Introducción.....	26

2.2 Objeto de estudio	26
2.2.1 Problema y situación problemática:	26
2.2.1.1 Problema:	26
2.2.1.2 Situación Problemática:	26
2.3 Objeto de automatización.	27
2.4 Información que se maneja.....	28
2.5 Propuesta de sistema	28
2.6 Modelo del negocio	29
2.6.1 Descripción de los procesos de negocio.....	29
2.6.2 Representación de los casos de uso del negocio.....	31
2.6.2.1 Actores	31
2.6.2.2 Trabajadores	31
2.6.2.3 Descripción de los Casos de Uso del Negocio	31
2.6.3 Diagramas del Modelo del Negocio.....	34
2.6.3.1 Diagrama Casos de Uso del Negocio.	34
2.6.3.2 Diagramas de Actividades del Negocio.....	35
2.6.3.3 Diagrama de Clases del Modelo de Objeto.	35
2.7 Especificación de los requisitos de software.....	35
2.7.1 Dependencias y Relaciones	35
2.7.2 Requerimientos Funcionales	36
2.7.3 Requerimientos no funcionales	38
2.8 Definición de los Casos de Uso	40
2.8.1 Definición de los actores	40
2.8.2 Listado de casos de uso	41
2.8.3 Descripción Casos de Uso.....	41
2.8.4 Diagrama de casos de uso	44
2.8.5 Casos de uso por ciclo.....	44
2.8.6 Casos de Uso expandidos.....	45
2.9 Conclusiones.....	46
<i>Capítulo 3. Análisis y Diseño del sistema.....</i>	<i>47</i>
3.1 Introducción.....	47
3.2 Análisis	47
3.2.1 Definición del modelo de análisis	47
3.2.2 Modelo de Clases de Análisis	48
3.3 Diseño	48
3.3.1 Diagramas de Interacción.....	48
3.3.2 Diagramas de Clases	48
3.3.2.1 Descripción de las clases	49
3.3.3 Diseño de la BD	49
3.3.3.1 Diagrama Entidad Relación de la BD.....	49
3.3.3.2 Descripción de las tablas	49
3.3.4 Tratamiento de errores	49
3.3.5 Seguridad	49
3.3.6 Interfaz	50
3.3.7 Concepción de la ayuda.....	51
3.4 Conclusiones.....	51
<i>Conclusiones.....</i>	<i>53</i>

<i>Recomendaciones</i>	54
<i>Referencias Bibliográficas</i>	55
<i>Bibliografía</i>	59
<i>Anexos</i>	62
Anexo 1 Diagramas del Modelo del negocio	62
Anexo 2 Diagramas de Actividades del Modelo del Negocio.	63
Anexo 3 Diagrama de Clases del Modelo de Objetos.	65
Anexo 4 Casos de Uso Expandidos	66
Anexo 5 Diagrama de Clases del Análisis	72
Anexo 6 Diagramas de Interacción.	73
Anexo 7 Diagramas de Clases	79
Anexo 8 Descripción de las Clases	84
Anexo 9 Diagrama Entidad Relación	93
Anexo 10 Descripción de las Tablas	97
<i>Glosario de Términos.</i>	102

Introducción

La creciente demanda laboral, a nivel mundial, ha traído como consecuencias un aumento del nivel de preparación de los aspirantes a los puestos y por tanto la necesidad de existencia de mecanismos que faciliten la selección de estos a gran escala.

Por otra parte el mito de la computadora trae consigo la posibilidad de explotar todas las vías de enseñanza y aprendizaje, realizando los términos de auto preparación, auto estudio y auto evaluación; aunque este último, en nuestro país, se encuentre un poco en desuso, pues necesariamente tenemos que dirigirnos a sistemas de certificación o evaluación de competencias, existentes en el mercado internacional, cuyos recursos no están al alcance de nuestras posibilidades o simplemente tenemos del desconocimiento del idioma.

La Universidad de las Ciencias Informáticas (UCI) utiliza el modo de evaluación tradicional, a través de los exámenes establecidos en el plan docente, pero si un alumno desea medir sus conocimientos en una materia que estudia de forma independiente debe remitirse a sistemas existentes que pueden o no estar en Internet y que por tanto, no son de fácil acceso.

Cuando se debe medir la calidad de los estudiantes dado un criterio, el proceso se hace muy lento y trabajoso. Por ejemplo, si necesitamos escoger los mejores estudiantes para un proyecto determinado, tendríamos seguramente que recurrir al criterio de sus profesores, o a los resultados en las pruebas (que también es un número considerable); ambas variantes consumen mucho tiempo y recursos. Este proceso se hará cada vez más engorroso a medida que aumente el número de estudiantes y proyectos en los que estos deberán trabajar. Actualmente la labor de los profesores a la hora de la calificación de los exámenes y las preguntas escritas (las cuales se realizan semanalmente) se hace difícil pues deben dedicarle mucho tiempo para esta tarea.

El problema está dado por la carencia, en la Universidad de las Ciencias Informáticas, de un sistema que permita certificar académicamente a sus estudiantes, así como agilizar los procesos de confección, aplicación y revisión de los exámenes.

Este trabajo, por tanto, se desarrolla con el objetivo general diseñar y desarrollar el Módulo Aplicación y Calificación en el Sistema Inteligente de Certificación Académica (SICA), para así proveer a la Universidad de las Ciencias Informáticas de un sistema de evaluación, de fácil acceso para estudiantes y profesores, que permita comprobar los conocimientos. Como objetivo específico, evaluar a los estudiantes sin la intervención continua de los profesores; certificarlos académicamente en el conocimiento de una aplicación o técnica determinada; y permitir la aplicación y calificación automática de pruebas para la auto evaluación.

En el tema a tratar consultamos bibliografía y sistemas de certificación académica existentes en el mundo, como <http://www.BrainBench.com>, <http://www.ExamsOnLine.com> y <http://www.NCSAcademy.com>. El estudio se enfocó hacia sus potencialidades y las ventajas y desventajas según nuestras necesidades específicas. Además consultamos las teorías sobre los Test Automatizados, Computer Adaptive Tests (CATs) como tecnología mayormente utilizada en los mejores sistemas de certificación del mundo.

Para cumplir estos objetivos se planteó como objeto de estudio los sistemas de certificación académica existentes y las nuevas tecnologías en el área de Testing.

El campo de acción es el Sistema Inteligente de Certificación Académica basado en la realización de Test Adaptativos Informatizados (TAIs) para la aplicación y calificación de exámenes, en la Universidad de las Ciencias Informáticas por parte de los profesores y los alumnos del centro.

Como aportes prácticos de este trabajo debemos destacar:

- Rapidez y eficacia en la creación, aplicación y calificación de exámenes.
- Facilidades de preparación y evaluación, para estudiantes, de manera autodidacta.
- Reconocimiento del nivel cognoscitivo de los estudiantes en una materia, dado un criterio.
- Agilización de las estadísticas generales docentes.

Las tareas definidas para dar cumplimiento a estos objetivos se detallan a continuación, las cuales se llevarán a cabo haciendo uso del Rational Rose, herramienta que se basa en el Lenguaje Unificado de Modelación (UML: Unified

Model Language) y que permite una mejor construcción, brindando como resultado un software que satisface al máximo las necesidades del cliente:

- Realizar un estudio detallado de los fundamentos teóricos de los procesos a automatizar.
- Obtener los requerimientos del trabajo.
- Determinar los casos de uso.
- Elaborar el diagrama de casos de uso.
- Expandir los casos de uso del núcleo central.
- Elaborar los diagramas de interacción y diagrama de clases.
- Diseñar la base de datos.
- Avanzar en la implementación del primer ciclo de desarrollo del sistema.

Este documento, distribuido en capítulos, ayudará y guiará en la comprensión del trabajo desarrollado para la concepción y creación del Sistema Inteligente de Certificación Académica.

En el Capítulo 1: Fundamentación Teórica, se establecen algunos presupuestos de partida sobre la certificación académica, se aclaran aspectos acerca del estado del arte del tema tratado a nivel internacional, así como sobre las tecnologías, metodologías y Softwares usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta, sobre los que es necesario profundizar.

El Capítulo 2: Características del Sistema, se describe el objeto de estudio, especificando el objeto de automatización así como la propuesta del sistema, también incluye los requerimientos, tanto funcionales como no funcionales, y los modelos del negocio y el sistema propuesto.

El Capítulo 3: Análisis y diseño del sistema, se razona sobre los aspectos internos del sistema y lo modelamos para encontrar su forma, representando a través de diagramas el flujo de procesos. Facilitando visualmente la evolución del sistema para su posterior implementación.

A continuación aparecen las conclusiones, recomendaciones, bibliografía, anexos y glosario de términos.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En el presente capítulo damos un amplio recorrido sobre las teorías de los Test Adaptativos Informatizados y las tendencias modernas de la evaluación y certificación. Además de una reseña y análisis de los productos de software existentes en el mercado, que dan solución a problemas similares, así como una pequeña descripción de las técnicas, tecnologías y lenguajes usados para el análisis, diseño e implementación en el presente trabajo.

1.2 Sobre la certificación

El calificativo certificación académica es utilizado en algunos ámbitos para nombrar a un documento que recoge los resultados cuantitativos de las evaluaciones realizadas durante un determinado período académico, en otras palabras la certificación de notas. En su lugar debería referirse a un sistema de certificación de competencias, pues está más a tono con las tendencias actuales de la formación.

Es conveniente admitir que la evaluación y la certificación de competencias laborales responde a la necesidad de introducir políticas de formación más amplias, coherentes y flexibles, a fin de encauzar, de la mejor manera posible, las grandes transformaciones sociales y económicas que resultan de una mayor vinculación de las realidades internas de un país con el mundo exterior, en el marco de sus propias necesidades y recursos. Entre los factores asociados a la esfera internacional que más inciden sobre dichas transformaciones, cabría señalar los siguientes:

- La globalización comercial.
- Los avances científicos y tecnológicos.
- Los nuevos esquemas en la reorganización del trabajo.
- El aumento en las expectativas de vida de los individuos.
- El incremento de la demanda de servicios educativos y oportunidades de empleo (aunados al crecimiento poblacional y al incremento del promedio de vida productiva).
- La diversidad y movilidad de los mercados laborales.
- La emergencia de las llamadas era de la información y sociedad del conocimiento.

En las instituciones educativas, como entre instructores y docentes, es común que las medidas de evaluación y certificación sean percibidas como una seria amenaza a intereses particulares o gremiales, más que como elementos de superación individual y colectiva. Para los funcionarios del sector educativo, la instauración de políticas de evaluación con frecuencia acarrea más problemas que buenos resultados. Por su parte, gran cantidad de padres de familia, ante el pobre desempeño de sus hijos en la escuela o la falta de recursos económicos con que apoyarles, pueden pensar que conforme aumenten las exigencias académicas, menores serán las posibilidades de formación y progreso para ellos. En la tónica de estos elementos de juicio, la calidad de la educación resulta menos importante que la accesibilidad a la educación.

Nuestro país, y en específico la Universidad de las Ciencias Informáticas (UCI) siguen las tendencias actuales de la educación y cada día se inclina más a la educación no presencial y la formación profesional, y a la certificación y evaluación mediante el uso de las nuevas tecnologías. De manera general muchas instituciones han tenido la grandiosa idea de desarrollar este tipo de sistemas, lo que a pesar de lo fructífero que sería su realización, no han sido llevadas a cabo, o han sido abandonadas en medio de su desarrollo por falta de motivación, recursos o simplemente por no contar con un buen respaldo.

1.3 Evaluación y automatización. Las nuevas tecnologías en el área de Testing

La renovación en las ciencias es consustancial a la profesión docente. Pero esta renovación no se refiere sólo a la llamada actualización científica, es decir, a la puesta al día en los conocimientos de las distintas disciplinas; se refiere también, muy particularmente, a la renovación de los métodos de enseñanza y de los procedimientos de evaluación. En este quehacer el profesor necesita, sin duda, orientaciones e instrumentos nuevos.

El ser humano desde que nace necesita de guías que le muestren el camino para desarrollar sus habilidades. Estos guías, tutores o maestros con sus enseñanzas forman o crean inquietudes en las cuales se basa nuestro desarrollo cognoscitivo. Los tutores humanos proveen una alta eficiencia en el entorno de la enseñanza debido a que son capaces de estimar el grado de entendimiento de sus alumnos sobre un tema. Esto es posible gracias a la interacción directa tutor-alumno. Con el desarrollo de la tecnología se planteó una cuestión interesante ¿Será posible

duplicar un tutor humano mediante alguna máquina? Si fuera así, la única máquina capaz de emular las labores humanas en la actualidad es la computadora. Los sistemas inteligentes de enseñanza o mejor conocidos por su nombre en inglés “Intelligent Tutoring Systems” o solo ITS, son herramientas de software que utilizan técnicas descubiertas en la inteligencia artificial para lograr la enseñanza de algún tema o habilidad. Los sistemas inteligentes de enseñanza son comúnmente referidos como CAI que son las siglas de Instrucción Asistida por Computadora.

Los sistemas inteligentes de enseñanza son verdaderamente sistemas de comunicación de conocimiento. Pueden ser definidos de esa manera debido a que el énfasis principal en el desarrollo de estos sistemas es proveerlos con acceso a la representación del conocimiento que es lo que el sistema intenta de comunicar al estudiante. En un ITS el énfasis está puesto en el conocimiento (qué) a ser comunicado al estudiante y no en el mecanismo (cómo) de comunicación usado para presentar el conocimiento al estudiante.

1.4 Algo sobre los TAIs (Tests Adaptativos Informatizados)

El uso del ordenador ha permitido la incorporación de estrategias singulares en la presentación, análisis de respuesta y puntuación de los Tests, así como avances importantes en la elaboración, construcción y administración de los mismos.

El uso de los TAIs proporciona una evaluación de habilidades o conocimientos, precisa y eficiente, así como que presenta posibilidades de aplicación en la mejora de los procesos de enseñanza y aprendizaje.

Uno de los campos en los que los TAIs han creado más expectativas es el diagnóstico de las habilidades de los estudiantes. Estas expectativas encuentran una justificación en el hecho de que algunos de los objetivos asociados a los TAIs están próximos a aquellos que se establecen en el ámbito del diagnóstico. Un TAI se construye bajo el supuesto de que la presentación de un ítem cualquiera a un sujeto es el resultado de un proceso de análisis en el que se tienen en cuenta las características del ítem y las del propio sujeto. Pues bien, el problema básico del diagnóstico que se apoya en el uso de Tests es determinar bajo qué condiciones, restricciones o reglas se elige un ítem de modo que nos permita profundizar en el proceso que sigue un individuo para afrontar una tarea. En ambos campos, el problema es un problema no sólo de contenido sino, sobre todo, de proceso: ¿cuál

es el siguiente ítem? y ¿qué características debe poseer? Un TAI que pretenda utilizarse con fines diagnósticos debería responder satisfactoriamente a esas dos cuestiones.

La posibilidad de aplicar este tipo de Tests en la evaluación de habilidades complejas, de introducirlos como instrumentos asociados a los procesos de enseñanza y aprendizaje o de utilizarlos como herramientas en los procesos diagnósticos no resulta, desde luego, un asunto baladí. Precisamente, la elaboración de criterios para determinar el valor de los TAIs en la evaluación de habilidades cognitivas y en la mejora del proceso de enseñanza y aprendizaje, es el tema que nos ocupará en la siguiente tabla:

Selección de ítems y puntuación del Test	<ul style="list-style-type: none"> • El procedimiento para la estimación de la habilidad y la selección de ítems debe ser documentado de forma explícita y detallada. • El procedimiento debería incluir un método de variación de los ítems seleccionados para evitar usar exclusivamente unos pocos ítems. • El algoritmo utilizado por el ordenador debería administrar los ítems diseñados y registrar las respuestas como operaciones independientes, sin interferir con el proceso adaptativo. • El ordenador debería ser capaz de elegir el primer ítem apoyándose en información previa.
Factores humanos	<ul style="list-style-type: none"> • El ambiente del aula de informática en el que se administre el Test debería ser tranquilo, confortable y libre de distracciones. • La disposición de la pantalla debería evitar reflejos o deslumbramientos. • Es necesario valorar empíricamente la legibilidad de la presentación en pantalla de los ítems. • La presentación debería poder incluir representaciones gráficas en las que se aprecien los detalles.

Como vemos en la tabla, no es un asunto fácil realizar un TAI y mucho menos aplicarlo a los procesos docentes educativos, su creación y aplicación exige condiciones, tanto psicológicas como físicas, adecuadas, Así como una buena selección de hardware, software y dispositivos.

Podemos considerar que los Tests Adaptativos Informatizados (TAIs) tienen ya unos 30 años de historia. La mayor parte de este tiempo ha estado en el laboratorio, donde la principal preocupación de los investigadores ha sido obtener las estrategias más eficientes, precisas y válidas posibles. Sólo en la última década se han empezado a utilizar de forma masiva. [Hotangas, 2000].

Conviene no confundir “Test Adaptativo Informatizado” con “Test Informatizado”. Un Test Informatizado es un Test tradicional en el que se utiliza un ordenador para la administración de ítems (o reactivos), recogida de respuestas, corrección del Test, emisión de la calificación. Un Test Adaptativo Informatizado (o TAI) también presenta los ítems y recoge las respuestas de la persona utilizando un ordenador, pero el uso del ordenador no es su rasgo más característico. Lo más genuino es su capacidad para adaptarse al rendimiento de la persona que está siendo medida, de ahí el “Adaptativo” de TAI. La idea básica es presentar exclusivamente a cada persona los ítems que nos informan de su nivel de rasgo. A una persona con mucho nivel de vocabulario inglés no tiene mucho sentido pedirle la traducción de “DOG”, pues es seguro que la sabrá. Análogamente, a una persona con muy bajo nivel, tampoco se le debería preguntar por la traducción de “SLED” (trineo), pues es muy poco probable que la sepa. Un TAI evitaría ambas preguntas a personas de alto y bajo nivel de vocabulario inglés, respectivamente. Una consecuencia de sólo administrar los ítems realmente útiles es que se puede conseguir la misma precisión en la medida del rasgo con muchos menos ítems.

El auge actual de los TAI se debe básicamente a dos razones: la primera es la existencia de ordenadores personales de gran capacidad a bajo coste. La segunda es la aparición de una nueva teoría de los Tests, la Teoría de la Respuesta al Ítem (TRI), que tiene claras ventajas frente a la tradicional Teoría Clásica, [Muñiz, 1997]. El principio de invarianza de la TRI permite obtener estimaciones de los niveles de rasgo en la misma escala sean cuales sean los ítem administrados. Es decir, podemos aplicar unos ítems fáciles a una persona de bajo nivel e ítems difíciles a otra de alto nivel y obtener para uno y otro sus niveles de rasgo en una misma escala, lo que permite su comparación.

1.4.1 Ventajas que poseen los TAIs o frente a los Test Estáticos

TAIs requiere menos preguntas para arribar a un estimado más exacto de la habilidad de los encuestados que los Test de papel y lápiz.

- La tecnología de los TAIs permite recibir retroalimentación acerca de su funcionamiento.
- Cada encuestado administra diferentes Sets de preguntas aumentando la seguridad del los Test.
- TAI no guarda el número de preguntas respondidas correctamente, solamente los ítems respondidos correctamente.
- Un usuario que responde correctamente un conjunto de preguntas de más dificultad reciben una mayor puntuación que el uno que responde correctamente un conjunto de preguntas más sencillas.
- El resultado obtenido es reflejo del número de preguntas respondidas adecuadamente y el nivel de dificultad de esas preguntas.

Las preguntas generalmente son de selección múltiple, son hechas para evaluar el conocimiento del mundo real contrario al conocimiento académico. Se puede esperar obtener código de ejemplo, imágenes y escenarios asociados con muchos de las preguntas. Cada pregunta tiene solo una respuesta correcta. [Hotangas, 2000].

Sus elementos básicos son cuatro: el banco de ítems, el procedimiento adaptativo, la estimación del nivel de rasgo y el criterio de parada.

1.4.2 Elementos básicos de los TAIs

1.4.2.1 Banco de ítems

Un TAI necesita de un conjunto de ítems en el que se cumplan los supuestos de la TRI (unidimensionalidad) y se conozcan sus parámetros. El banco de ítems es una base de datos donde se almacena cada ítem, sus parámetros, y la información que se pueda considerar relevante. Dos aspectos importantes de los bancos de ítems son su tamaño y su actualización. En cuanto al tamaño, el número de ítems del banco habrá de ser varias veces mayor al de ítems que el Test administre en promedio. Un segundo asunto es la actualización el banco. Con las sucesivas administraciones del Test, hay ítems que conviene eliminar y reemplazar por otros nuevos.

El banco debe tener un procedimiento de renovación de ítems y de actualización de la información sobre cada ítem, tras cada administración.

1.4.2.2 La selección de los ítems

Normalmente el TAI elige el primer ítem al azar de entre los que tienen dificultad media. A partir del primer ítem se tendrá la respuesta de la persona y si ha habido acierto o no. La TRI proporciona métodos de estimación del nivel de rasgo. Utilizando uno de los posibles métodos se llega a obtener una primera estimación provisional. Se han propuesto diversos procedimientos para seleccionar el segundo y sucesivos ítems. Uno de ellos es el basado en el principio de máxima información, que consiste en administrar de todos los ítems disponibles el más informativo para el último nivel estimado de rasgo. Es decir, se trataría de seleccionar el ítem que más ayuda a estimar con precisión el nivel de rasgo. Se han propuesto también procedimientos de selección de ítems bayesianos [van der Linden, 1998], por lo tanto, tras cada ítem, se obtiene una estimación provisional a partir de las respuestas a los ítems presentados, se selecciona el ítem, se administra, se vuelve a estimar el nivel de rasgo. La estimación tras el último ítem será de hecho la puntuación de la persona en el Test.

Uno de los problemas derivados de elegir en cada Test los mejores ítems es que si calculamos su tasa de exposición (es decir, el promedio de cuantas veces ha sido administrado un ítem en cada 100 Tests) nos encontramos que hay ítems que son administrados en más del 80% de los Tests y que hay muchos ítems del banco, en algún caso, más del 50%, que no se administran nunca.

Lo anterior plantea dos tipos de problemas: uno de validez, pues los ítems de muy alta tasa terminan siendo conocidos por los que van a hacer el Test y pueden dejar de medir lo que se supone que miden. El segundo tipo de problemas es la paradoja que supone haber elaborado un banco grande para que la mitad o más de los ítems del banco no se administren nunca. Para resolver este problema han surgido los métodos de control de la exposición, que consiguen incrementar la tasa de exposición de los ítems infrautilizados, disminuir la tasa de los utilizados en exceso, a costa de una pérdida conocida en la precisión del Test.

1.4.2.3 Estimación del nivel de rasgo

En el marco de la TRI se han propuesto varios estimadores de los niveles de rasgo y se han determinado sus características. Los TAIs aplican estos mismos estimadores, pero tienen problemas específicos, como por ejemplo, los derivados de la menor longitud del Test. Se han hecho estudios (por ejemplo, [Cheng, 2000] para determinar si las características de los estimadores (su sesgo, variabilidad...) en las condiciones ordinarias de aplicación de un TAI.

1.4.2.4 Criterio de parada

El Test puede detenerse cuando se ha administrado un cierto número de ítems, cuando se ha alcanzado una cierta precisión o cuando se alcance uno de los dos criterios anteriores.

1.5 Ampliación de los TAIs: algunos modelos

1.5.1 TAIs con ítems politómicos

En la actualidad hay una gran cantidad de modelos politómicos propuestos para diferentes tipos de ítems [van der Linden, 1996]. Una buena parte de ellos ha sido utilizada como base psicométrica para elaborar Tests Adaptativos (TAIP).

Las investigaciones se han interesado en sus ventajas frente a los TAIs con ítems dicotómicos, en comparar diferentes modelos entre sí y en analizar las propiedades de los heurísticos de selección de ítems y de los métodos de estimación de la habilidad. Los principales beneficios de los TAIPs se derivan de la mayor información que ofrecen sus ítems frente a los ítems dicotómicos. Al resultar más informativos, se pueden obtener estimaciones más precisas con menos ítems, por lo que el tamaño del banco no necesita ser tan grande, si con ello no se compromete su seguridad.

Las pruebas realizadas indican que a partir de un banco de 30 ítems puede obtenerse una precisión adecuada, sin problemas de convergencia en la estimación del rasgo [Dodd, 1989]; Dodd, 1993]; [Dodd, 1994], siempre que la localización de los ítems no se focalice en uno de los extremos de la distribución. A pesar de los buenos augurios que les vaticinan una de sus mayores limitaciones es necesitar grandes muestras para estimar los parámetros de los ítems.

Para estimar la habilidad, los primeros trabajos utilizan en su mayoría el método ML, que en este contexto presenta el problema de no poder hacer una estimación inicial cuando se eligen categorías extremas. La solución propuesta es aplicar estrategias stepsize [Koch, 1989]; [Dodd, 1990]; [Dodd, 1989]. Posteriormente, la investigación se ha centrado más en los métodos EAP y MAP [De Ayala, 1992]; [Cheng, 1997]; [Cheng, 1998]; [Hou, 1996]. Estos métodos permiten estimar cualquier patrón de respuestas y tienen menor error cuadrático medio. Los estudios anteriores señalan ciertas ventajas frente a ML, especialmente cuando se utilizan suficientes puntos de cuadratura (al menos 20) en el método EAP. Sin embargo, presentan cierto sesgo, especialmente en Tests cortos, y su bondad depende de la adecuación de la distribución a priori utilizada, aunque el sesgo es pequeño incluso con distribuciones a priori erróneas. Recientemente, [Wang, 1999] han comparado los métodos ML, EAP, MAP y MLP respecto al sesgo y error típico, entre otros aspectos, con los modelos de respuesta graduada y crédito parcial generalizado. Al parecer MLP es superior a ML y ambos presentan menor sesgo pero mayor error típico que EAP y MAP.

Las reglas de parada analizadas consisten en finalizar el Test cuando: a) los ítems sin aplicar no proporcionan información suficiente y b) la precisión alcanza un nivel adecuado o, sin alcanzarlo, la longitud del Test es la máxima permitida. El segundo criterio de parada, es más eficiente en términos de número de ítems aplicados, frecuencia de estimaciones no convergentes y precisión [Dodd, 1989]; [Dodd, 1993]; [Dodd, 1990]; [De Ayala, 1989]. Si la función de información del Test es uniforme, la diferencia entre ambas reglas se hace más pequeña [Dodd, 1989]. Ambas resultan más eficientes que fijar la longitud [De Ayala, 1992]; [Pastor, 1999].

En general, los TAIPs son más cortos porque ofrecen más información que los TAIs dicotómicos [De Ayala, 1992], pero el aumento en información no siempre es homogéneo a lo largo de todo el rango de la habilidad [De Ayala, 1989].

1.6 Condiciones de aplicación

1.6.1 Construcción, mantenimiento y renovación del banco de ítems

Seguramente los TAIs tendrían mayor mercado si no costase tanto el proceso de calibración del banco. El procedimiento más frecuente para la renovación, denominado calibración online, consiste en administrar progresivamente nuevos ítems junto a los del banco actual.

Sin embargo, el proceso de calibración de los nuevos ítems tiene dificultades porque se obtienen datos incompletos, en pocos ítems y para niveles de habilidad bastante homogéneos.

En [Ban, 2000] pusieron a prueba cinco métodos de calibración on-line, y concluyeron que el método de estimación máximo verosímil marginal con múltiples ciclos proporciona menores errores de estimación de parámetros. Este método consiste en: 1) estimar la distribución posterior de habilidad con los ítems del TAI, 2) estimar los parámetros de los nuevos ítems que maximizan la función de verosimilitud marginal, 3) usar los parámetros estimados de todos los ítems para estimar la distribución posterior de habilidad, 4) obtener una nueva estimación de los mismos dejando libres los parámetros de los nuevos ítems y 5) repetir el proceso hasta que se alcance un criterio de convergencia.

El marco de lo que se denomina generación automática de ítems [Irving, 2000] abre la posibilidad de aplicar un TAI sin necesidad de calibrar el banco. La idea es predecir la dificultad de los ítems a partir de un modelo psicológico (que explique cómo los sujetos resuelven los ítems) y de un modelo psicométrico componencial (que relacione los componentes que determinan la dificultad con la probabilidad de acierto y los niveles de habilidad). En teoría, estos procedimientos permitirían disponer del universo completo de ítems sin necesidad de elaborarlos previamente, con los consiguientes beneficios en cuanto a costes y seguridad. Sobre los inconvenientes que se van descubriendo en esta línea de trabajo puede consultarse a [Revuelta, 1999]. Otra posibilidad es la generación de ítems isomorfos: una vez calibrado un banco, se modifican ciertas características estructurales de los ítems originales manteniendo en lo posible sus propiedades psicométricas [Revuelta, 1998].

1.6.2 TAI's fáciles

Cualquier evaluando, independientemente de su nivel de habilidad, obtiene en un TAI tasas de acierto en torno al 50 ó al 60 %. Los sujetos de alta habilidad pueden tener una cierta sensación de frustración porque cometen muchos fallos, lo que desde un punto de vista motivacional (y por tanto para optimizar su rendimiento) quizás no resulte lo mejor. En [Bergstrom, 1999] se estudió la eficiencia de TAI's en los que se presentan ítems más fáciles de los que correspondería aplicar según el criterio de máxima información.

Por ejemplo, se puede incluir en el algoritmo una instrucción para que se elija en cada momento el ítem que resulta más informativo para un nivel de habilidad que resulta de restar una constante al nivel provisional estimado. Estos autores prefieren incrementar de esta forma el confort de los evaluandos aún teniendo que incrementar la longitud y el tiempo de aplicación del TAI para conseguir un nivel prefijado de precisión. En [Ponsoda, 1999] comprobó que, tanto en TAIs como en Tests Autoadaptados Informatizados (TADIs), el descenso de ansiedad va asociado a las condiciones de evaluación donde los sujetos obtenían más aciertos. Es posible, por tanto, que el grado de confortabilidad asociado a la situación de evaluación tenga algo que ver con la sensación subjetiva de éxito.

1.6.3 Revisión

Por su experiencia previa en contextos educativos, los evaluandos prefieren revisar y, en su caso, cambiar sus respuestas en Tests fijos. Cuando se permite la revisión, los evaluandos pueden reducir su ansiedad ante la evaluación, ganan control sobre la situación y consideran que el Test es más justo. En un TAI la posibilidad de revisar y cambiar respuestas puede tener efectos no deseables, como el incremento del tiempo de aplicación, la reducción de la precisión del Test o, lo que seguramente sería más importante, y el uso de estrategias de respuesta que llevan ganancias ilegítimas. Respecto a los dos primeros inconvenientes, se ha analizado empíricamente el efecto de la revisión en los aciertos, la habilidad estimada y su precisión [Olea, 2000]; [Vispoel, 1998]. Adicionalmente, se han propuesto diferentes procedimientos de revisión [Vispoel, 2000] y las consecuencias de omitir o diferir respuestas [Lunz, 1994]. En cuanto al problema de las ganancias ilegítimas, se han estudiado los efectos que tiene fallar deliberadamente ítems en la primera aplicación, para responderlos correctamente después de la revisión [Stocking, 1997]; [Vispoel, 1999]. Parece que la revisión por bloques de ítems, en lugar de al final de la aplicación, puede resolver en parte el problema de sesgo y precisión. [Kingsbury, 1996] planteó otra posible estrategia de ganancia ilegítima: si un evaluando se percata de que después de fallar un ítem se le presenta otro más fácil, podría cambiar la respuesta al primero en la revisión.

La realidad actual es que sólo uno de los TAIs operativos [Bergstrom, 1999] incorpora la posibilidad de revisar las respuestas. Deben clarificarse estos riesgos para que los beneficios motivacionales no conlleven consecuencias peores.

1.7 Reseña sobre los sistemas de certificación existentes

1.7.1 BrainBench (<http://www.brainbench.com>)

Los Test de BrainBench consisten en 40 preguntas que toman entre 45 minutos y 1 hora para completar. Cada pregunta dispone generalmente de 180 segundos mostrando una advertencia cuando restan 30 segundos. Los resultados incluyen lo siguiente:

- Una puntuación entre 1.0 y 5.0 puntos donde 5 es el máximo, acompañado del texto con la descripción del significado de la puntuación.
- Una puntuación porcentual mostrando la comparación con todos los que se hallan evaluado antes.
- Un resumen de tus fortalezas y debilidades comparadas con tu puntuación en conjunto.

Aproximadamente el 40% de los que completan un Test de certificación son certificados (alcanzan una puntuación de 2.75). De estos solo el 12% alcanzan el nivel master (4.0 puntos).

Los Test de BrainBench están desarrollados, validados y revisados por profesionales del área que ya han sido certificados. Todos los Test han sido desarrollados usando el certificado ISO 9001. Las pruebas son enviadas mediante el Computer Adaptive Tests (CATs).

BrainBench proporciona también un programa de certificación de roles laborales. Una completa certificación laboral incluye una selección de un núcleo de habilidades así como también habilidades electivas que se pueden seleccionar de acuerdo con las necesidades específicas. Mientras una certificación de habilidad pueda probar sus conocimientos sobre una habilidad específica, una certificación laboral prueba que tiene el núcleo de habilidades requeridas para desempeñar un trabajo.

1.7.2 ExamsOnLine (<http://www.examsonline.com>)

Los exámenes gratis constan de 15 preguntas y los pagados de 105.

Brinda 3 opciones para la realización del examen:

Simulación de examen aleatorio: Las preguntas se muestran en un orden aleatorio con un límite de tiempo de 20 minutos. Permite simular un examen real, las respuestas no son mostradas durante la realización del examen.

Examen completo: Todas las preguntas son mostradas, no hay límite de tiempo. Puedes ver las respuestas durante la realización del examen.

Examen personalizado: No hay límite de tiempo. Se puede modificar la pregunta inicial y la final. Este modo se utiliza para ver las preguntas en un rango determinado. Las preguntas no son aleatorias. Se podrán ver las mismas preguntas cada vez que se seleccione el mismo rango de preguntas. Puedes ver las respuestas durante la realización del examen.

Todas las opciones permiten una revisión al finalizar el examen.

Permite bajar exámenes.

1.7.3 NCSAcademy (<http://www.ncsacademy.com>)

Características de los exámenes:

El examen consta de 35 a 50 preguntas en dependencia del tema a evaluar, tomándose hasta cerca de una hora para contestarlas todas. Todas las preguntas son de selección múltiple.

La puntuación es en base a 4.0 puntos. Una puntuación superior a los 2.0 puntos indica que se ha superado el examen y es elegible para la certificación. Una puntuación inferior no te permite ser certificado. Se deben esperar 7 días para repetir un examen.

Se tienen 90 segundos para contestar cada pregunta. El sistema proporciona un contador que muestra el tiempo restante para responder la pregunta. El tiempo también es controlado por el servidor y cuando termina no hay posibilidades de cambiar la respuesta dada.

Ofrece una evaluación por temáticas permitiendo conocer donde se encuentran los problemas.

1.7.4 Análisis de la factibilidad de la alternativa

Dados los sistemas de certificación académica existentes en el mundo y el estudio de sus principales características encontramos que los mismos se encuentran fundamentalmente en idioma inglés, no existe código fuente ni datos de diseño y no hay un ajuste a las necesidades que han surgido UCI por lo tanto se decidió comenzar el desarrollo de un nuevo sistema que respondiera a las necesidades del proceso docente productivo que es está llevando a cabo en la Universidad de las Ciencias Informáticas.

La propuesta viene siendo como la fusión de estos sitios, que recoge lo mejor de los sistemas de certificación existentes hasta este momento y se caracteriza principalmente por las posibilidades que brinda de adaptarse al individuo que se examina en ese momento. Y provee mejoras como:

- Interfase visual y agradable.
- Trabajo en el idioma del país.
- No necesidad de buscar otros Sistemas de evaluación de competencias o certificación académica en Internet.

1.8 Tecnologías

1.8.1 ASP (Active Server Pages)

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

1.8.2 ASP.NET (Active Server Pages)

La tecnología ASP.NET es la nueva apuesta de Microsoft para la creación de páginas Web dinámicas. Sustituye a su predecesor ASP en el intento de desbancar a PHP como lenguaje estrella para la creación de sitios dinámicos y además promete

dar a los creadores de sitios y aplicaciones Web un nivel de control y flexibilidad sin precedentes al momento de desarrollar sus sitios.

Las páginas ASP permiten mezclar las etiquetas HTML con código como Java o VBScript. Bajo este esquema cuando un usuario accede a las páginas HTML o ASP puede disparar la ejecución de código que se encuentra en páginas ASP. Esta da respuesta al cliente como una página HTML. Una ventaja del código del lado del servidor en las páginas ASP es la seguridad ya que el usuario no accede al código. Una desventaja es el desempeño ya que el código se compila en el momento de ser accedido.

ASP.NET es la tecnología que permite hacer formularios Web y Servicios Web. El formulario Web es una página dinámica que puede acceder a los recursos del servidor o a otros servidores. Por ejemplo, una página ASP.NET puede ejecutar código para acceder a SQL Server y luego armar una respuesta al usuario como código HTML. Como el código se ejecuta del lado del servidor, la respuesta puede adaptarse al Explorador de Internet del usuario.

Como ASP.NET esta construido sobre el .NET Framework, su código puede ser escrito en cualquier lenguaje compatible con .NET.

1.8.3 ASP versus ASP.NET

Tras el análisis de las dos tecnologías de Microsoft, se puede comprobar como ASP tiene un digno sucesor, pero hay que darse cuenta que no es simplemente una actualización de la tecnología para desarrollo de aplicaciones Web, sino que es un nuevo enfoque de computo para el desarrollo de software.

Mientras ASP se escribía en VBScript, ASP.net puede ser escrito en cualquier lenguaje soportado por el .net Framework, es decir: VB.net; C# y JScript.net. Otro cambio radical es que ASP.net es un lenguaje totalmente orientado a objetos.

Sin duda, es mucho mejor que el ASP tradicional, ASP.net trae diversas mejoras entre las cuales se destacan:

Rendimiento: la aplicación se compila en una sola vez al lenguaje nativo, y luego, en cada petición tiene una compilación Just In Time, es decir, se compila desde el

código nativo, lo que permite mucho mejor rendimiento. También permite el almacenamiento del caché en el servidor

Rapidez en programación: mediante diversos controles, podemos con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.

Servicios Web: trae herramientas para compartir datos e información entre distintos sitios.

Seguridad: tiene diversas herramientas que garantizan la seguridad de nuestras aplicaciones.

1.8.4 Modelo Cliente-Servidor

La tecnología denominada Cliente-Servidor es utilizada por todas las aplicaciones de Internet / Intranet:

- Un servidor es un ordenador remoto -- en algún lugar de la red -- que proporciona información según petición.
- Un cliente funciona en su ordenador local, se comunica con el servidor remoto, y pide a éste información.
- El servidor envía la información solicitada.
- Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

1.8.4.1 Servidor

En el lado servidor, toda la aplicación utiliza un patrón de diseño conocido como MVC (por sus siglas en inglés, Model-View-Controller, o modelo-vista-controlador). Este esquema permite separar la presentación -lo que el usuario "ve"-, el modelo de interacción -lo que el usuario "hace" y cómo responde el sistema-, y la representación lógica de los datos presentados. Para construir el lado servidor de la capa de aplicación se utilizan tecnologías que permiten generar dinámicamente el contenido que será enviado al cliente, que puede ser HTML, WML o XML.

1.8.4.2 Cliente

El software cliente recibe los modelos de presentación desde el servidor y los presenta al usuario, para que éste pueda manipular la información que hay en el sistema y tomar acciones sobre la misma. Los aspectos críticos del lado cliente de la capa de presentación son la facilidad de uso, la versatilidad y la velocidad de la interfaz.

1.9 Lenguajes

1.9.1 C#

Visual Studio .NET 2003 incluye abundantes mejoras en Visual Basic y Visual C++, así como un nuevo lenguaje de programación, Visual C#, lenguaje de propósito general diseñado por Microsoft para su plataforma .NET (leído en inglés "C Sharp" y en español "C Almohadilla"). Aunque la plataforma .NET admite muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET.

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con la creación del nuevo lenguaje, es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.

Un lenguaje que hubiese sido ideal utilizar para estos menesteres es Java, pero debido a problemas con la empresa creadora del mismo -Sun-, Microsoft ha tenido que desarrollar un nuevo lenguaje que añadiese a las ya probadas virtudes de Java las modificaciones que Microsoft tenía pensado añadirle para mejorarlo aún más y hacerlo un lenguaje orientado al desarrollo de componentes.

1.9.2 SQL (Structured query language)

El SQL es un lenguaje de consulta estructurado, surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar internacional de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan.

El SQL trabaja con estructura cliente/servidor sobre una red de ordenadores. El ordenador cliente es el que inicia la consulta; el ordenador servidor es que atiende esa consulta. El cliente utiliza toda su capacidad de proceso para trabajar; se limita a solicitar datos al ordenador servidor, sin depender para nada más del exterior. Estas peticiones y las respuestas son transferencias de textos que cada ordenador cliente se encarga de sacar por pantalla, presentar en informes tabulados, imprimir, guardar, etc., dejando el servidor libre.

El SQL permite:

- Definir una base de datos mediante tablas.
- Almacenar información en tablas.
- Seleccionar la información que sea necesaria de la base de datos.
- Realizar cambios en la información y estructura de los datos.
- Combinar y calcular datos para conseguir la información necesaria.

En otras palabras, SQL es el lenguaje de comunicación entre el programa cliente y programa servidor.

1.9.3 UML (Unified Modeling Language) y rational rose

Es un lenguaje de propósito general para el modelado Orientado a Objetos. Y es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes. En un principio existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelamiento de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos, es decir, UML pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML (como lenguaje de modelado visual) se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software y además para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Se encarga de captar la información sobre la estructura estática y el comportamiento dinámico de un sistema.

Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

Es necesario esclarecer que UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

1.10 Gestor de Base de datos

1.10.1 SQL Server 2000

SQL Server 2000 es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con una interfase intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET.

Microsoft SQL Server 2000 es la última versión del sistema de gestión de bases de datos relacionales (SGBDR) que aprovecha la sólida base establecida por su predecesor SQL Server 6.5. Y 7. Como la mejor base de datos para Windows NT, SQL Server 2000 es el ideal para un amplio espectro de clientes corporativos y fabricantes independientes de software (ISV). Las necesidades y requisitos del cliente han dado lugar a innovaciones significativas en SQL Server versión 2000, entre las que se incluyen la facilidad de uso, escalabilidad y fiabilidad, y almacenamiento de datos.

Los usuarios acceden a la información que está en el servidor a través de una aplicación. Los administradores acceden al servidor directamente para realizar tareas de configuración, administrativas y de mantenimiento de la base de datos.

1.11 Metodologías y Herramientas

1.11.1 RUP (Rational Unified Process)

Es un producto desarrollado y mantenido por Rational, actualizado constantemente para obtener las mejores prácticas de acuerdo con la experiencia, quien describe como utilizar de forma efectiva procedimientos comerciales, probados en el desarrollo de software para equipos de desarrollo de software y constituye una guía de cómo usar UML de forma efectiva.

RUP es un proceso de desarrollo de software, plantea la forma disciplinada de asignar las tareas y responsabilidades en una empresa de desarrollo (quien hace qué, cuándo y cómo). Su principal objetivo como metodología es: asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Está dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).

RUP permite aumentar la productividad de los desarrolladores mediante acceso a base de conocimiento, plantillas y herramientas y se centra en la producción y mantenimiento de los modelos, más que en producir documentos.

1.11.2 Rational Rose

Rational Rose es la herramienta CASE desarrollada por los creadores de UML, que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y 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 utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Por su parte Racional Rose nos proporciona innegables ventajas: permite separar el desarrollo de la vida del software en etapas predefinidas como análisis, diseño e implementación y con esto facilita una mejor comprensión del sistema en su etapa de desarrollo y construcción.

1.11.3 Visual Studio .NET

Visual Studio .NET es la herramienta completa para crear e integrar rápidamente servicios Web XML, aplicaciones de Microsoft Windows y soluciones Web. Este entorno ofrece las herramientas necesarias para diseñar, desarrollar y mantener servicios Web de una forma mucho más sencilla de la que tendríamos al programarlos desde cero.

El desarrollo y puesta en producción de servicios Web en la plataforma Microsoft .NET no tiene porqué pasar obligatoriamente por el uso de Visual Studio .NET. Con un simple editor de textos podemos construir toda la programación y ficheros de configuración necesarios para la puesta en explotación de un servicio Web "real", pero sin duda la utilización de este moderno entorno de desarrollo nos brinda muchas facilidades a la hora de desarrollar nuestros servicios Web.

1.11.4 Embarcadero ERStudio 5.5

Erwin es una herramienta CASE para modelar, que ayuda a diseñar bases de datos de alto desempeño para cliente/servidor y web/intranet. Erwin no solo ayuda a diseñar modelos de datos lógicos, también construye automáticamente estructuras de datos físicos con la información del diagrama. Permite visualizar la estructura, elementos clave y optimizar el diseño de las bases de datos, genera tablas u otras especificaciones en dependencia de la plataforma seleccionada. Tiene como ventajas:

- Facilidades de diseño de diagramas Entidad-Relación y Entidad-Relación extendido y transformación de este al modelo relacional (en tercera forma normal, preservando las dependencias funcionales y sin pérdidas de información).
- Comparación comprensiva entre el modelo de datos y la base de datos.
- Soporta la separación del modelo lógico y del físico.

1.12 Conclusiones

En este capítulo se abordaron los temas más actuales sobre la certificación académica, para una mejor comprensión de los conceptos manejados y de los objetivos que nos trazamos, así como los temas más actuales tratados en el mercado en cuanto a tecnologías, metodologías, herramientas y lenguajes usados para realizar el análisis y el diseño en el presente trabajo.

Lo más factible según la investigación realizada es realizar una aplicación Web que responda a las necesidades del proceso docente productivo que es está llevando a cabo en la UCI con una interfase agradable y en idioma español e incorporando lo mejor de los sistemas de certificación existentes. Usar para la implementación la tecnología ASP. NET, debido a las ventajas que presenta con su antecesor ASP, en cuanto a lenguaje de programación utilizado (una variedad significativa), rapidez en

programación, ampliación con servicios Web y seguridad que aportan sus herramientas a las aplicaciones. Como lenguaje C#, quien toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo, además de ser más sencillo e intuitivo, y el único diseñado específicamente para la plataforma .NET.

Capítulo 2. Estudio Preliminar

2.1 Introducción

El presente capítulo se centra en la investigación del problema, no en la manera de definir la solución. Para crear una aplicación de software hay que describir el problema, el entorno de dicho problema y las necesidades o requerimientos: en qué consiste el conflicto y qué debe hacerse. [Larman, 1999].

Se realiza una descripción del problema y su ubicación; así como el entorno en que se desarrollan los procesos. También se hace un estudio detallado del modelo del negocio y se plantean los requerimientos y casos de uso del sistema.

2.2 Objeto de estudio

2.2.1 Problema y situación problemática:

2.2.1.1 Problema

El problema está dado por la carencia, en la Universidad de las Ciencias Informáticas, de un sistema que permita certificar académicamente a sus estudiantes así como agilizar los procesos de confección, aplicación y revisión de los exámenes.

2.2.1.2 Situación Problemática

En la Universidad de las Ciencias Informáticas los estudiantes son evaluados a través de los exámenes de su programa docente; pero si un alumno desea medir sus conocimientos en una materia que estudia de forma independiente debe remitirse a sistemas existentes que pueden o no, estar en Internet y que por tanto no son de fácil acceso. Lo mismo pasa en el caso de los profesores.

Por otra parte, cuando se debe medir la calidad de los estudiantes dado un criterio, el proceso se hace muy lento y trabajoso. Por ejemplo, si necesitamos escoger los mejores estudiantes para un proyecto determinado, tendríamos seguramente que recurrir al criterio de sus profesores, o a los resultados en las pruebas (que también es un número considerable); ambas variantes consumen mucho tiempo y recursos. Este proceso se hará cada vez más engorroso a medida que aumente el número de estudiantes y proyectos en los que estos deberán trabajar.

Actualmente, al aplicar un examen, debe hacerse con la participación simultánea y obligatoria de los alumnos y los profesores en el aula, la labor de de estos últimos, a la hora de la calificación de los exámenes y las preguntas escritas (las cuales se realizan semanalmente) se hace difícil pues deben dedicarle mucho tiempo para esta tarea.

2.3 Objeto de automatización

Con la propuesta se pretende poner en explotación un sistema que brinde una forma fácil, didáctica y tecnológicamente avanzada de aplicación y calificación de exámenes.

Para ello se debe crear una infraestructura que permita ilimitadas ventajas a los profesores en la actividad docente evaluativa y los estudiantes como parte de la auto evaluación. Y por encima de todo, hacer honor a uno de los objetivos de La Universidad de las Ciencias Informáticas, la formación profesional de forma no presencial, facilitando la certificación académica en determinadas áreas del conocimiento.

Entre las fórmulas a aplicar tenemos:

- Aplicar la clave de calificación para obtener la calificación del estudiante y del equivalente de la calificación en base a 5 puntos.
- Tiempo promedio de respuesta, suma de los tiempos de respuesta de cada pregunta dividido entre la cantidad de preguntas.
- Cantidad de preguntas respondidas correctamente. Sumar la cantidad de preguntas con el máximo de puntuación.

Debemos obtener además un resumen, que le será de mucha utilidad a los usuarios y que estarán dados principalmente por sus fortalezas y debilidades a la hora de realizar el examen, este resumen se elabora a partir de los objetivos, cumplidos o no, en cada pregunta del examen.

Además, en la Universidad de las Ciencias Informáticas (UCI) existe un Sistema de Gestión Académica, al cual en ciertas y determinadas circunstancias sería necesario exportarle las calificaciones de los exámenes realizados, con los datos pertinentes al usuario.

2.4 Información que se maneja

La información que se maneja son preguntas y exámenes, que deben garantizar redacción y coherencia. En cuanto a las respuestas a las preguntas deben recepcionarse de forma segura e individual, evitando la propagación de estas a manera de fraude.

El tiempo de respuesta del sistema desde el momento en que se retira una pregunta y se genera la próxima no debe ser mayor de 10 segundos ni menor que este tiempo, facilitando así que el usuario se tome un pequeño reposo entre una pregunta y la siguiente.

El sistema contará con dos grandes grupos de datos, a los que llamamos bancos de preguntas y respuestas, debido a la gran cantidad de información que se almacena en los mismos y el nivel de importancia; cada uno de estos datos mantendrá el derecho del propietario, evitando que algunas preguntas, de uso restringido, sean visibles cuando no deben serlo, guardando seguridad e integridad de los datos

2.5 Propuesta de sistema

El sistema contará con una base de datos con un servicio para usuarios con tres niveles de acceso;

- Administradores del sitio.
- Profesores y/o Profesores Responsables.
- Usuarios y/o Alumnos Registrados.

Además contará con un conjunto de exámenes previamente confeccionados.

El usuario a examinarse solamente debe autenticarse en el sitio o página principal y luego de acceder al listado de exámenes (dirigidos, dinámicos o estáticos), a los cuales tiene acceso en dependencia de sus características cognitivas, y seleccionar uno. El sistema mostrará el examen correspondiente a la selección del usuario y generará las preguntas correspondientes. Se irá chequeando constantemente el tiempo de muestreo, ya sea de una pregunta o del examen en total. Cumplido el tiempo establecido se retira la pregunta y se procede a la calificación de la misma. El proceso se repetirá hasta que se agoten todas las preguntas del examen.

Aunque también existe la posibilidad de que el examen sea dinámico, quien para generar y calificar las preguntas utilizará un modelo matemático (CAT), muy utilizado en exámenes psíquico-pedagógicos, donde el examen puede adaptarse a las características del usuario y en dependencia de ciertos parámetros de cumplimiento es que se genera la próxima pregunta.

En el análisis comparativo con otras propuestas, como <http://www.BrainBench.com>, <http://www.ExamsOnLine.com> y <http://www.NCSAcademy.com> mejoramos aspectos como:

- Interfase visual y agradable.
- Trabajo en el idioma del país.
- No necesidad de buscar otros Sistemas de evaluación de competencias o certificación académica en Internet.

La propuesta viene siendo como la fusión de estos sitios, que recoge lo mejor de los sistemas de certificación existentes hasta este momento y se caracteriza principalmente por las posibilidades que brinda de adaptarse al individuo que se examina en ese momento.

2.6 Modelo del negocio

Por muy pequeño o sencillo que parezca un sistema, debe dividirse en piezas o partes para su mejor estudio y comprensión. Esta división recibe el nombre de modelo del negocio.

Estas piezas tienen que cumplir con determinadas propiedades: ser coherentes y estar relacionadas, de manera que permita representar de forma abstracta las características esenciales del sistema.

2.6.1 Descripción de los procesos de negocio

En la Universidad de las Ciencias Informáticas los procesos de evaluación, o de aplicación y calificación de exámenes transcurren de manera tradicional. Los profesores imparten sus asignaturas y en el cierre de cada período docente, deben evaluar el estado académico y cognoscitivo de sus estudiantes, para los cuales aplican exámenes, que son elaborados y respondidos mediante de lápiz y papel y en el mejor de los casos, algunos profesores, los más jóvenes o los vinculados al

desarrollo de las nuevas tecnologías utilizan los recursos informáticos para sus comprobaciones. Estos test aplicados en las computadoras generalmente tiene las mismas características que los tradicionales, un examen que cuenta de 2 a 5 preguntas, planificadas a ser respondidas en un tiempo de 50 a 110 min., en muchos casos las mismas para todo el alumnado y en el menor de ellos con dos o tres baterías (examen con aproximadamente el mismo nivel de complejidad pero con distintas preguntas).

Al término de este tiempo los exámenes son recogidos para ser calificados en un plazo correspondiente a los cinco días hábiles después de su aplicación. Para la calificación de cada examen se elabora una clave de calificación con el fin de hacer más certeros los juicios acerca del progreso de cada estudiante, distribuyendo la puntuación de las preguntas en aquellos objetivos más importantes y donde el estudiante debió hacer más hincapié.

Con ayuda de esta clave se determina la calificación que obtuvo el estudiante en el examen y se les entregan los resultados, luego se pasa a un periodo de revisión, donde los estudiantes insatisfechos con su calificación pueden pedir revisión, aquí el profesor recalifica por si cometió algún error le da los resultados a los alumnos y archiva la nota en el registro de calificaciones.

¿Qué trae consigo esto este proceso? Pérdida de tiempo del profesor a la hora de las calificaciones y recalificaciones; y en cuanto al proceso de seleccionar a un grupo de estudiantes con ciertas características, solo se tiene a medir por el criterio del profesor o los resultados de las pruebas, proceso que se hace lento y trabajoso y aun más si se incrementa el número de estudiantes y proyectos en los que estos deberán trabajar.

Este proceso de aplicación y calificación de exámenes se automatizará, esto no quiere decir que dejarán de existir los exámenes tradicionales, sino simplemente será una herramienta más con la que pueden contar los profesores, el cual fermentará el estudio y la auto evaluación personal, puesto que mucho se acomodan en espera de los exámenes y no buscan otras formas de comprobar sus conocimientos.

2.6.2 Representación de los casos de uso del negocio

2.6.2.1 Actores

La definición de un actor es: rol que alguien o algo juega cuando interactúa con el negocio para beneficiarse de sus resultados.

Actores del negocio:	Justificación
Jefe de Disciplina.	<p>Se encarga de determinar si una clave de calificación es adecuada para un examen determinado.</p> <p>Recibe información sobre los resultados docentes del los alumnos, y basándose en ellas, toma decisiones y las registra.</p> <p>Ejerce como profesor (por lo que cumple además las mismas funciones).</p>
Alumno	Es evaluado por el profesor a través de exámenes.

2.6.2.2 Trabajadores

Abstracción de una persona, máquina o sistema automatizado que actúa en el negocio realizando una o varias actividades, manipulando entidades del negocio.

Trabajadores del negocio	Justificación
Profesor	<p>Es quien aplica y califica los exámenes.</p> <p>Lleva un registro del las calificaciones obtenidas por un grupo de estudiantes en estas evaluaciones.</p> <p>Se encarga de determinar la clave de calificación.</p>

2.6.2.3 Descripción de los Casos de Uso del Negocio

Describen el flujo de trabajo asociado a un Caso de Uso del Negocio (CUN).

Nombre del caso de uso del negocio:	Realizar examen.	
Actores del negocio:	Alumno	
Propósito:	Comprobar los conocimientos docentes.	
Resumen:	<p>El caso de uso se inicia cuando el alumno recibe el examen de mano del profesor, porque este se dispone a comprobar sus conocimientos. La prueba consta de un conjunto de pregunta a los cuales los alumnos deben responder en un tiempo predeterminado, el cual es controlado por el profesor. El caso de uso finaliza cuando el profesor recoge los exámenes.</p>	
Casos de uso asociados:	-	
Flujo de trabajo		
Acción del actor	Respuesta del negocio	
2. Los alumnos reciben el examen y comienzan a responder.	<p>1. El profesor entrega los exámenes y empieza a controlar el tiempo que debe durar este.</p> <p>3. Agotado este tiempo recoge los exámenes.</p>	
Prioridad:	-	
Mejoras:	Este proceso se realizará de forma automática, el control del tiempo lo llevará el sistema.	
Cursos alternos:	-	

Nombre del caso de uso del negocio:	Calificar.
Actores del negocio:	Alumno
Propósito:	Determinar los resultados de un examen.
Resumen:	<p>El caso de uso se inicia cuando ya el profesor tiene los exámenes recogidos y procede a la calificación de estos. El caso de uso finaliza con la notificación de la</p>

nota al alumno, así como el almacenamiento de estas en el registro.	
Casos de uso asociados:	-
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:
<p>4. El Alumno recibe una nota por su examen.</p> <p>6. Decide si esta de acuerdo o no.</p>	<p>1. El profesor procede a la calificación según la clave seleccionada y a su valoración.</p> <p>2. Determina la nota para cada Alumno.</p> <p>3. Se dispone a comunicarle los resultados a los Alumnos.</p> <p>7. Registra las notas en el registro de calificaciones.</p>
Prioridad:	-
Mejoras:	Este proceso se realizará de forma automática, y en dependencia de si el examen es estático o dinámico, se podrá asociar al caso de uso Examinar, donde a medida que se examina se califique la pregunta.
Cursos alternos: -	
Acción del actor:	Respuesta del negocio:
6.1 Si no está de acuerdo, la primera vez pide revisión.	<p>6.2 El profesor recalifica.</p> <p>6.3 Determina nota final y se la informa a los alumnos.</p>

Nombre del caso de uso del negocio:	Determinar reglas de calificación.
Actores del negocio:	Jefe de la disciplina
Propósito:	Definir las claves de calificación.
Resumen:	
El caso de uso se inicia cuando se decide aplicar un examen y el profesor le presenta una propuesta de clave de calificación al jefe de la disciplina. El caso de uso finaliza cuando el Jefe de la disciplina revisa la propuesta, la aprueba si cree que es apropiada y la almacena para ser utilizada en el examen.	
Casos de uso asociados:	-
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:
3. El Jefe de la disciplina revisa la propuesta y si es aceptada la archiva.	<ol style="list-style-type: none"> 1. El profesor se reúne con otros profesores y determinan una clave. 2. Se le presenta la propuesta al Jefe de la disciplina para que la apruebe.
Prioridad:	La clave debe estar determinada antes de proceder a examinar y a calificar.
Mejoras:	Se almacenarán distintas claves de calificación en una Base de Datos y en caso que el examen sea estático entonces se elegirá la adecuada para dicho examen.
Cursos alternos: -	

2.6.3 Diagramas del Modelo del Negocio

2.6.3.1 Diagrama Casos de Uso del Negocio

Diagrama que representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. [Jacobson, 2000].

VER Diagrama de Casos de Uso del Negocio en ANEXO 1

2.6.3.2 Diagramas de Actividades del Negocio

Describen el flujo de trabajo asociado a un Caso de Uso del Negocio, donde muestra a través de las calles las responsabilidades de los trabajadores del negocio y a través del flujo de objetos cómo se utilizan las entidades del negocio. [Jacobson, 2000].

VER Diagramas de Actividades en ANEXO 2

2.6.3.3 Diagrama de Clases del Modelo de Objeto

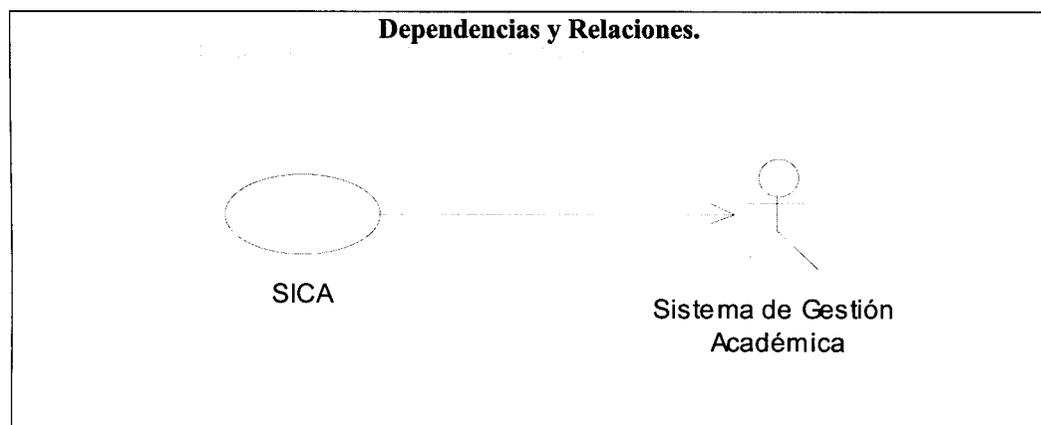
Muestra la relación que se establece entre los trabajadores y entidades del negocio. [Jacobson, 2000].

VER Diagrama de Clases del Modelo de Objeto en ANEXO 3

2.7 Especificación de los requisitos de software

2.7.1 Dependencias y Relaciones

El sistema SICA es independiente, aunque muy útil en la Universidad de las Ciencias Informáticas (UCI), tendrá interacción con el Sistema de Gestión Académica, a quien le exportará los datos de los exámenes realizados siempre y cuando se desee. Para así, de alguna forma vincular nuestro sistema directamente a la docencia y al sistema de evaluación.



2.7.2 Requerimientos Funcionales

- 1 Actualizar Modelo Matemático de Calificación. Existen 3 tipos de actualización de una regla de calificación.
 - 1.1 Se crea o añade una regla de evaluación o clave, en la base de datos.
 - 1.1.1 La regla de evaluación puede ser registrada por un profesor o un profesor responsable, quien sería el propietario.
 - 1.1.2 La regla o clave posee los siguientes datos:
 - 1.1.2.1 Nombre del Modelo, identificador.
 - 1.1.2.2 Valor mínimo y máximo del rango en las que se encuentra cierta calificación
 - 1.1.2.3 Calificación para ese rango y equivalente de valor si este es diferente de un valor numérico en base 5 puntos.
 - 1.2 Una vez añadida la regla puede ser modificada.
 - 1.2.1 Para ser modificada la regla solo necesita introducir el código de la regla a modificar y los datos cambiantes.
 - 1.3 Una regla puede ser eliminada
 - 1.3.1. Solo puede ser eliminada por un profesor o profesor responsable.
 - 1.3.2. Solamente necesita el código de la regla.
- 2 Iniciar Examen en 1 materia y tipo de examen, es cuando el estudiante desea comprobar sus conocimientos o un profesor haya dispuesto que ciertos estudiantes pertenecientes a un grupo deban examinarse en una materia y tipo de examen.
 - 2.1 El usuario se autentifica y selecciona el examen a realizar dentro de los disponibles.
 - 2.2 El examen puede ser estático:
 - 2.2.1 Si es estático ya han sido seleccionadas sus características con anterioridad, es decir, cumple determinados objetivos docentes o funcionales.
 - 2.2.1.1. El examen puede ser dirigido, es decir programado para que se visualice a una hora o fecha determinada.
 - 2.2.2 El examen se importa del banco de exámenes, no es más que hacer presencia física del examen.
 - 2.3 El examen puede ser dinámico:
 - 2.3.1 Si es dinámico automáticamente el sistema haciendo uso del modelo matemático (CAT) se comienzan a generar las preguntas para que los

estudiantes puedan comenzar a realizarlos y estas preguntas se irán adaptando a las características cognitivas del usuario que se va a examinar.

3 Calificar, es el acto de evaluar el examen realizado por el alumno.

3.1 El tiempo del examen es controlado.

3.1.1 Mantener control del tiempo de evaluación

3.1.2 Quitar las preguntas cuando esta se ha dado como respondida. Una pregunta se da como respondida cuando el usuario la termina o cuando ha concluido el tiempo de muestreo.

3.1.3 Mostrar mensajes de aviso para advertir sobre el consumo del tiempo de respuesta.

3.2 El acto de calificar las preguntas:

3.2.1 Calificar de forma automática las preguntas no abiertas:

3.2.1.1 Se almacena el resultado de la respuesta a la pregunta y se genera la próxima pregunta. Y así sucesivamente hasta que se termine el examen.

3.2.2 Calificar de forma manual las preguntas abiertas:

3.2.2.1 Se califican con calificadores preestablecidos, que puede ser el profesor.

3.3 Cerrar Calificación, se hace cuando se ha terminado la calificación del examen, es decir, ya todas las preguntas del examen han sido evaluadas.

3.3.1 Se emite la calificación final del examen.

4 Generar Reporte

4.1 En el reporte van implícitos:

4.1.1 La calificación obtenida por los estudiantes

4.1.2 Rango en que se encuentra comparado con otros resultados, no es más que mostrar todas las calificaciones obtenidas por los estudiantes que han realizado pruebas de ese tipo, es decir de esa materia y tipo de examen.

4.1.3 Tiempo promedio de respuesta, no es más que el tiempo en que se demora en realizar el examen y el tiempo promedio que demora en cada pregunta.

4.1.4 Cantidad de preguntas respondidas correctamente, suma de las preguntas con el máximo en puntuación.

4.1.5 Mostrar el rendimiento global para el objetivo, describir cuales son los objetivos que domino correctamente en el examen

4.1.6 Fortaleza y debilidades en base a objetivo basado en las preguntas contestadas erróneamente decir los objetivos en los q esta flojo, no ha vencido debe mejorar, y basadas en las que respondió correctamente decir cual es su fuerte o que contenido domina.

2.7.3 Requerimientos no funcionales

Apariencia o interfaz externa

- Interactivo y simple de usar.
- Profesional o tipo ejecutivo.
- Autoritario, para que los usuarios se sientan confiados.
- La interfaz debe ser personalizada para los niveles de usuario correspondientes.

Usabilidad

- Atractivo para el usuario, garantizando su asiduidad al sistema.
- Facilidad para ser comprendido.

Rendimiento

- El sistema debe ser estable, con un procesamiento de datos que satisfaga los tiempos de respuestas de las aplicaciones; este tiempo debe ser corto (debe oscilar entre 0 y 5 segundos) y el sistema deberá permanecer disponible en todo momento soportando al menos 100 usuarios simultáneamente.

Soporte

- Fácil instalación.

Portabilidad

- Facilidad para adaptarlo a diferentes ambientes sin utilizar otros medios que los previstos.
- El diseño de la base de datos será multiplataforma.

Seguridad

- Se debe implementar varios niveles de usuarios, con permisos que correspondan con el rol que desempeñan en la aplicación.
- Desarrollar una aplicación IDS (Intrusion Detection System) para alertar a los administradores del sistema sobre intrusos reales o tentativas de ataques.

Políticos-culturales

- En este Software no influyen directamente la política y la cultura, pero no se debe entrar en conflicto con los sistemas educativos extranjeros.

Legales

- Debe cumplir con las normas y leyes establecidas en el país.

Confiabilidad

- El sistema debe estar disponible 24 horas al día.
- Evaluar mecanismos de tolerancias a faltas.
- Predicción de fallos.
- Protección contra los fallos.
- Recuperación de fallos.

Interfaz interna

- Debe estar soportado por una red de 100 Mbps a 1 Gbps de velocidad.

Ayuda y documentación en línea

- Documentación de usuario, material de entrenamiento.

Software

- Microsoft Windows NT 4.0, Windows 9X and Win2000

Hardware

- Compatibles con procesadores X.486 o superior con 64 RAM de memoria.
- Un mínimo de 200 Mb de espacio libre en el disco.
- Un display compatible-SVGA (256 o más colores) con una resolución de 800 X 600 pixels.

Restricciones en el diseño y la implementación

- El sistema se implantará inicialmente en SQL Server y después, en dependencia de los resultados que arroje el estudio, se mantendrá en este gestor de base de datos o se transportará para Oracle. El diseño de la base de datos será multiplataforma garantizando la portabilidad del sistema.
- Para el análisis y el diseño del sistema debe ser utilizado el RUP (Racional Unified Process)

2.8 Definición de los Casos de Uso

Son los encargados de establecer un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

Y según [Jacobson, 2000] “Artefacto narrativo que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario”.

2.8.1 Definición de los actores

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema. [Jacobson, 2000]. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema. Y tienen características tales como:

Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Actores	
Profesor	Realiza un seguimiento de los resultados docentes de los alumnos. Se encarga de elaborar el modelo matemático de calificación conjuntamente con el Profesor Responsable.
Profesor Responsable	Operaciones que realiza un profesor normal. Debe haber como mínimo un profesor responsable en cada materia.
Usuario	Son todos aquellos usuarios que podrán acceder a los exámenes públicos.
Alumno Registrado	Es un usuario que dispondrá de un perfil personalizado en el sistema
Reloj Contador del	Llevará el control del tiempo de evaluación.

sistema	
Calificador Automático	Se encarga de calificar las preguntas no abiertas del examen, ya sean mediante un modelo matemático predefinido.
Calificador Humano	Se encarga de calificar las preguntas abiertas del examen.

2.8.2 Listado de casos de uso

1. Gestionar Modelo Matemático de Calificación.
2. Iniciar Examen
3. Realizar Examen Estático.
4. Realizar Examen Dinámico.
5. Calificar Pregunta.
6. Cerrar Calificación.
7. Controlar tiempo.
8. Generar Reporte.

2.8.3 Descripción Casos de Uso

En cada tabla se describe el funcionamiento de cada caso de uso, desde como se inicia hasta como termina, es decir, recoge los elementos que clarifican al modelo de casos de uso.

CU1	Gestionar Modelo Matemático de Calificación.
Actor	Profesor (inicia), Profesor Responsable
Descripción	Es cuando el profesor desea o debe actualizar la clave de clave de calificación. Estas actualizaciones pueden ser, crear, modificar o eliminar.
Referencia	R1

CU2	Iniciar Examen
Actor	Usuario
Descripción	El Usuario se autentifica, y el sistema le muestra los exámenes disponibles que pueda realizar, este selecciona uno. El sistema importa el examen del banco de examen.
Referencia	R2.1

CU3	Realizar Examen Estático
Actor	Usuario(inicia)
Descripción	<p>El usuario se dispone a comenzar el examen, entonces el sistema inicializa el Reloj Contador del sistema en 0.</p> <p>El sistema genera y presenta pregunta al usuario luego de chequear si el examen es dirigido o no.</p>
Referencia	R2.2

CU4	Realizar Examen Dinámico
Actor	Usuario(inicia)
Descripción	<p>El usuario se dispone a comenzar el examen.</p> <p>El sistema inicializa el Reloj Contador del sistema en 0 y genera pregunta según modelo matemático (CAT).</p>
Referencia	R2.3

CU5	Calificar Examen
Actor	Calificador Automático
Descripción	<p>El sistema detecta que se da como respondida una pregunta y chequea si la pregunta es abierta o cerrada.</p> <p>Si la pregunta es cerrada procede a Calificar pregunta, en caso contrario, almacena la respuesta y el ítems de la pregunta en espera que el Calificador Humano la califique.</p>
Referencia	R3.2.1

CU6	Cerrar Calificación
Actor	Calificador Automático, Calificador Humano

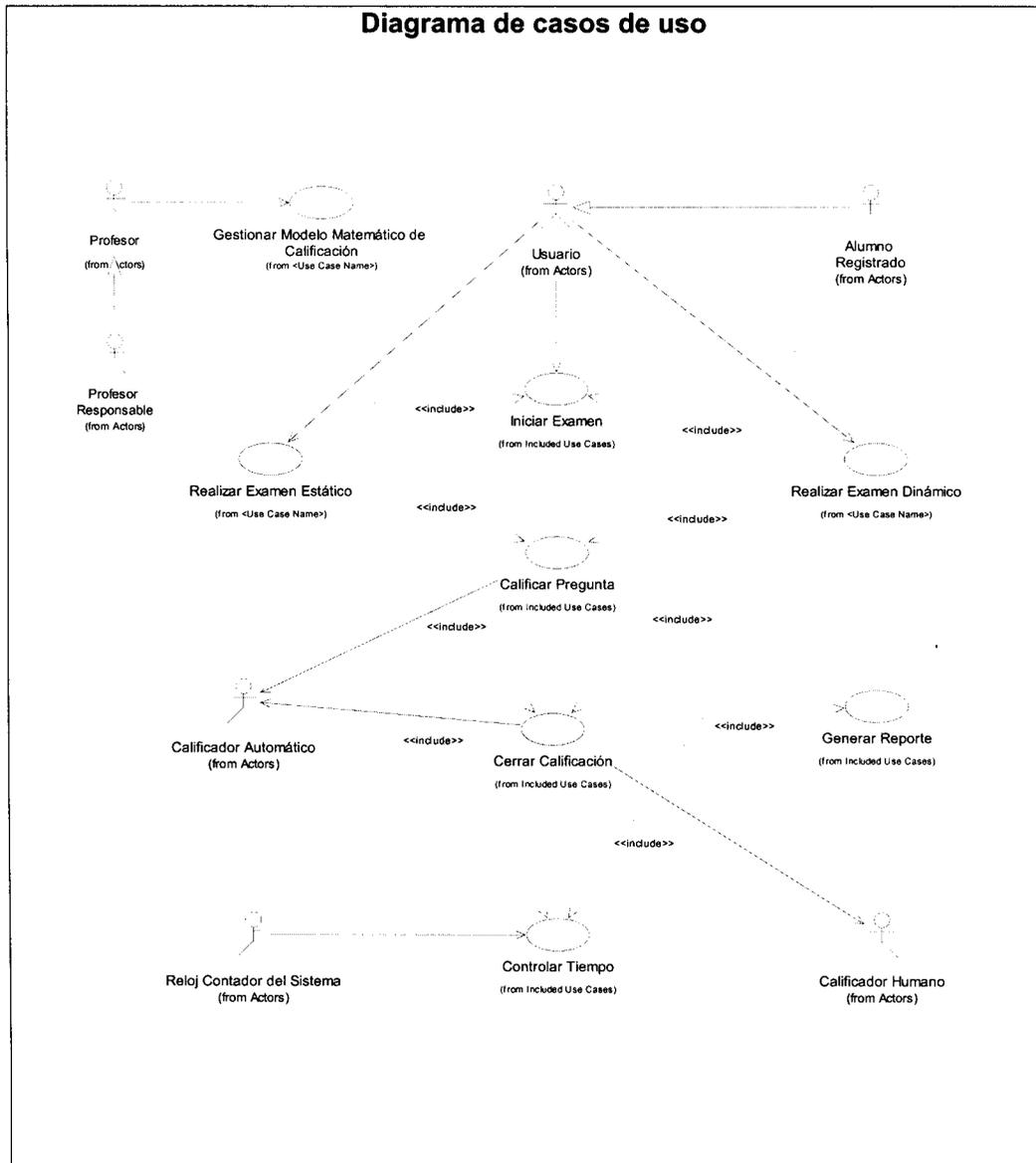
Descripción	El sistema detecta el tiempo de examen esta agotado, entonces chequea que no existan preguntas sin calificar. De existir espera que el Calificador Humano las califique, si no existen preguntas sin calificar, entonces el Calificador Automático del sistema cierra la calificación y almacena las calificaciones de los estudiantes en cada uno de sus perfiles si son alumnos registrados.
Referencia	R3.2.2 y R3.3

CU7	Controlar tiempo
Actor	Reloj contador de sistema(inicia)
Descripción	El Reloj controlará el tiempo de examen y el tiempo de cada pregunta si fueron establecidos. En caso de que el Reloj detecte que uno de estos tiempos se están agotando entonces el sistema envía un mensaje para advertir sobre el consumo del tiempo de respuesta, si el tiempo se agotó entonces el sistema retira la pregunta o el examen.
Referencia	R3.1

CU8	Generar Reporte
Actor	Usuario
Descripción	El Usuario desea conocer su calificación, entonces el sistema hace una búsqueda en la base de datos y visualiza la calificación y el resumen de sus fortalezas y debilidades que tuvo el usuario en el examen.
Referencia	R4

2.8.4 Diagrama de casos de uso

Un diagrama de casos de uso representa gráficamente a los procesos y su interacción con los actores.



2.8.5 Casos de uso por ciclo

A continuación se muestra la distribución de los casos de uso por ciclo de vida del proyecto.

Ciclo 1 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU2	Iniciar Examen.	Estos casos de uso forma el núcleo central de la aplicación. Sin ellos no tiene sentido hacer la aplicación. Se puede aplicar un Test Informatizado y calificarlo.
CU3	Realizar Examen Estático.	
CU5	Calificar Pregunta.	
CU6	Cerrar Calificación.	
CU7	Controlar tiempo.	

Ciclo 2 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU1	Gestionar Modelo Matemático de Calificación.	Estos casos de uso no son imprescindibles a la hora de realizar y calificar un examen.
CU8	Generar Reporte.	

Ciclo 3 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU4	Realizar Examen Dinámico.	Este caso de uso pasa al tercer ciclo porque aun no tenemos los conocimientos necesarios como para ponerlos en práctica.

2.8.6 Casos de Uso expandidos

Los casos de uso expandidos son muy útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos. Constituyen un documento narrativo que describe la secuencia de eventos de un actor que utiliza el sistema para completar un proceso. [Larman, 1999].

VER Casos de Uso Expandidos en ANEXO 4

2.9 Conclusiones

En este capítulo quedaron definidos los procesos del negocio, logrando un mayor entendimiento de cómo ocurre la aplicación y calificación de los exámenes en la Universidad de las Ciencias informáticas (UCI).

Además se obtuvieron los requerimientos funcionales y no funcionales de la aplicación SICA, Módulo Examinar y Calificar, a partir del modelo del negocio obtenido. Posteriormente, y partiendo de los requisitos del sistema fueron presentados los casos de uso y sus relaciones con los actores. Culminando con la obtención de el ciclo de desarrollo de la aplicación.

Capítulo 3. Análisis y Diseño del sistema

3.1 Introducción

Para desarrollar una aplicación, también es necesario contar con descripciones detalladas y de alto nivel de la solución lógica y saber cómo se satisfacen los requerimientos y las restricciones. El diseño pone de relieve la solución lógica: cómo el sistema cumple con los requerimientos. [Larman, 1999]

En este capítulo abordamos las etapas de análisis y diseño, importante en la obtención de destrezas para la creación y mantenimiento exitoso del sistema, donde se ponen de manifiesto a través del modelo conceptual y los diagramas de interacción y clases.

3.2 Análisis

Para crear una aplicación de software hay que describir el problema y las necesidades investigación o requerimientos: en que consiste el conflicto y que debe hacerse. El Análisis se centra en una investigación del problema, no en la manera de definir una solución. [Larman, 1999] Por ejemplo, si se desea un nuevo sistema de información computarizada de una biblioteca, debemos preguntarnos cuales procesos de la institución se relacionan con su uso.

En esta etapa de análisis podremos razonar más sobre los aspectos internos del sistema. También podremos usar un lenguaje más formal para apuntar detalles relativos a los requisitos del sistema.

3.2.1 Definición del modelo de análisis

El lenguaje que utilizamos en el modelo de análisis se basa en un modelo de objetos conceptual, que llamamos modelo de análisis. El modelo de análisis ayuda a refinar los requisitos y nos permite razonar sobre los aspectos internos del sistema, incluido sus recursos compartidos internos. De hecho un recurso interno, puede modelarse como un objeto en el modelo de análisis.

Además el modelo de análisis nos ofrece un mayor poder expresivo y una mayor formalización, como por ejemplo la que nos proporcionan los diagramas de interacción que se utilizan para describir los aspectos dinámicos del sistema. [Jacobson, 2000]

En otras palabras el modelo de análisis puede considerarse como una primera aproximación del modelo de diseño aunque es un modelo por si mismo. Mediante la conservación de la estructura del modelo de análisis durante el diseño, obtenemos un sistema que debería ser también mantenible como un todo: será flexible en los cambios en los requisitos, e incluir elementos que podrán ser reutilizados cuando se construyan sistemas parecidos.

3.2.2 Modelo de Clases de Análisis

El paso esencial de un análisis orientado a objetos es descomponer el problema en conceptos u objetos individuales. El modelo conceptual es la representación gráfica de los mismos. [Larman, 1999]

VER Modelo de Clases del análisis o Modelo conceptual en ANEXO 5

3.3 Diseño

En la etapa de diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos. [Larman, 1999]

3.3.1 Diagramas de Interacción

Los diagramas de interacción explican gráficamente cómo los objetos interactúan a través de mensajes para realizar las tareas. [Larman, 1999]

VER Diagramas de Interacción en ANEXO 6

3.3.2 Diagramas de Clases

El diagrama de clases describe gráficamente las especificaciones de las clases de software [Larman, 1999]. Contiene la siguiente información:

- Clases, asociaciones y atributos.
- Métodos.
- Información sobre los tipos de atributos.
- Navegabilidad.
- Dependencias.

VER Diagrama de Clases en ANEXO 7, el mismo ha sido separado en cuatro partes.

3.3.2.1 Descripción de las clases

En la descripción de las clases se nombra y especifica el tipo a la que pertenecen (Entidad, Interfaz o Controladora), así como se mencionan los atributos y las operaciones que tienen bajo su responsabilidad., cada uno con la correspondiente explicación.

VER Descripción de las clases en ANEXO 8

3.3.3 Diseño de la BD

3.3.3.1 Diagrama Entidad Relación de la BD

El Diagrama Entidad Relación (también conocido como DER, o diagrama E-R) es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema, en otras palabras es el encargado de mostrar las relaciones entre las diferentes entidades dentro de una aplicación.

VER Diagrama Entidad Relación en ANEXO 9, el mismo ha sido separado en cuatro partes.

3.3.3.2 Descripción de las tablas

La descripción de las tablas consiste en nombrar y explicar cada una, mencionando cada uno de los atributos y explicando en que consiste cada uno de ellos.

VER Descripción de las tablas en ANEXO 10

3.3.4 Tratamiento de errores

El tratamiento de errores es vital en la aplicación, de su enfoque y corrección depende el éxito y el buen funcionamiento del sistema.

Algunos de los errores más frecuentes que podemos encontrar son los generados por el Gestor de Base de Datos, pero estos se capturan internamente por el sistema y son tratados de forma que lleguen al usuario en un lenguaje de fácil interpretación.

3.3.5 Seguridad

No existe una definición estricta de lo que se entiende por seguridad informática, puesto que ésta abarca múltiples y muy diversas áreas relacionadas con los

sistemas informáticos. Áreas que van desde la protección física del ordenador como componentes hardware, de su entorno, hasta la protección de la información que contiene o de las redes que lo comunican con el exterior.

Tampoco es único el objetivo de la seguridad. Son muy diversos tipos de amenazas contra los que debemos protegernos. Desde amenazas físicas, como los cortes eléctricos, hasta errores no intencionados de los usuarios, pasando por los virus informáticos o el robo, destrucción o modificación de la información. No obstante, sí hay tres aspectos fundamentales que definen la seguridad informática: la confidencialidad, la integridad y la disponibilidad.

La confidencialidad, a veces denominada secreto o privacidad, se refiere a la capacidad del sistema para evitar que personas no autorizadas puedan acceder a la información almacenada en él.

Se entiende por integridad el servicio de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado. El sistema no debe modificar o corromper la información que almacene, o permitir que alguien no autorizado lo haga. Esta propiedad permite asegurar que no se ha falseado la información.

Un sistema seguro debe mantener la información disponible para los usuarios. Disponibilidad significa que el sistema, tanto hardware como software, se mantiene funcionando eficientemente y que es capaz de recuperarse rápidamente en caso de fallo.

Para ello la protección del sistema se hará mediante usuario y contraseña. Para entrar en los servidores será necesaria la autenticación del dominio con su correspondiente validación en el servidor. Y para la aplicación también hará falta usuario y contraseña. En los casos que se estime conveniente por el mismo tema de la seguridad, los servidores no estarán conectados a la red, sino solo con las máquinas (Servidores) que sean necesarias.

3.3.6 Interfaz

La interfaz debe ser diseñada de forma amigable, legible e interactiva; que capte la atención del usuario y garantice su estancia en el sistema.

Esto se logra teniendo en cuenta la estructura y organización de los elementos que utilizamos en la pantalla como:

- No de usaran colores fuertes ni brillantes.
- Equilibrio de la información con los componentes del mismo.
- La ubicación de los componentes e información que se repita en las páginas estén ubicados en el mismo lugar y no varíen de tamaño, forma ni color.
- Aprovechamiento y optimización del espacio libre.
- El formato de letra será estándar y de querer resaltar un texto respecto a otro se usará la negrita.

Y en cuanto al lenguaje de comunicación y/o vocabulario utilizado en el sistema resultará familiar a los usuarios pues serán términos de materias conocidas, y en caso de que no sea así, se contará con un glosario de términos, con sus correspondientes explicaciones.

3.3.7 Concepción de la ayuda.

La ayuda del sistema se realizará de forma contextual y relacionada con la parte del sistema que se este ejecutando en ese momento, de manera que el usuario pueda hacer uso de ella en cualquier momento. Además se dará la posibilidad de acceder a los temas de ayuda como una opción más del sistema, y así escogería el tema que le interesa.

3.4 Conclusiones

Con desarrollo de este capítulo se obtuvo el diagrama de clases del análisis, donde se plantearon los conceptos más significativos en el dominio del problema planteado. Este diagrama esta formado por clases preliminares, que se refinaron y perfeccionaron hasta obtener un diagrama de clases, diagrama de clases Web, que se representa por estereotipos como: Server Page, Client Page y HTML Form (páginas servidoras, páginas clientes y formularios), entre otras. Además de las relaciones y asociaciones entre ellas, navegabilidad, roles y multiplicidad.

Quedaron definidos diagramas de interacción, quienes son unos de los artefactos más importantes y a su construcción se dedica la mayor parte de la etapa de diseño. La función principal de estos es que permite asignar responsabilidades a las clases.

Capítulo 3

Además, fue elaborado el Diagrama Entidad Relación, modelo que facilita la generación y creación de la Base de Datos del Sistema. Así como las descripciones de las clases y tablas, de los diagramas de Clases y Diagrama Entidad Relación, respectivamente.

Conclusiones

Del análisis de los resultados de esta investigación se plantean las conclusiones siguientes:

- Los sistemas de certificación académica existentes no se ajustan a las necesidades que han surgido en la UCI por que se decidió desarrollar un nuevo sistema que recoge lo mejor de los sistemas analizados.
- Lo más factible según la investigación realizada es realizar una aplicación Web usando tecnología ASP. NET y con C# como lenguaje.
- Las necesidades de los clientes fueron identificadas a través de la definición de los requerimientos.
- Fueron identificados ocho casos de uso, cinco de los cuales reflejados en el primer ciclo de desarrollo. En la correspondiente descripción y expansión de los casos de uso, se detallan los procesos y especificaciones de cada uno.
- Se realizó el análisis y diseño del primer ciclo de desarrollo, mostrando a través de los diagramas de clases del análisis y diseño, y los de interacción; los conceptos más significativos en el dominio del problema así como el comportamiento interno del sistema.
- Se dispone una Base de Datos capaz de adaptarse fácilmente a las posteriores versiones del sistema

Con el diseño e implementación del módulo Aplicación y Calificación de exámenes en el Sistema Inteligente de Certificación Académica se provee a la Universidad de las Ciencias Informáticas de un sistema de evaluación que permite evaluar a los estudiantes sin la intervención continua de los profesores y aplicar y calificar test para la auto evaluación.

Recomendaciones

- Hacer un estudio más profundo de la seguridad en las bases de datos en todos los niveles que existen para evitar el mal uso de los recursos a la hora de la aplicación y calificación de los exámenes.
- Profundizar acerca de la administración del sistema, principalmente referente a los permisos de escritura/lectura.
- Garantizar la seguridad del banco de ítems frente a los intentos de apropiación ilegítima de su contenido.
- Completar los ciclos de desarrollo de la aplicación, para darle al sistema toda la funcionalidad requerida.
- Elaborar la ayuda en línea.
- Probar SICA en cualquier centro de enseñanza, con el fin de explotar sus potencialidades y contribuir de esa forma a la revolución en el área de la enseñanza y aprendizaje, así como las formas más modernas de evaluación, que nuestro país esta llevando a cabo.

Referencias Bibliográficas

- [Ban, 2000]** Ban, J., Hanson, B.A., Wang, T., Yi, Q. y Harris, D.J. A comparative study of online pretest item calibration/scaling methods in CAT. *Annual meeting of the AERA*, New Orleans. 2000.
- [Bergstrom, 1999]** Bergstrom, B. y Lunz, M. CAT for certification and licensure. En F. Drasgow y J. B. Olson-Buchanan (eds.), *Innovations in computerized assessment*. Mahwah, NJ: LEA. 1999.
- [Cheng, 1997]** Chen, S.-Y, Hou, L., Fitzpatrick, S.J. y Dodd, B.G. The effect of population distribution and method of theta estimation on CAT using the rating scale model. *Educational and Psychological Measurement*, 1997, 57, 422-439.
- [Cheng, 1998]** Chen, S.-Y, Hou, L. y Dodd, B.G. A comparison of maximum likelihood estimation and expected a posteriori estimation in CAT using the partial credit model. *Educational and Psychological Measurement*, 1998, 58, 569-595.
- [Cheng, 2000]** Cheng, P. E. y Liou, M. Estimation of trait level in computerized adaptive testing. *Applied Psychological Measurement*, 24, 257-265. 2000.
- [De Ayala,1989]** De Ayala, R.J. A comparison of the nominal response model and the three parameter logistic model in computerized adaptive testing. *Educational and Psychological Measurement*, 1989, 49, 789-805.
- [De Ayala,1992]** De Ayala, R.J. The nominal response model in computerized adaptive testing. *Applied Psychological Measurement*, 1992, 16, 327-343.
- [Dodd, 1989]** Dodd, B.G., Koch, W.R. y De Ayala, R.J. Operational characteristics of adaptive testing procedures using the graded response model. *Applied Psychological Measurement*, 1989, 13, 129-143.

- [Dodd, 1990]** Dodd, B.G. The effect of item selection procedures and stepsize on computerized adaptive attitude measurement using the rating scale model. *Applied Psychological Measurement*, 1990, 14, 355-366.
- [Dodd, 1993]** Dodd, B.G., Koch, W.R. y De Ayala, R.J. Computerized adaptive testing procedures using the partial credit model: effects of item pool characteristics and different stopping rules. *Educational and Psychological Measurement*. 1993, 53, 61-77.
- [Dodd, 1994]** Dodd, B.G. y De Ayala, R.J. Item information as a function of the threshold values in the rating scale model. En M.Wilson (ed.), *Objective measurement: Theory into practice* (vol. 2). Norwood, NJ: Ablex. 1994.
- [Hotangas, 2000]** Hontangas, P.M, Ponsoda, V., Olea, J. y Abad, F. Los tests adaptativos informatizados en la frontera del siglo XXI: una revisión. *Metodología de las Ciencias del Comportamiento*, 2000, .2 (2), 183-216.
- [Hou, 1996]** Hou, L., Chen, S. , Dodd, B. G. y Fitzpatrick, S.J. The effects of methods of theta estimation, prior distribution, and number of quadrature points on CAT using the graded response model. *Annual meeting of the AERA*, New York. 1996.
- [Irving, 2000]** Irvine, S. y Kyllonen, P. *Item generation for tests development*. Hillsdale, NJ: LEA. 2000.
- [Jacobson, 2000]** I. Jacobson, G. Booch, J, Rumbaugh *El Proceso Unificado de desarrollo de Software*, Primera edición, PEARSON EDUCACIÓN, S.A. Madrid, 2000.
- [Kingsbury, 1996]** Kingsbury, G.G.. Item review and adaptive testing. *Annual meeting of the NCME*, New York. 1996.
- [Koch, 1989]** Koch, W.R. y Dodd, B.G. An investigation of procedures for computerized adaptive testing using partial credit scoring. *Applied Measurement in Education*, 1989, 2, 335-357.

- [Larman, 1999] Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Primera edición, Prentice Hall Hispanoamericana, México, 1999.
- [Lunz, 1994] Lunz, M.A. y Bergstrom, B.A. An empirical study of computerized adaptive test administration conditions. *Journal of Educational Measurement*, 1994, 31, 251-263.
- [Muñiz, 1997] Muñiz, J. *Introducción a la Teoría de Respuesta a los Ítems*. Madrid: Pirámide, 1997.
- [Olea, 2000] Olea, J.; Revuelta, J.; Ximénez, M.C. y Abad, F.J. Psychometric and psychological effects of review on computerized fixed and adaptive tests. *Psicológica*, 2000, 21, 157-173.
- [Pastor, 1999] Pastor, D.A., Chiang, C., Dodd, B.G. y Yockey, R.D. Performance of the Sympon-Hetter exposure control algorithm with a polytomous item bank. *Annual meeting of the AERA*, Montreal. 1999.
- [Ponsoda, 1999] Ponsoda, V., Olea, J., Rodríguez, M. S. & Revuelta, J. The effects of test difficulty manipulation in computerized adaptive testing and self-adapted testing. *Applied Measurement in Education*, 1999, 12, 167-184.
- [Revuelta, 1998] Revuelta, J., Bejar, I.I. y Stocking, M.L. Computerized Adaptive Testing with item isomorphs: effects on the error of the tests scores (Research report), Princeton, NJ: ETS. 1998.
- [Revuelta, 1999] Revuelta, J. y Ponsoda, V. (1999). Generación automática de ítems. En J. Olea, V. Ponsoda y G. Prieto (eds.), *Tests informatizados: Fundamentos y aplicaciones*. Madrid: Pirámide.
- [Stocking, 1997] Stocking, M.L. Revising item responses in computerized adaptive tests: a comparison of three models. *Applied Psychological Measurement*, 1997, 21, 129-142.
- [Van der Linden, 1998] Van der Linden, W.J. Bayesian item selection criteria for adaptive testing. *Psychometrika*, 1998, 63, 201-216.

- [Vispoel, 1998]** Vispoel, W.P. Reviewing and changing answers on computer-adaptive and self-adapted vocabulary tests. *Journal of Educational Measurement*, 1998, 35, 328-345.
- [Vispoel, 1999]** Vispoel, Rocklin, Wang y Bleiler, Psychometric and psychological effects of review on computerized fixed and adaptive tests, 1999.
- [Vispoel, 2000]** Vispoel, W.P., Hendrickson, A.B. y Bleiler, T. Limiting answer review and change on computerized adaptive vocabulary tests: Psychometric and attitudinal results. *Journal of Educational Measurement*, 2000, 37, 21-38.
- [Wainer, 2000b]** Wainer, H. Rescuing computerized testing by breaking Zipf's law. *Journal of Educational and Behavioral Statistics*, 2000, 25, 203-224.
- [Wang, 1999]** Wang, S. y Wang, T. Precision of Warm's weighted likelihood estimation of ability for a polytomous model in CAT. *Annual meeting of the ERA*, Montreal. 1999.

Bibliografía

[Boggs, 2000] Boggs Wendy, Boggs Michel. UML with Rational Rose 2000
<http://biblioteca.uci.cu/bives/libroddig/pdf/0782140173.pdf> [consulta: 5/05/2004]

[Ortiz, 2001] Ortín Ibáñez, María José. *Un proceso basado en UML para aplicaciones Web (I y II)* ppt, Universidad de Murcia Julio 2001.

[Ponsoda, 2001] Ponsoda, V., Olea, J., Hontangas, P., Revuelta, J. Ponencia presentada en el XXVIII Congreso Iberoamericano de Psicología. Santiago de Chile, 29 de julio a 3 de agosto, 2001.

González Morenón, Oscar. Introducción a Web Services con herramientas de desarrollo Microsoft [en línea] Artículos técnicos Grupo Danysoft:
<http://www.danysoft.com/free/webserv.pdf> Equipo Grupo Danysoft, abril de 2002.

González Saco, José Antonio. El lenguaje de programación C# [en línea]
http://www.programacion.com/tutorial/csharp/2/#_Toc521044951 1999-2004
 Programación en castellano. [hecho: 25/07/2002] [consulta: 15/05/2004].

Guerrero, Luis A. Universidad de Chile, Departamento de Ciencias de la Computación CC61J-Taller de UML
<http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/clase2.ppt> [consulta: 13/05/2004].

Padilla, Franco Jávitt Hígmarr Nahitt; Lara Rogriguez, Amado y Marquez Gutierrez, Pedro Rafael. Sistema Inteligente para la Enseñanza de las Matemáticas. Instituto Tecnológico de Chihuahua
http://www.depi.itchihuahua.edu.mx/electro/electro2001/mem2001/articulos/cm_p6.pdf

Sánchez Medina, Edmundo. Sistema de Evaluación en Línea en el Centro Universitario de la Costa de la Universidad de Guadalajara.
<http://www.somece.org.mx/memorias/2000/docs/521.DOC> [consulta: 04/03/2004].

Tamayo Taype, Miguel Ángel; Climént Bonilla, Juan B. La Evaluación y Certificación de Competencias Laborales en el Contexto de América Latina. Algunas Reflexiones en el caso de México. Documentos presentados en el II Foro Iberoamericano sobre Formación y Empleo. Evaluación y Certificación de Competencias Profesionales. Río de Janeiro, Brasil, 12-14 de julio de 1999. <http://www.oei.org.co/iberfop/brasil2.htm> [consulta: 04/03/2004]

Concepto de base de Datos [en línea] CURSO 95/96 Universidad de Navarra, Introducción a SQL y ORACLE <http://www.unav.es/cti/manuales/Intro SQL/intro SQL.html> [consulta: 21/9/95].

La Evaluación de Test adaptativos Informatizados [en línea] http://www.uv.es/RELIEVE/v4n2/RELIEVEv4n2_6.htm Revista Electrónica de Investigación y Evaluación Educativa. 1998, Volumen 4, Número 2_6. ISSN 1134-4032. D.L. SE-1138-94.

Inc. Site designed and hosted by NetAblaze.com [en línea] <http://www.examsonline.com> 2000-2004. [consulta: 04/03/2004]

Asociación Española de Internet. [en línea] Ciberaula -c/ Villalobos, 135. - 28018 Madrid, España. <http://ciberaula.com>, 2002.

National Computer Science Academy [en línea] <http://www.ncsacademy.com> 2003.[consulta: 04/03/2004]

Introducción a la tecnología .NET [en línea] <http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/dotnet/default.msp> 2004 Microsoft Corporation. Windows Server System. [consulta 26/05/2004]

Brainbench is the leading global provider of online skills, aptitude, and knowledge measurement solutions. [en línea] <http://www.brainbench.com> [consulta: 13/03/2004].

Cd-Kit de Recursos ASP.NET 16-Ene-2003 Aplicaciones Spanishare...Comparte y Disfruta

Bibliografía

Migrando a ADO.NET [en línea]

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art88.asp>

MSDN Online en Español, 2004. [consulta: 15/05/2004].

El sitio para programadores ASP [en línea] <http://www.aspfacil.com/> 2000- 2004

Programación en castellano [en línea]

http://www.programacion.com/asp/articulo/aspnet_quees/ 1999-2004 [consulta:

13/05/2004].

Web Services. [en línea] Taller: Computación Distribuida: [http://web-](http://web-services.bankhacker.com)

[services.bankhacker.com](http://web-services.bankhacker.com) [consulta: 26/05/2004].

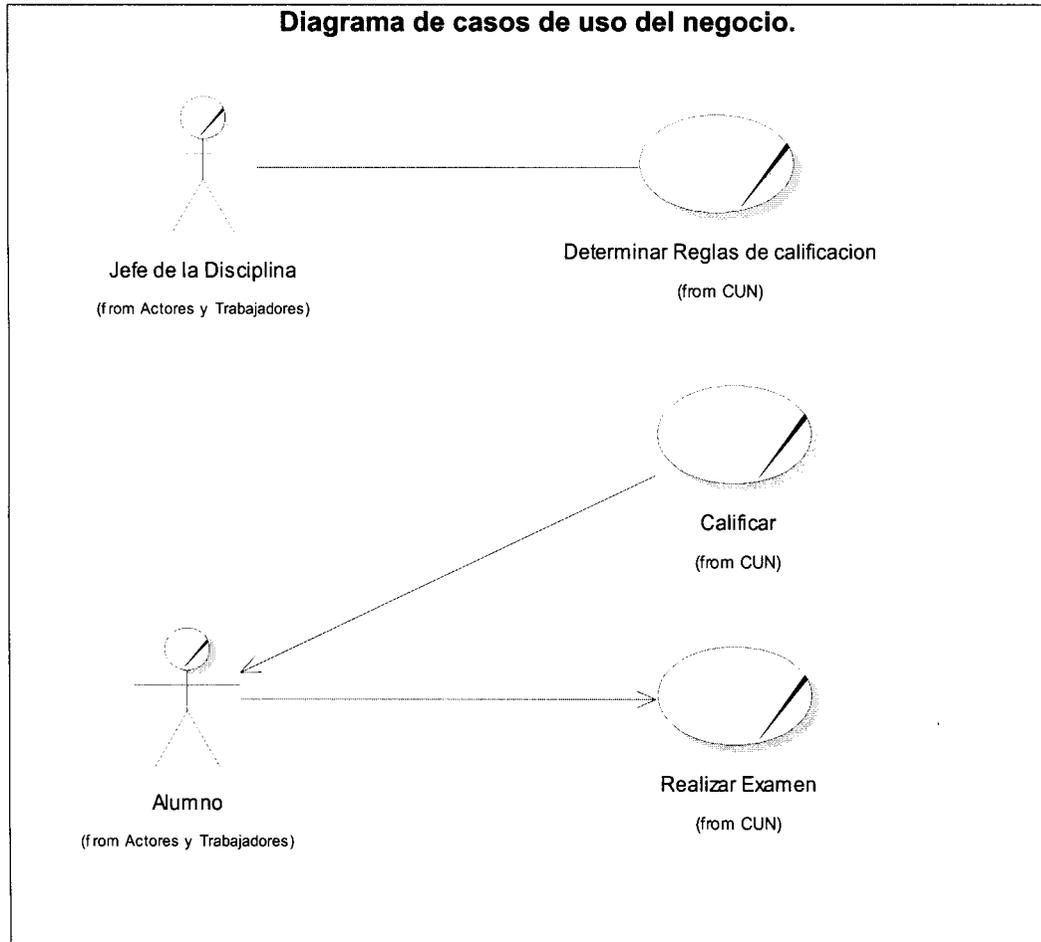
Programming Web Services with Perl, es un buen libro sobre la implementación de soluciones que usan XML-RPC y SOAP en Perl [en línea]

<http://linux.bankhacker.com/software/Programming+Web+Services+with+Perl/>

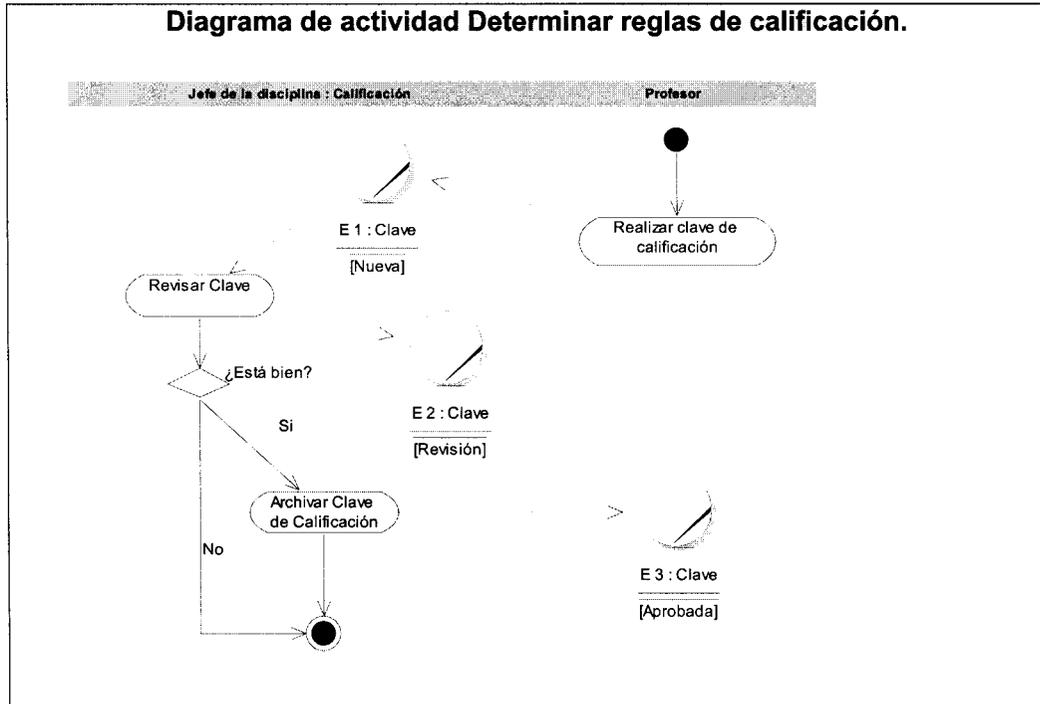
[consulta: 26/05/2004].

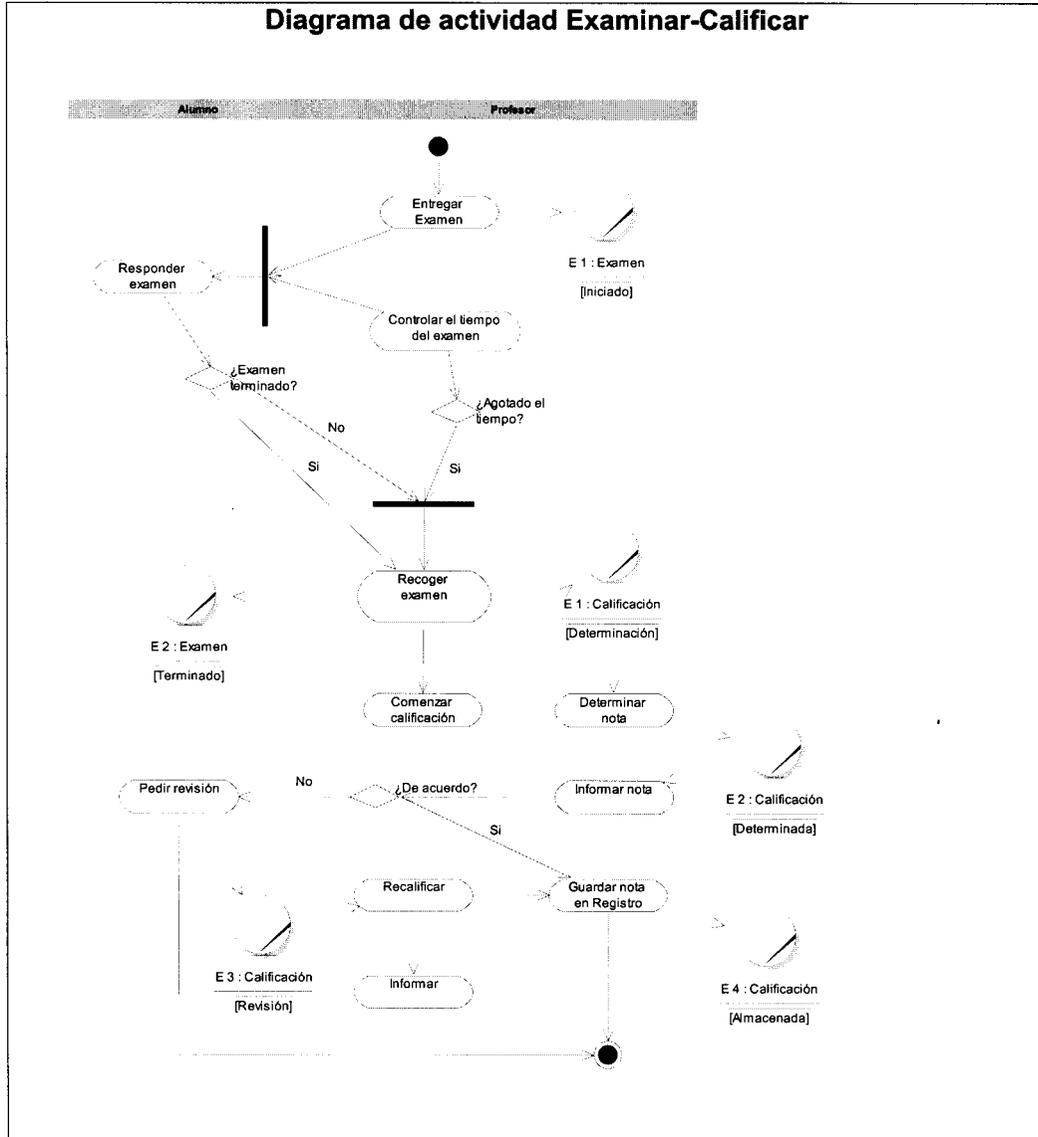
Anexos

Anexo 1 Diagramas del Modelo del negocio.

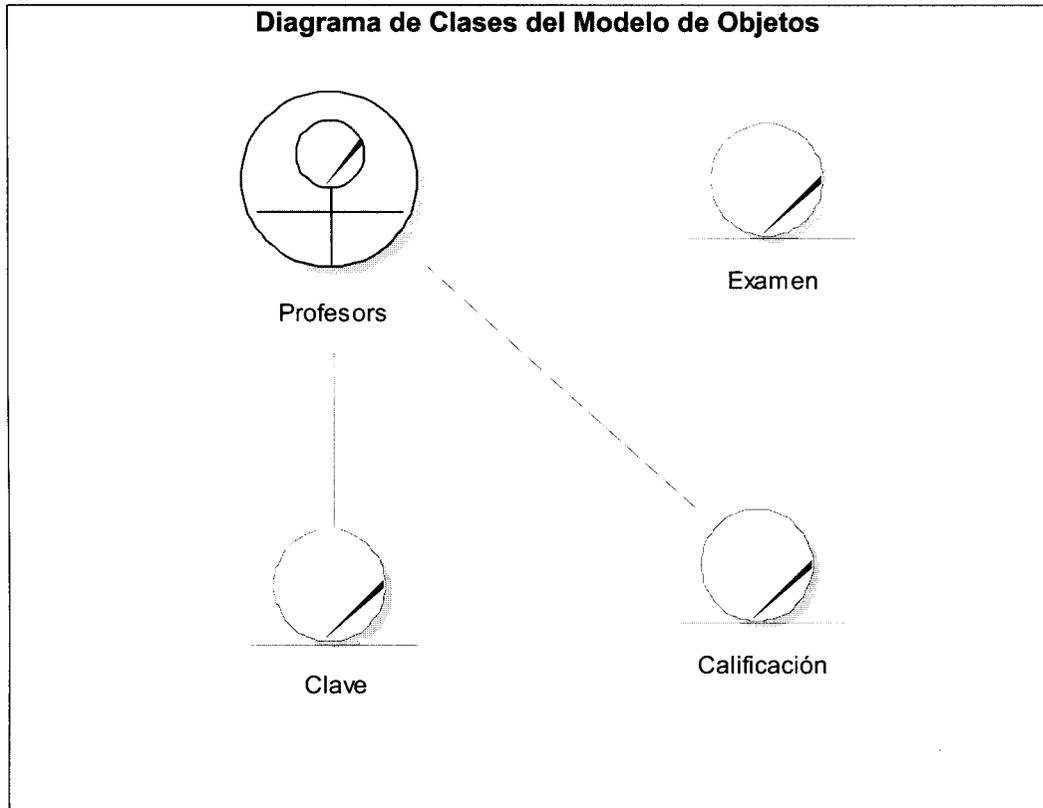


Anexo 2 Diagramas de Actividades del Modelo del Negocio.





Anexo 3 Diagrama de Clases del Modelo de Objetos.



Anexo 4 Casos de Uso Expandidos.

Caso de uso	
CU2	Iniciar Examen.
Propósito	Seleccionar examen a realizar.
Actores	
Usuario (inicia).	
Resumen:	
El caso de uso se inicia cuando el sistema le muestra los exámenes disponibles. El Usuario selecciona el examen a realizar. El caso de uso termina cuando el examen es importado del banco de examen.	
Referencias	R2.1
Acción del actor	Respuesta del sistema
1. Autenticar Usuario. 3. El Usuario selecciona el examen a realizar.	2. El sistema le muestra un listado de los exámenes disponibles. 4. El sistema importa el examen del banco de examen y se lo presenta al Usuario.
Flujo alternativo 4	
Acción del actor	Respuesta del sistema
	4.1 Si el sistema detecta que el examen es dirigido (es decir programado para una fecha y hora específico) lo visualiza al término de ese tiempo.
Puntos de extensión.	
1. Ver CU (Módulo 3 SICA) Autenticar Usuario	

Caso de uso	
CU3	Realizar Examen Estático.
Propósito	Comenzar a realizar el examen para comprobar sus conocimientos.
Actores	
Usuario (inicia).	
Resumen:	
El caso de uso se inicia cuando el Usuario decide comenzar a examinarse. Se generan las preguntas correspondientes al examen en cuestión. El caso de uso termina cuando el sistema detecta que el Usuario da como respondida una pregunta.	
Referencias	R2.2
Acción del actor	Respuesta del sistema
1. Iniciar Examen.	2. El sistema genera y presenta pregunta a Usuario.
	3. Controlar tiempo.
4. El Usuario responde pregunta.	5. Calificar Pregunta.
	6. Volver paso 3 hasta que se agoten las preguntas.
	7. Cerrar Calificación
Flujo alternativo -	
Acción del actor	Respuesta del sistema
-	-
Puntos de extensión.	

- | |
|---|
| 1. Ver CU2 Iniciar Examen.
3. Ver CU7 Controlar Tiempo.
5. Ver CU5 Calificar Pregunta.
7. Ver CU6 Cerrar Calificación. |
|---|

Caso de uso	
CU5	Calificar Pregunta.
Propósito	Evaluar cada pregunta del examen.
Actores	
Calificador Automático	
Resumen:	
<p>El caso de uso se inicia cuando el sistema detecta que el usuario da como respondida una de las preguntas. El caso de uso termina cuando el sistema antes de generarle la próxima pregunta almacena el resultado obtenido.</p>	
Referencias	R3.2.1
Acción del actor	Respuesta del sistema
2. El Calificador Automático chequea el si la pregunta es abierta o cerrada. 3. Si la pregunta es cerrada Calificador Automático procede a generar la calificación obtenida, en dependencia del valor asignado a la pregunta y si la respuesta es correcta o no.	1. El sistema detecta que se da como respondida una pregunta. 4. El sistema almacena la nota o calificación parcial de la pregunta y el ítem de esta en Resultados.
Flujo alternativo 3	
Acción del actor	Respuesta del sistema

	3.1 Si pregunta es abierta entonces se almacena el ítem y la respuesta de la pregunta en Resultados, en espera que el Calificador Humano solicite calificar pregunta.
Puntos de extensión.	
-	

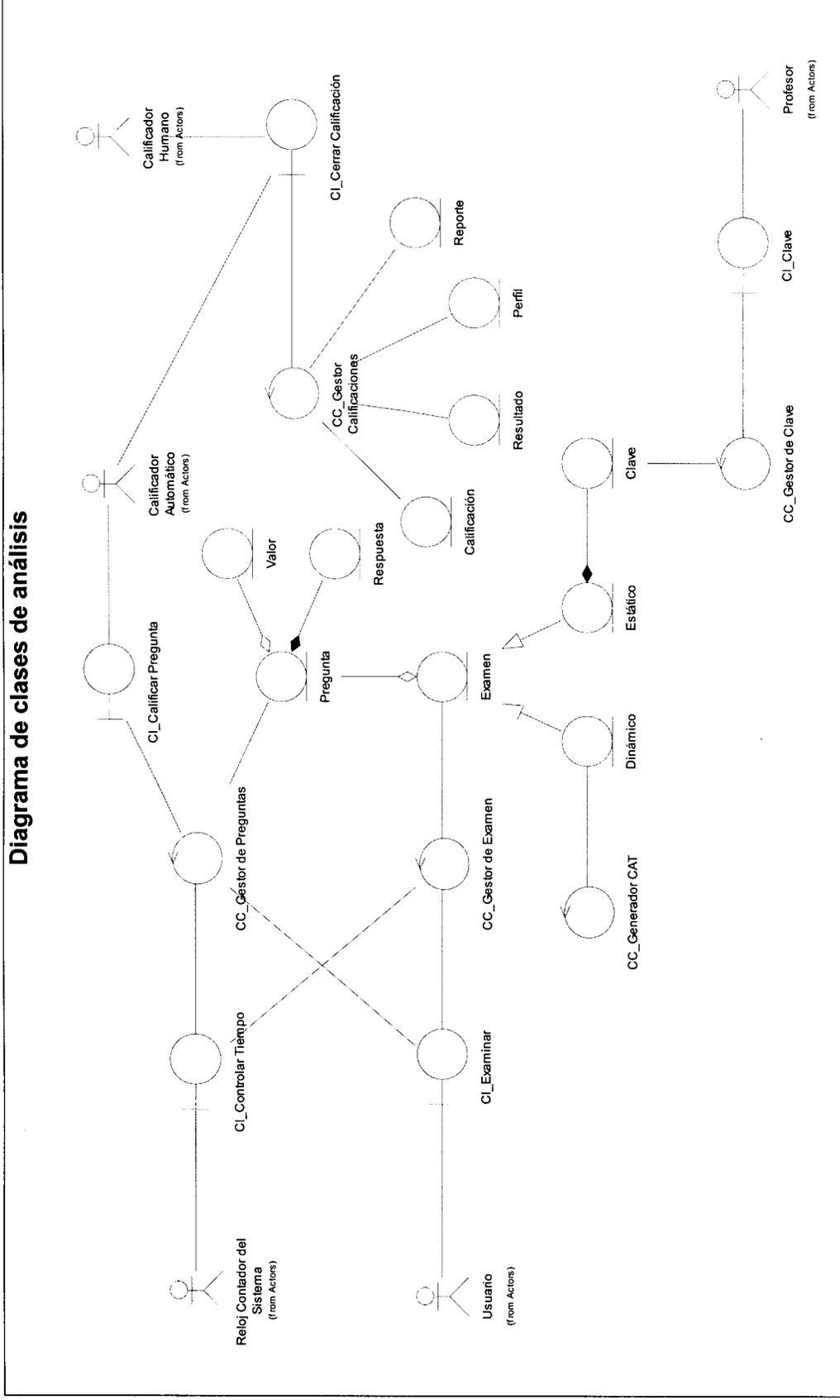
Caso de uso	
CU6	Cerrar Calificación
Propósito	Dar nota al Examen
Actores	
Calificador Automático, Calificador Humano	
Resumen:	
<p>El caso de uso se inicia cuando el sistema detecta el tiempo de examen esta agotado se dispone a sumar las calificaciones hechas. El caso de uso termina cuando, si el usuario es alumno registrado entonces se almacena en su perfil personal la calificación y el reporte dado.</p>	
Referencias	R3.2.2 y R3.3
Acción del actor	Respuesta del sistema
<p>2. El Calificador Automático chequea que no existan preguntas sin calificar.</p> <p>3. El Calificador Automático del sistema cierra la calificación, hallando el total de la puntuación y con este total se dirige a la Clave, para obtener la calificación final del Usuario.</p>	<p>1. El sistema detecta el tiempo de examen esta agotado.</p> <p>4. El sistema genera un reporte para cada uno de los usuarios.</p>
Flujo alternativo 2	

Acción del actor	Respuesta del sistema
<p>2.2 El Calificador humano procede a calificar la pregunta.</p>	<p>2.1 Si existen preguntas sin calificar queda en espera de que el Calificador Humano las califique.</p> <p>2.3 El sistema almacena la nota o calificación parcial en Resultados, dada por el Calificador Humano en cada pregunta abierta.</p>
Puntos de extensión.	
4. Ver CU8 Generar reporte	

Caso de uso	
CU7	Controlar tiempo
Propósito	Controlar el tiempo de duración tanto del examen, como de cada una de sus preguntas.
Actores Reloj contador de sistema(inicia)	
Resumen: El caso de uso se inicia cuando comienza el examen y el Reloj contador de sistema se inicia en 0. El tiempo de examen esta concluyendo, entonces el sistema envía un mensaje de aviso. El caso de uso termina cuando agotado este tiempo se retira la pregunta o el examen	
Referencias	R3.1
Acción del actor	Respuesta del sistema
<p>1. El reloj comienza a controlar el tiempo de duración del examen, inicializando el contador de examen en 0.</p> <p>2. Así como la duración de cada</p>	

<p>pregunta, inicializando el contador de pregunta en 0.</p> <p>3. El reloj envía un mensaje para advertir sobre el consumo del tiempo de respuesta de cada pregunta.</p> <p>6. El Reloj detecta que el tiempo de examen se esta agotando.</p> <p>7. El Reloj envía un mensaje para advertir que el tiempo del examen termina.</p>	<p>4. Si el tiempo se agotó entonces, el sistema quita las preguntas.</p> <p>5. Volver paso 2. Hasta que acaben las preguntas.</p> <p>8. Concluido este tiempo, el sistema retira el examen.</p>
<p>Flujo alternativo -</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>-</p>	<p>-</p>
<p>Puntos de extensión.</p>	
<p>-</p>	

Anexo 5 Diagrama de Clases del Análisis



Anexo 6 Diagramas de Interacción.

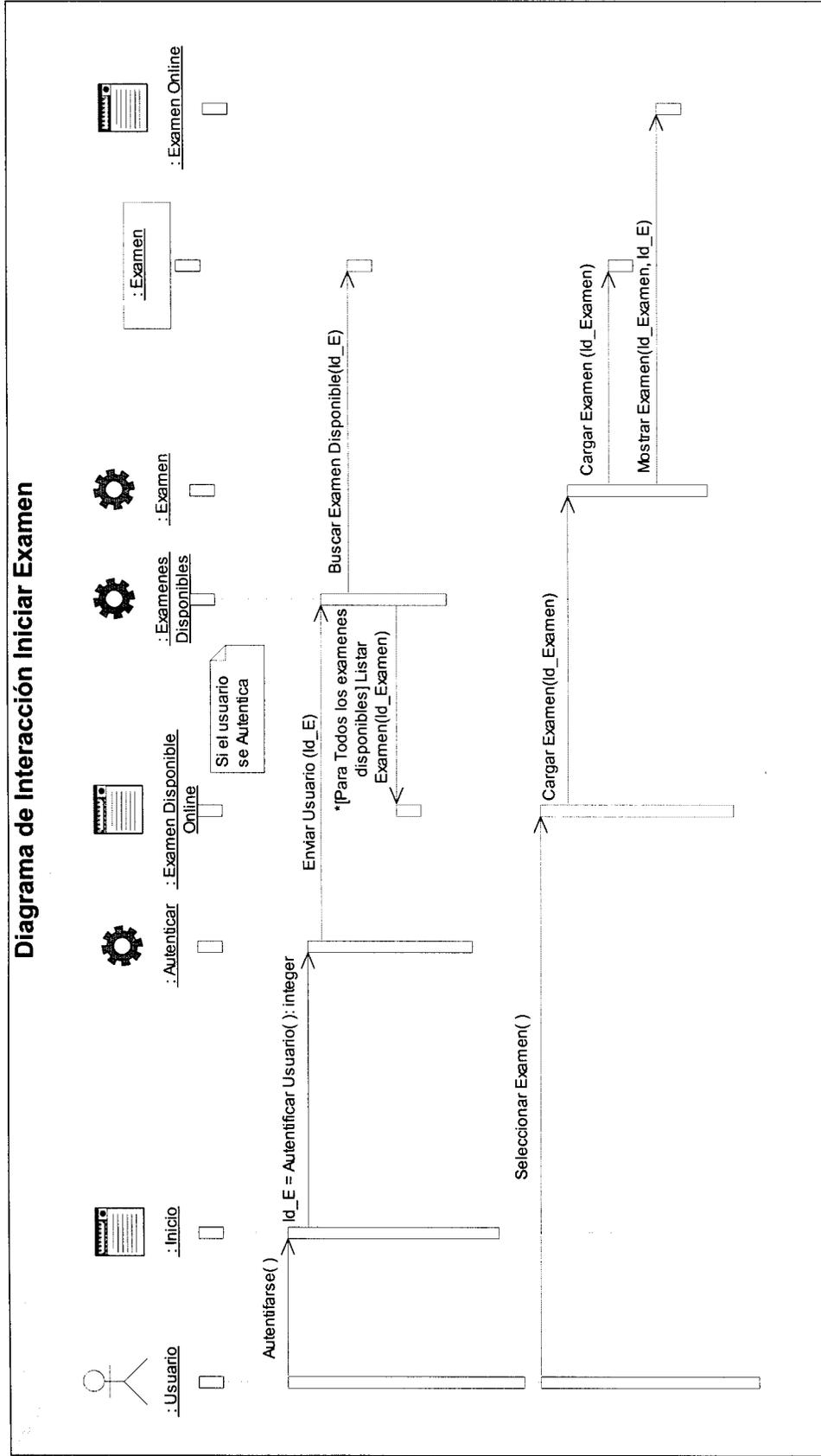


Diagrama de Interacción Realizar Examen Estático

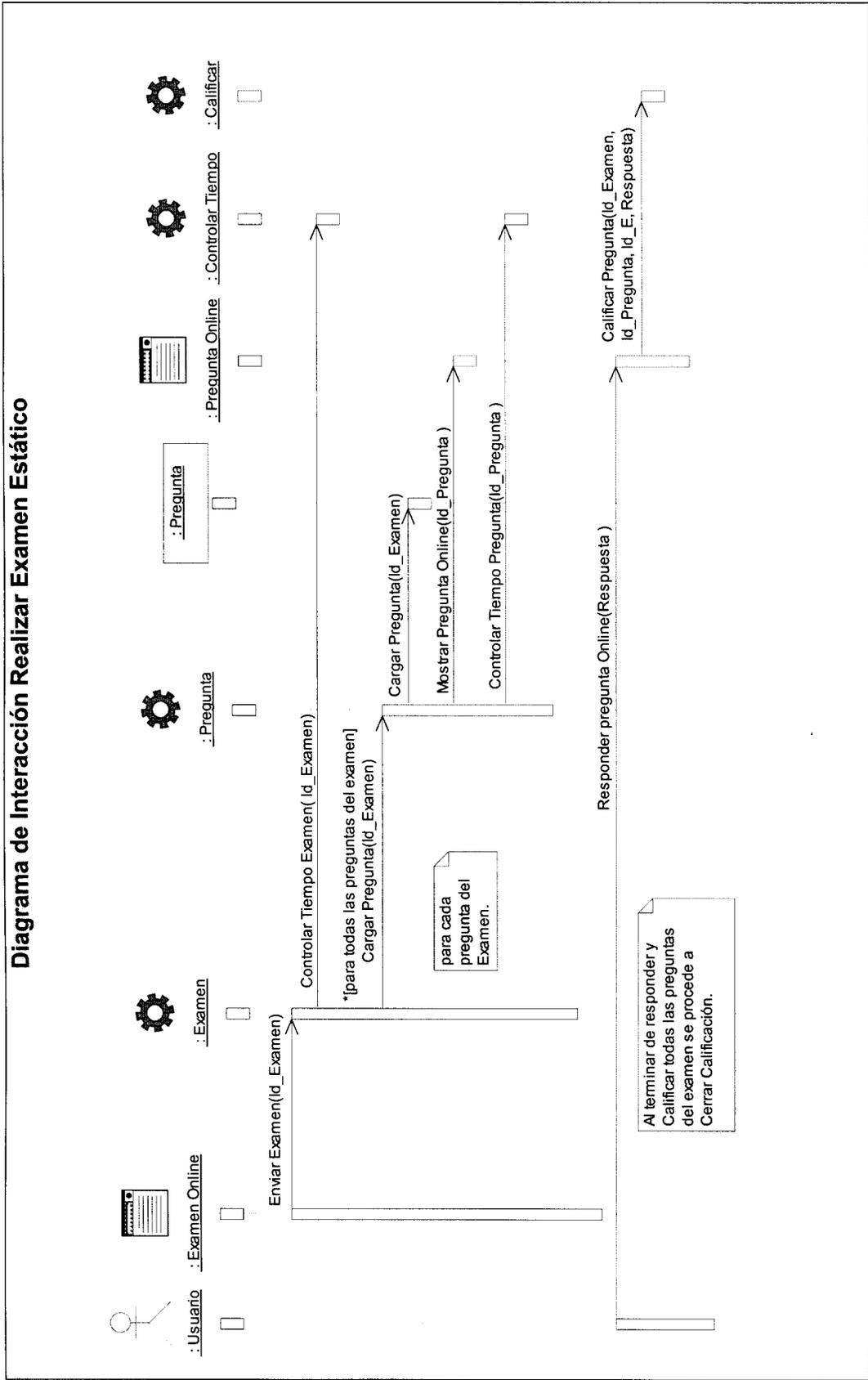


Diagrama de Interacción Controlar Tiempo Examen

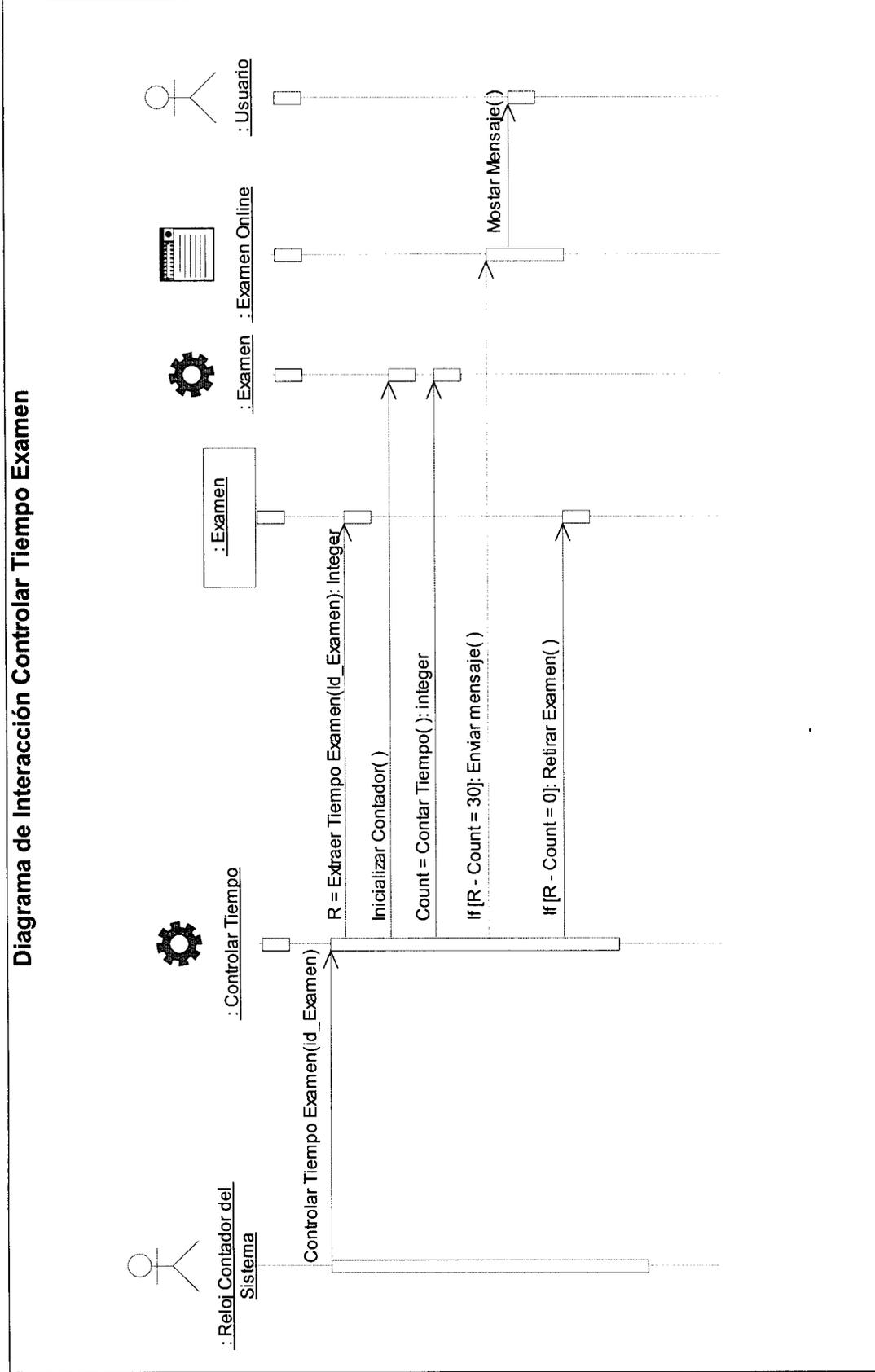


Diagrama de Interacción Controlar Tiempo Pregunta

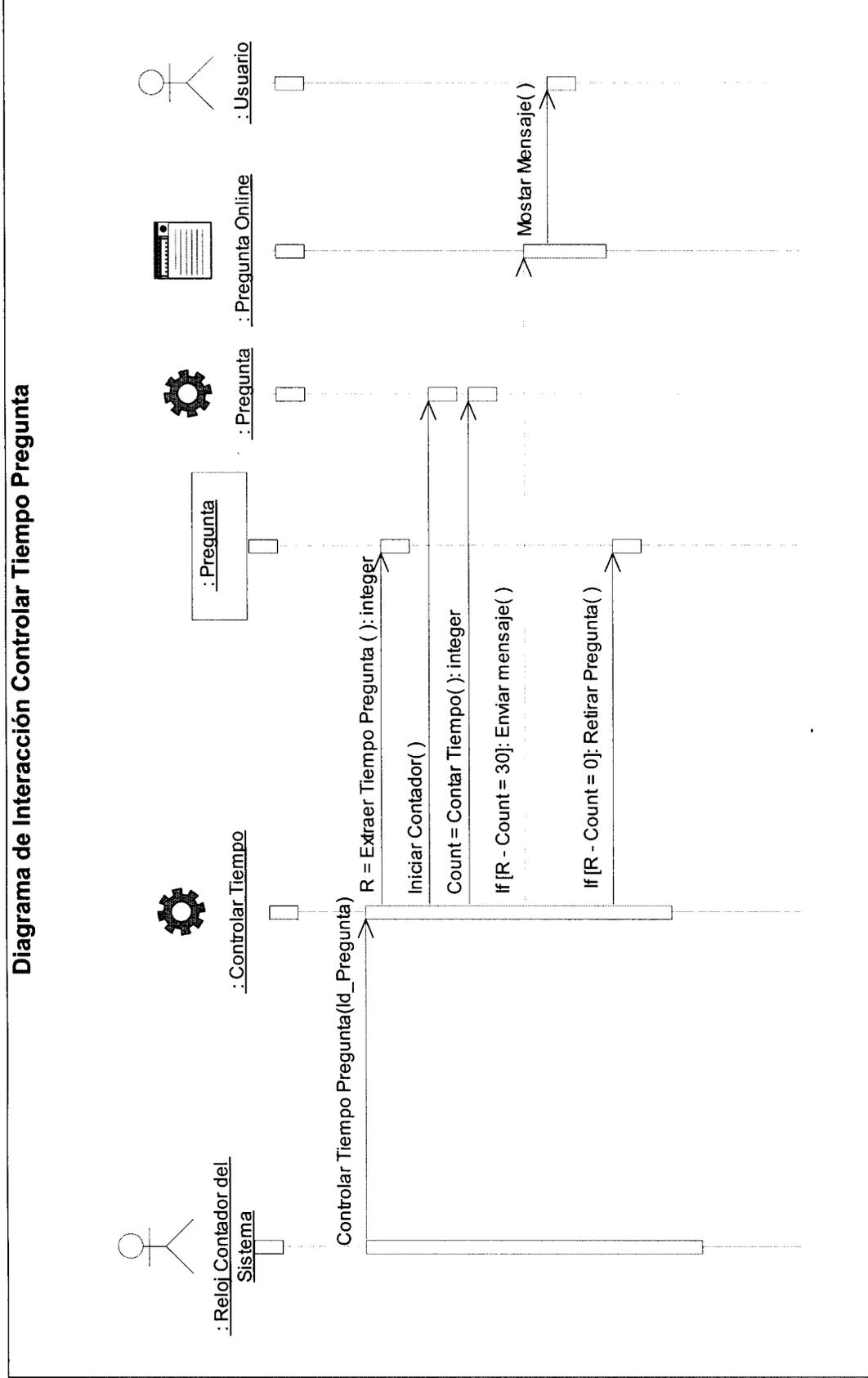


Diagrama de Interacción Calificar Pregunta

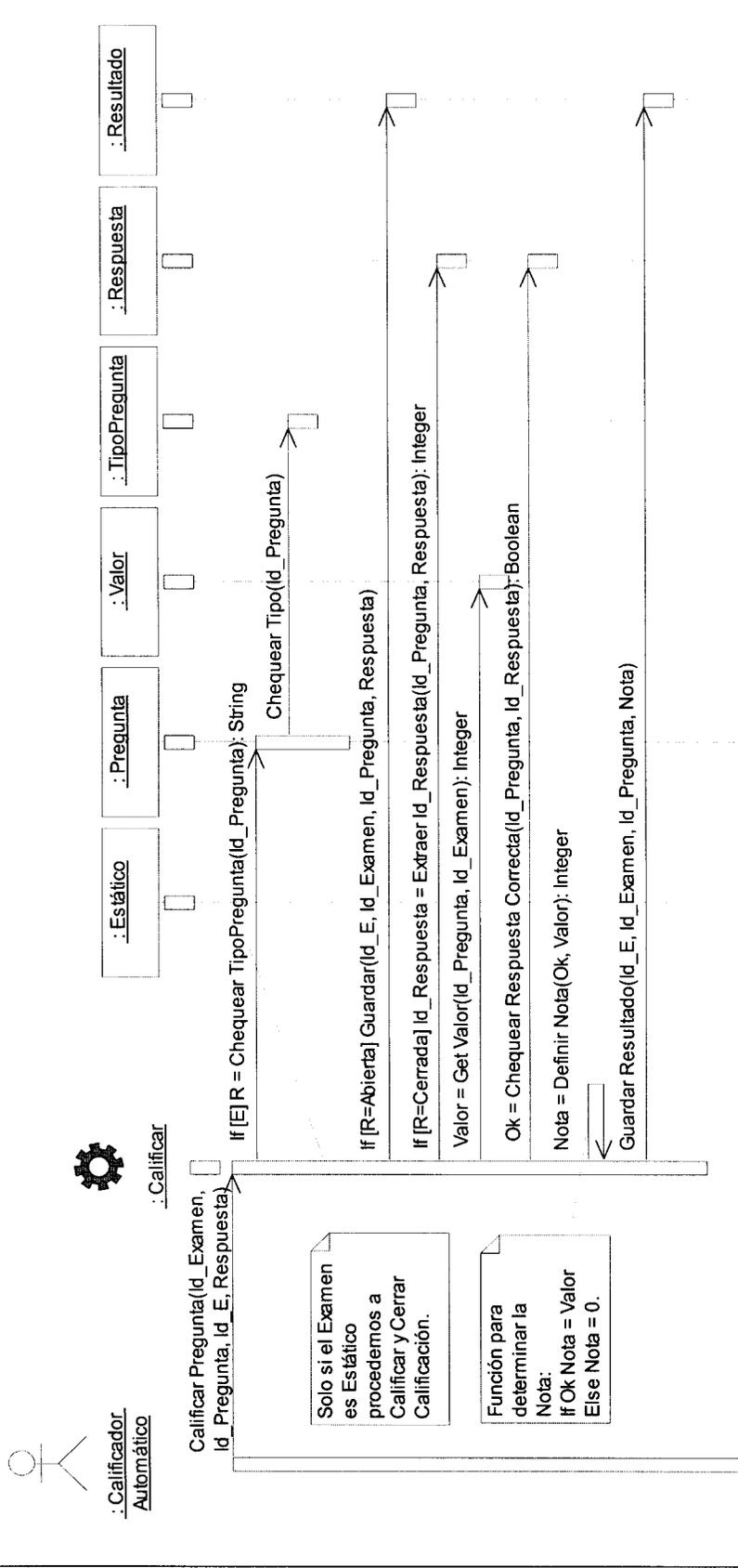
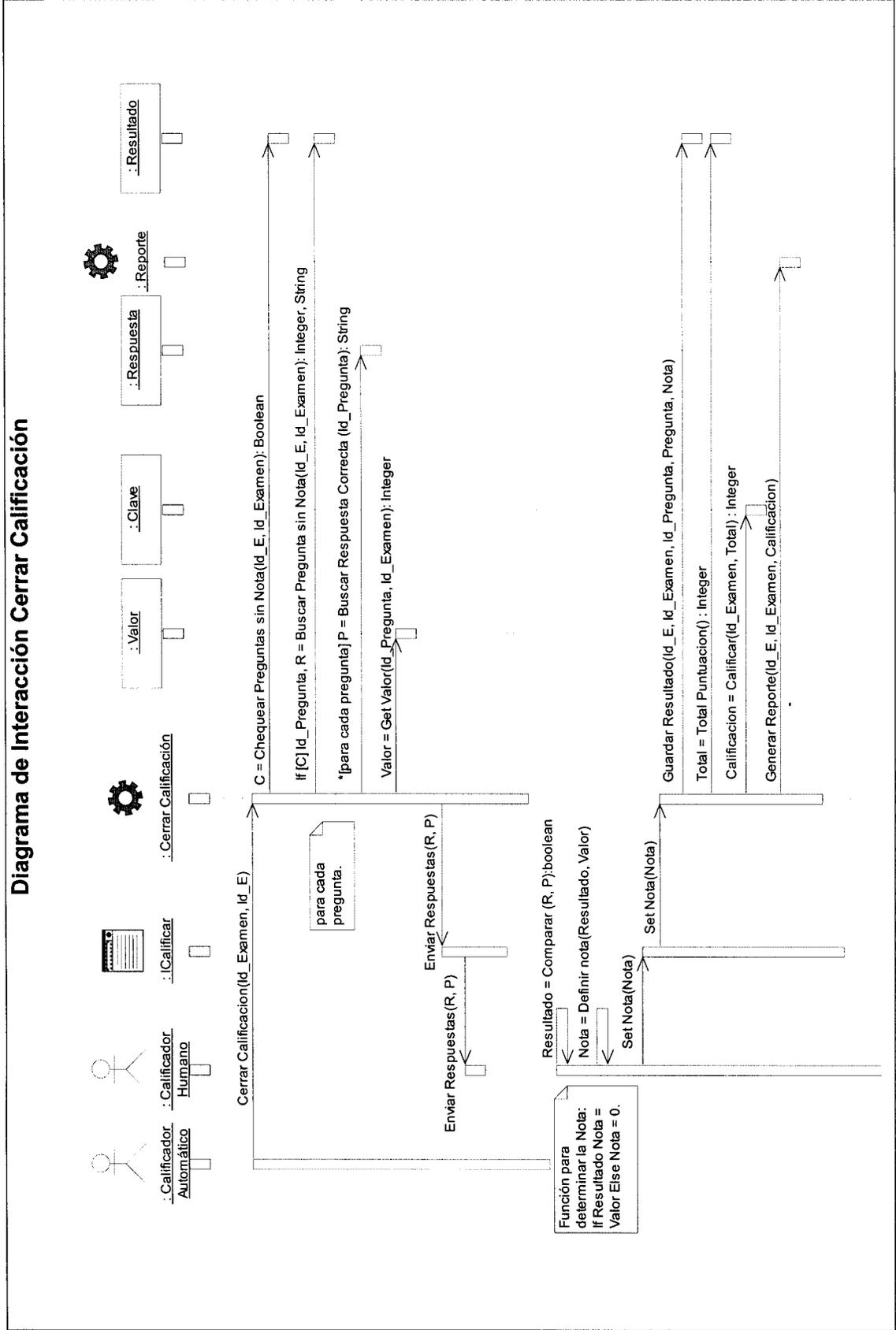


Diagrama de Interacción Cerrar Calificación



Anexo 7 Diagramas de Clases

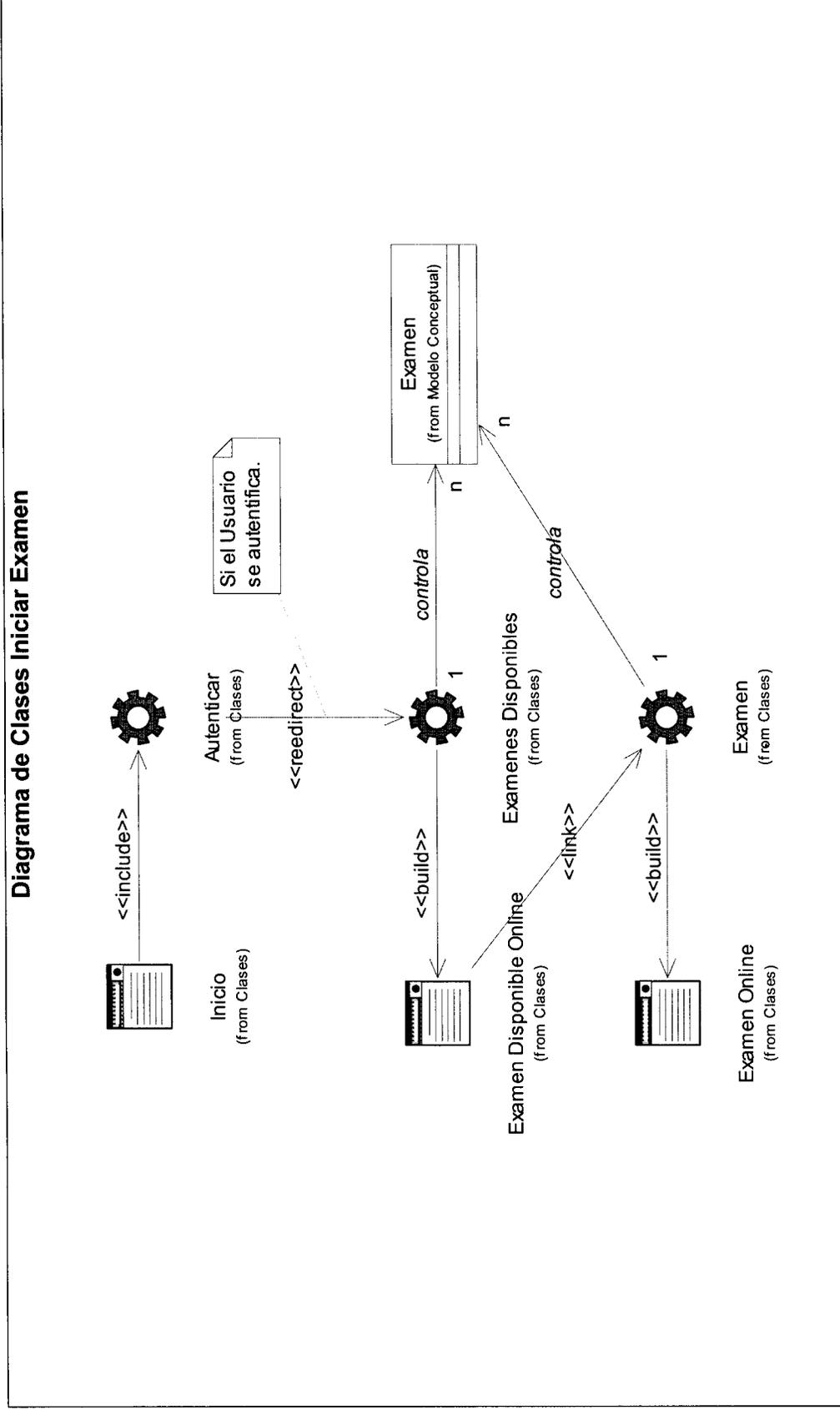


Diagrama de Clases Realizar Examen

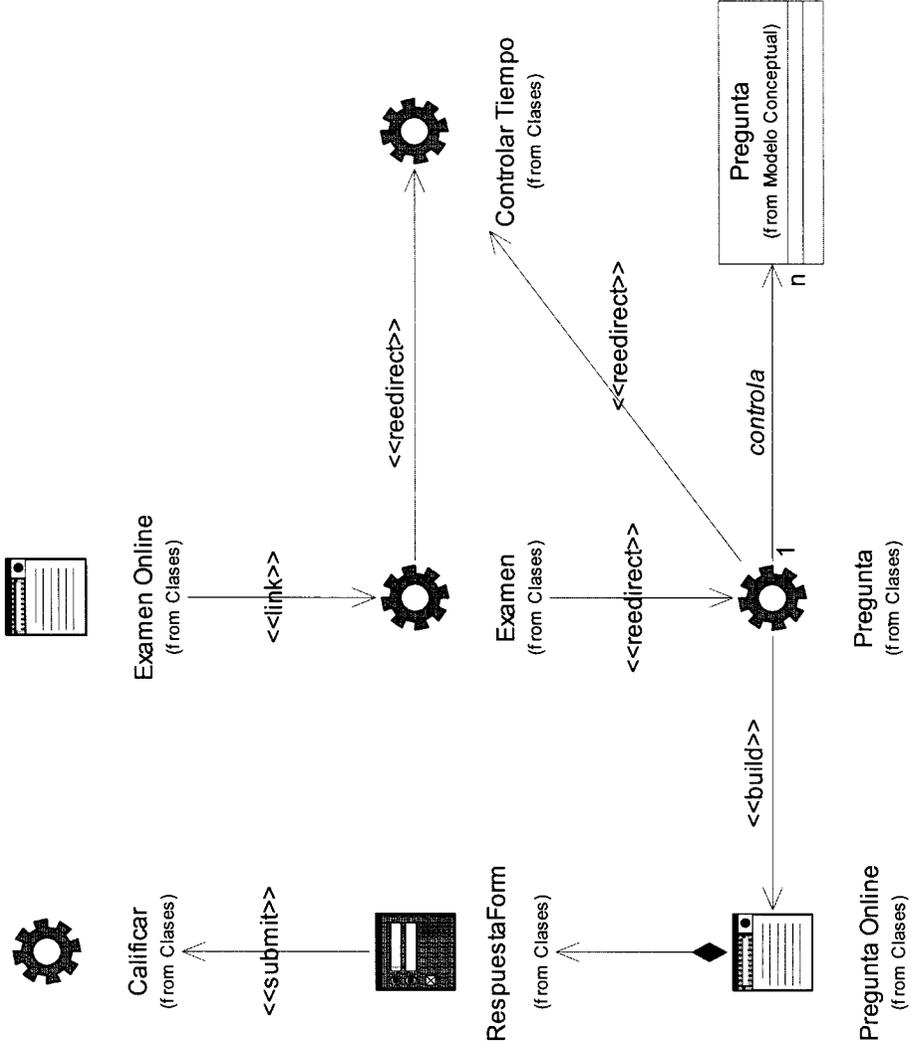


Diagrama de Clases Controlar Tiempo

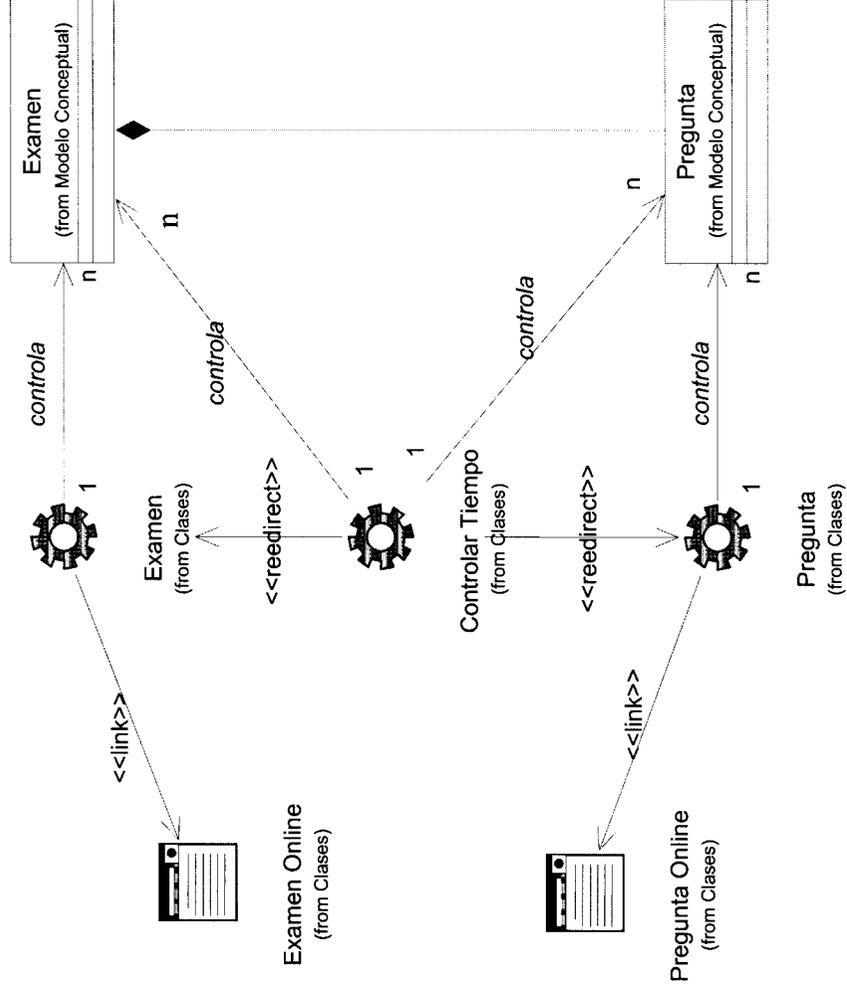


Diagrama de Clases Calificar

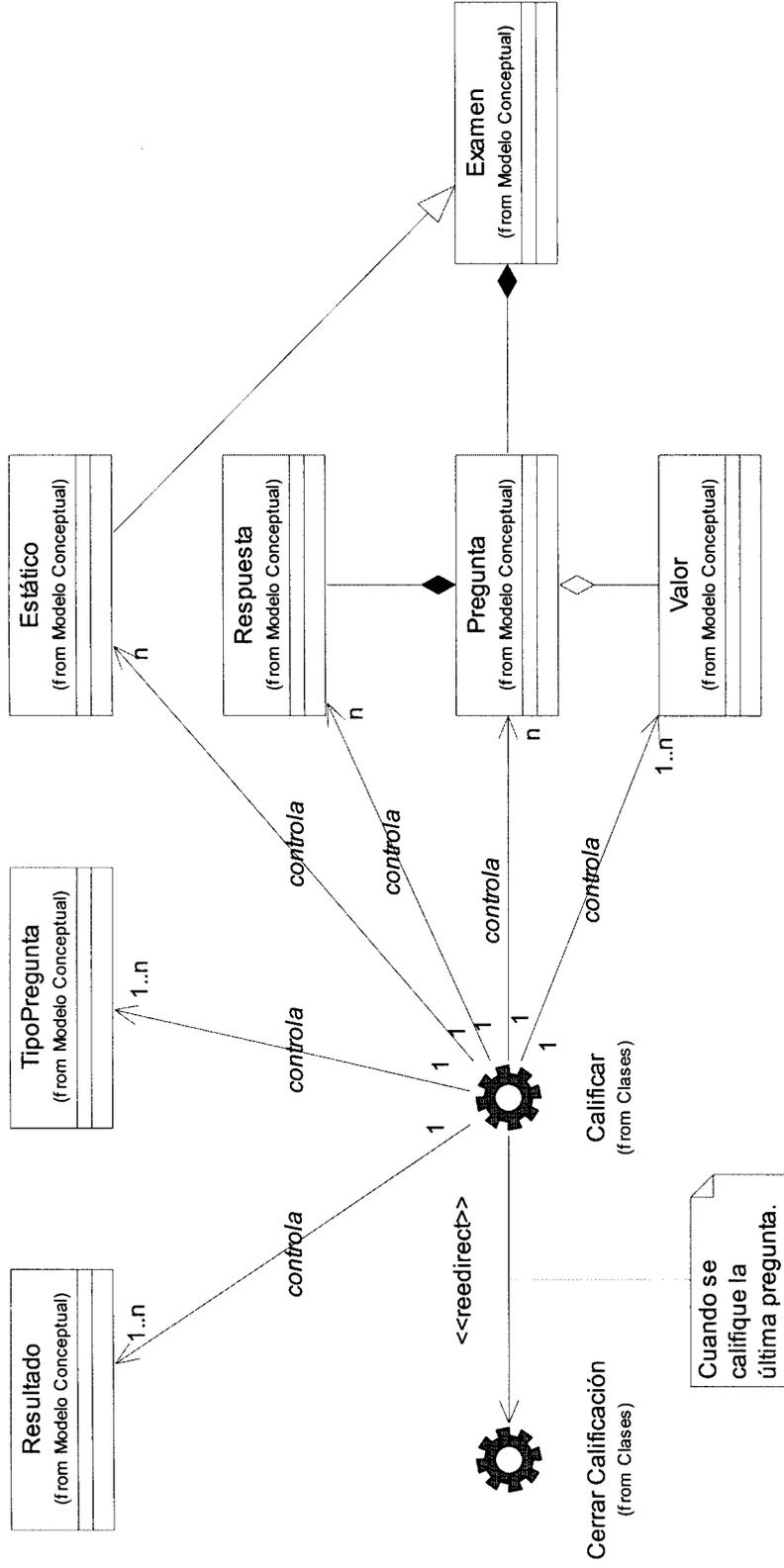
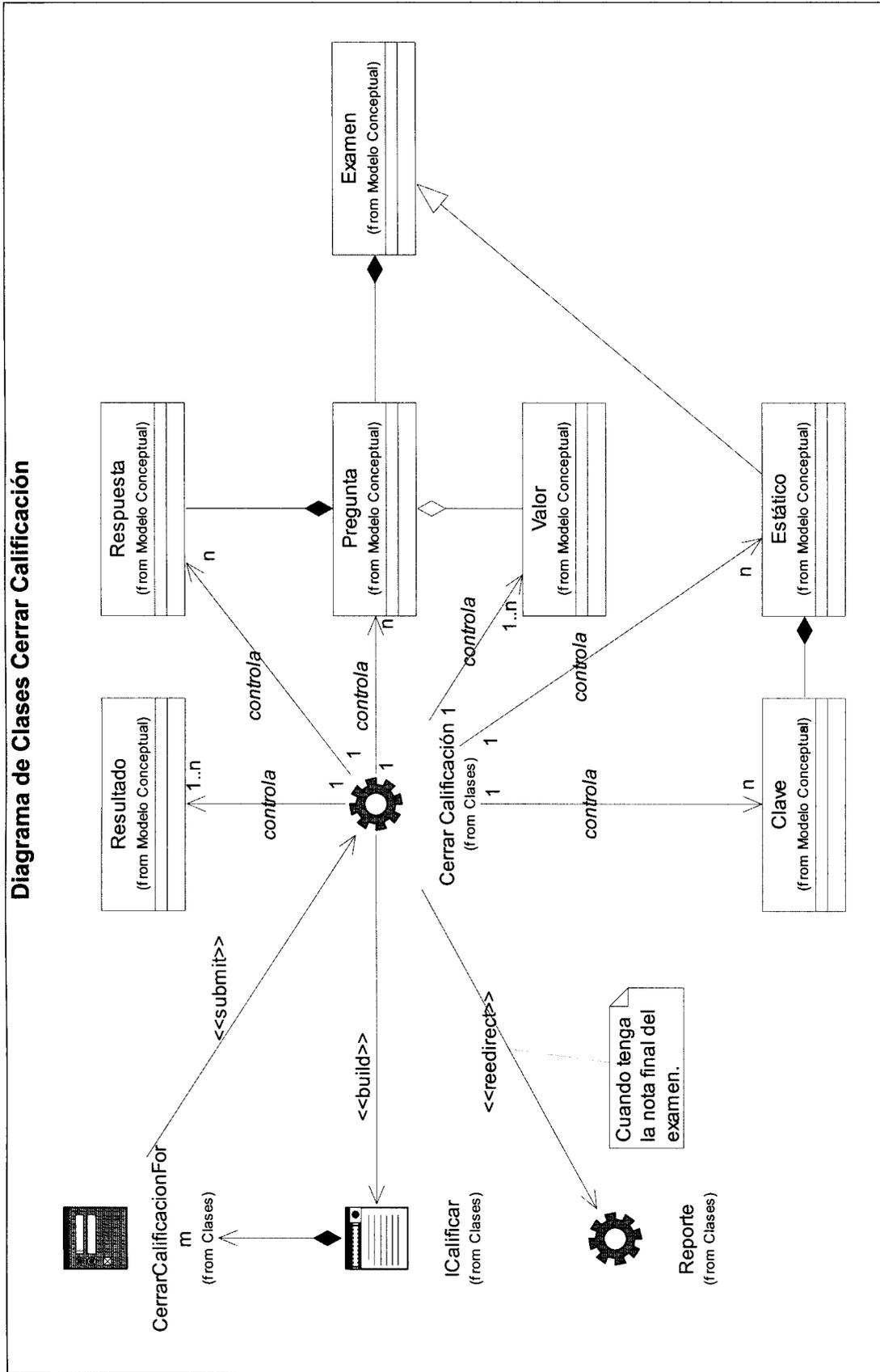


Diagrama de Clases Cerrar Calificación



Anexo 8 Descripción de las Clases

Nombre: Clave	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Nombre	String
Rango	Integer
Calif	String
Para cada responsabilidad:	
Nombre:	Descripción:
Calificar	Chequear el resultado de la puntuación obtenida en las notas parciales obtenido para determinar una calificación al examen.

Nombre: Estático	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Tiempo de Respuesta	String
Para cada responsabilidad:	
Nombre:	Descripción:
Chequear Examen	Verificar si el examen es estático o no para proceder a la calificación.

Nombre: Examen	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Título	String
Tipo de Examen	String
Disponibilidad	String
Propietario	String
Objetivos	String
Para cada responsabilidad:	
Nombre:	Descripción:

Cargar Examen	Visualizar el examen para empezar a responderlo.
Buscar Exámenes Disponibles	Listar los exámenes a los que pueda acceder el usuario.
Extraer Tiempo Examen	Necesita conocer este valor de tiempo para el que está asignado el examen.
Retirar Examen	Cuando se agote el tiempo predestinado para el examen, incluyendo si hay un tiempo extra, ocultar este de manera que el usuario no pueda continuar realizándolo.

Nombre: Pregunta	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Identificador	
Tiempo de Respuesta	Integer
Objetivos	String
Para cada responsabilidad:	
Nombre:	Descripción
Chequear Tipo Pregunta	Verificar que tipo de pregunta es, si abierta o cerrada.
Cargar pregunta	Hacer visible la pregunta para que el usuario pueda leerla y responderla.
Extraer Tiempo Pregunta	Necesita conocer este valor para contar el tiempo que la pregunta aparece disponible mientras el usuario la responde.
Retirar pregunta	Cuando se agote el tiempo predestinado para la pregunta, retirarla para su calificación, de manera que el usuario no pueda continuar respondiéndola.

Nombre: Respuesta	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Respuesta	String

Correcta	Boolean
Para cada responsabilidad:	
Nombre:	Descripción:
Chequear Respuesta	Confirmar si la respuesta es la correcta o no.
Extraer Id_Respuesta	Si la pregunta es cerrada extraigo el Identificador de la respuesta.

Nombre: Resultado	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Buscar preguntas sin Nota	Seleccionar las preguntas sin nota, o abiertas, para ser calificadas por el Profesor.
Chequear preguntas sin Nota.	Chequear si existe alguna pregunta sin nota.
Guardar	Guardar en la tabla los datos de las preguntas abiertas en espera de nota.
Guardar Resultado	Guardar en la tabla los datos de las preguntas cerradas con la nota en espera de que el profesor califique para poder dar calificación al usuario

Nombre: Tipo Pregunta	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Chequear Tipo Pregunta	Verificar que tipo de pregunta es, si abierta o cerrada.

Nombre: Valor	
Tipo de clase (interfaz, controladora, entidad--) Entidad	
Atributo	Tipo
Valor	Integer
Para cada responsabilidad:	
Nombre:	Descripción:
Get Valor	Extraer el valor de una pregunta en un examen para otorgar la nota correspondiente del estudiante en dicha pregunta.

Nombre: Autenticar	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Autenticar Usuario	Comprobar la existencia del usuario en el dominio.

Nombre: Calificar	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Calificar Pregunta	Evaluar la respuesta a esa pregunta, es decir, obtener una nota parcial.
Definir Nota	Definir la calificación parcial de la pregunta en dependencia del valor que le fue asignado, es decir, si respondió correctamente la nota es el valor, si no es así, entonces la nota es 0.

Nombre: Cerrar Calificación	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Cerrar Calificación	Obtener la calificación final de un usuario en un examen.
Definir Nota	Definir la calificación parcial de la pregunta en dependencia del valor que le fue asignado, es decir, si respondió correctamente la nota es el valor, si no es así, entonces la nota es 0.
Total Puntuación	Equivale a obtener la calificación final del usuario.
Set Nota	Recibe la nota o calificación parcial, asignada por el calificador humano, para manipularla y confeccionar el resumen pertinente.

Nombre: Controlar Tiempo	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Controlar Tiempo Examen	Llevar el conteo del tiempo del Examen.
Controlar Tiempo Pregunta	Llevar el conteo del tiempo de la Pregunta.

Nombre: Examen	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo

-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Cargar Examen	Extraer el examen de la base de datos para que el usuario pueda resolverlo.
Inicializar Contador	Poner contador en 0.
Contar Tiempo	Ir incrementando el contador a medida que transcurre el tiempo real.
Enviar Examen	Para conocer cual es el examen a realizar y buscar las preguntas correspondiente.

Nombre: Examen Disponible	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Cargar Examen Disponible	Ya seleccionado el examen por el usuario mandar a cargar este de la base de datos.
Enviar Usuario	Es cuando el usuario logra autenticarse y recibe el identificador de este para buscar cuales son los exámenes a los que puede acceder.

Nombre: Pregunta	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Cargar Pregunta	Extraer la pregunta de la base de datos para que el usuario pueda resolverlo.
Iniciar Contador	Poner contador en 0.

Contar Tiempo	Ir incrementando el contador a medida que transcurre el tiempo real.
---------------	--

Nombre: Reporte	
Tipo de clase (interfaz, controladora, entidad--) Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Generar Reporte	Trabaja con los datos almacenados en Resultado, con el fin de hacer un pequeño resumen de las fortalezas y debilidades del usuario, permitiéndole valorar cuales son los objetivos cumplidos y en cuales debe trabajar. Además de visualizarle la calificación en el examen realizado.

Nombre: Examen Online	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Mostrar Examen	Hacer el examen seleccionado visible para que el usuario pueda resolverlo.
Realizar Examen	Que el usuario comience a darle respuesta al examen.
Enviar Mensaje	Visualizar el mensaje de alerta cuando resten 30 seg., para que se agote el tiempo de respuesta del examen.

Nombre: Examen Disponible Online	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo

-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Seleccionar Examen	Brindarle la posibilidad al usuario de escoger un examen del listado de exámenes disponibles.
Listar Examen	Listar todos los exámenes disponibles y accesibles para el usuario, de manera que este seleccione cual desea dese realizar en ese momento.

Nombre: ICalificar	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Set Nota	Enviarle la nota o calificación parcial asignada por el calificador a la pregunta para ser almacenada y obtener la calificación final del usuario en el examen.
Enviar Respuesta	Ponerle visible al calificador humano las Respuestas, tanto la que dio el usuario a examinarse como la correcta, para que este pueda compararlas y llegar a conclusiones.

Nombre: Inicio	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Autenticar	Brindarle la posibilidad al usuario de autenticarse.

Nombre: Pregunta Online	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
Mostrar Pregunta Online	Mostrar la pregunta que el usuario debe responder.
Responder Pregunta Online	Recoger la respuesta del usuario.
Enviar Mensaje	Visualizar el mensaje de alerta cuando resten 30 seg., para que se agote el tiempo de respuesta de la pregunta.

Nombre: CerrarCalificacionForm	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
-	-

Nombre: RespuestaForm	
Tipo de clase (interfaz, controladora, entidad--) Interfaz	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Descripción:
-	-

Anexo 9 Diagrama Entidad Relación

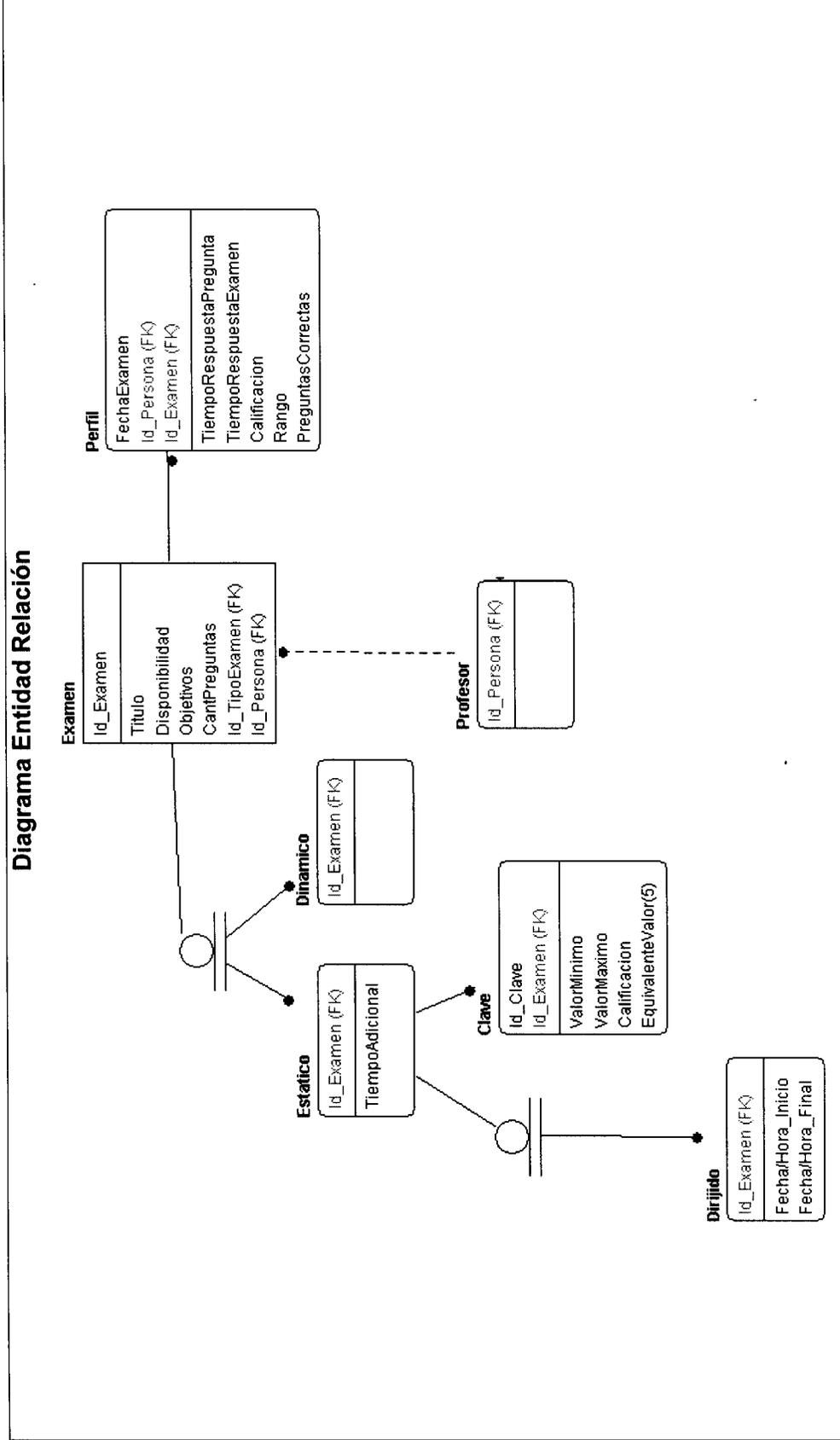


Diagrama Entidad Relación

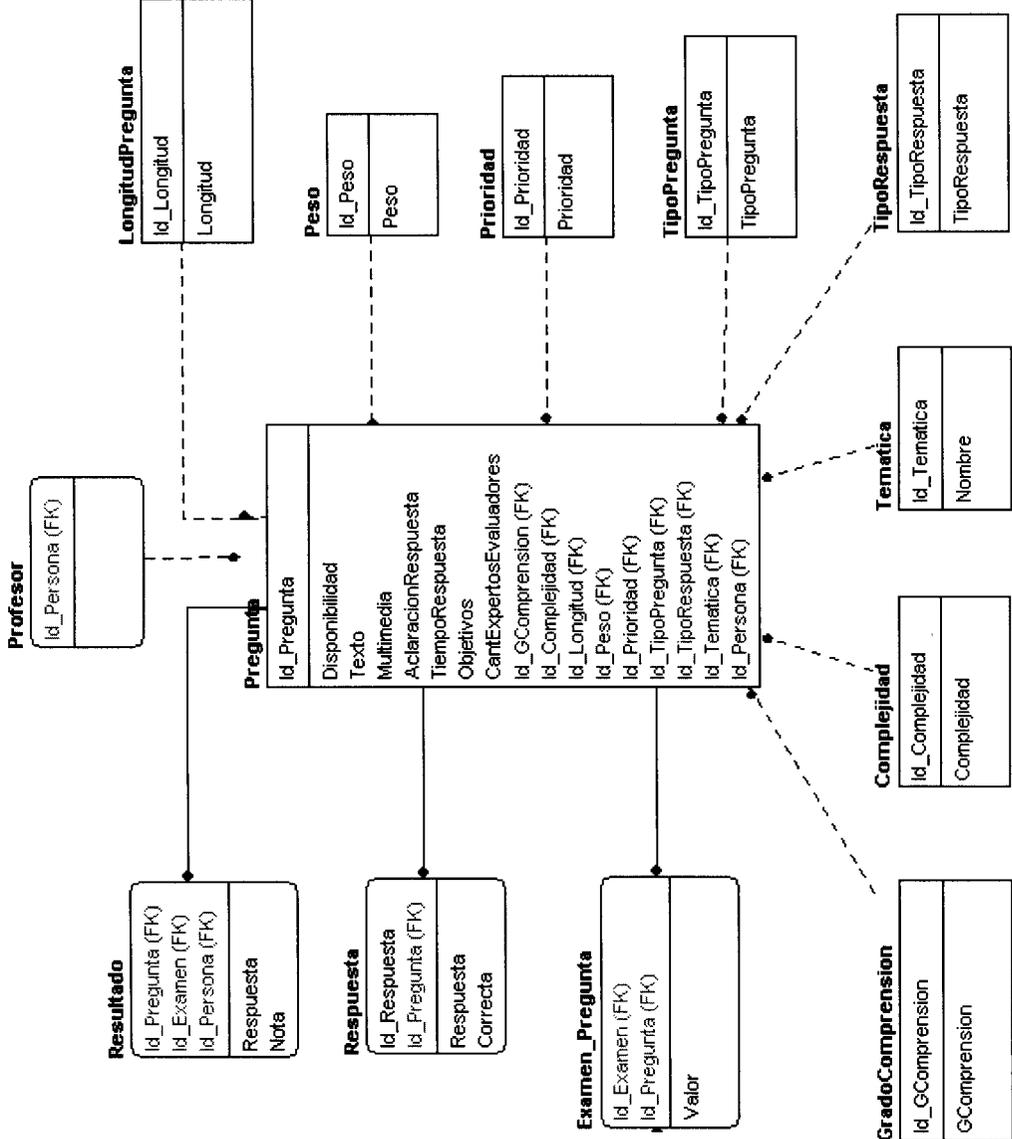


Diagrama Entidad Relación

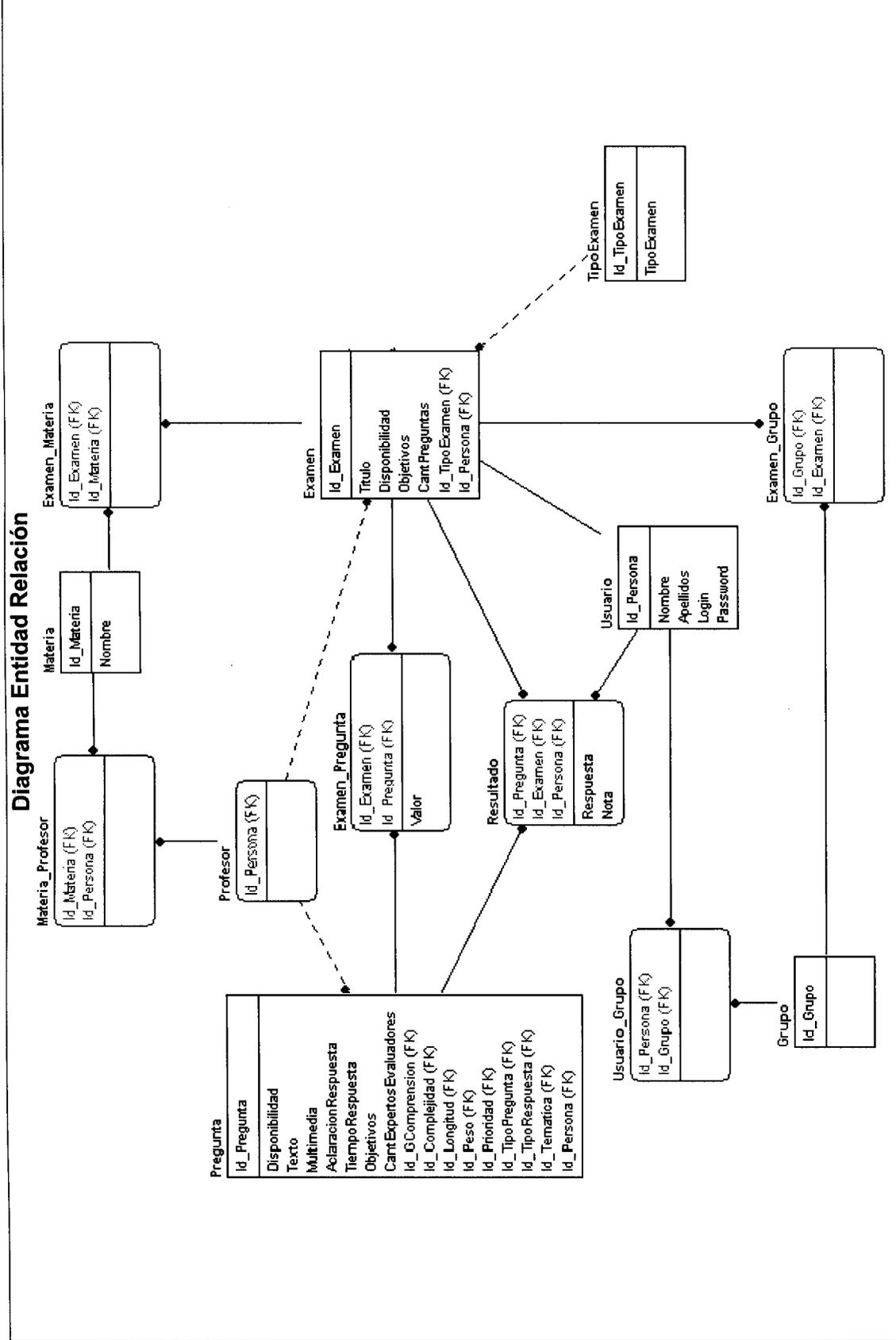
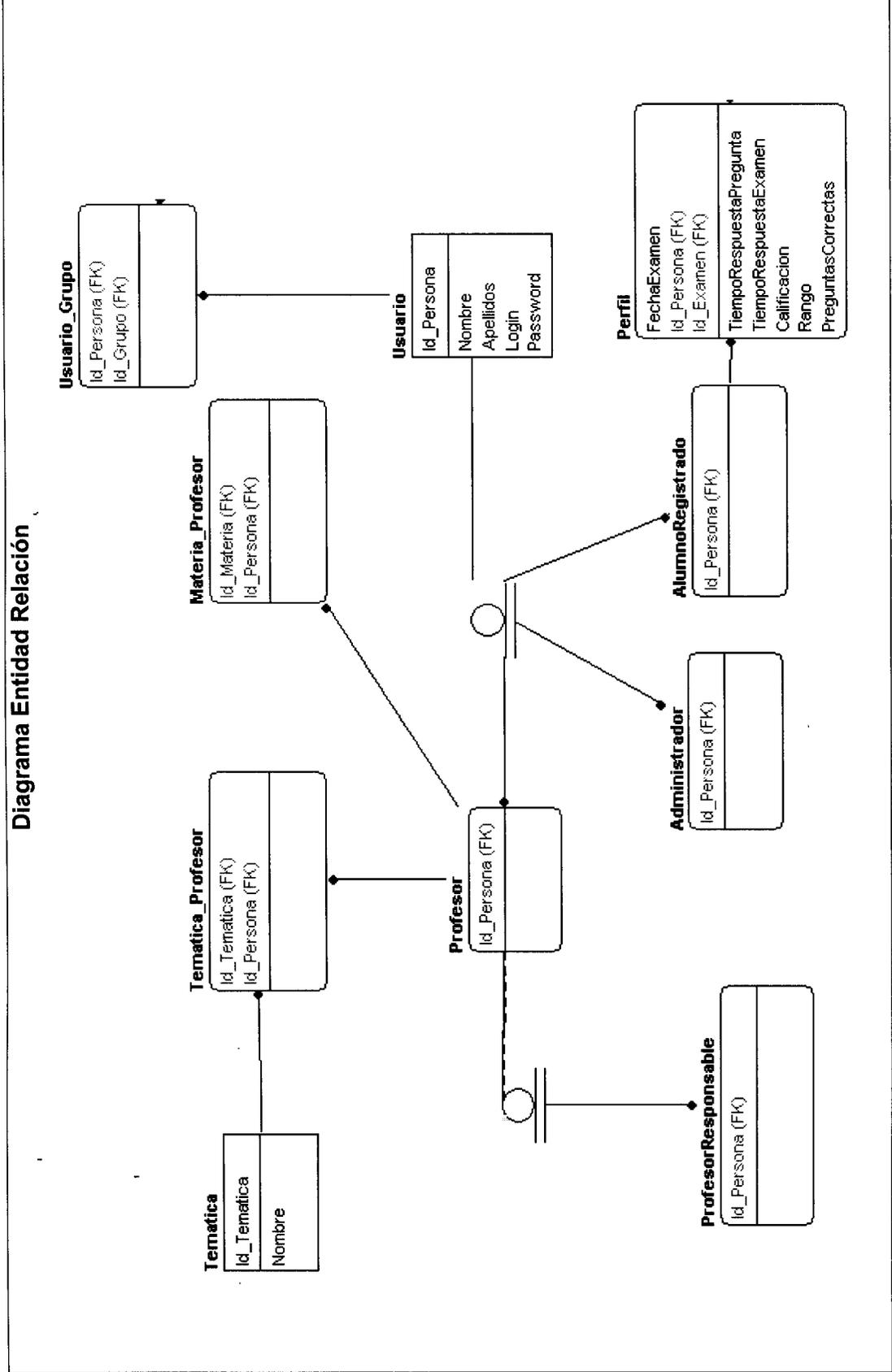


Diagrama Entidad Relación



Anexo 10 Descripción de las Tablas.

Nombre: Alumno Registrado		
Descripción: Especificación de un usuario, quien como dato adicional tendrá un perfil asociado.		
Atributo	Tipo	Descripción
Id_Persona	Integer	Identificador de Persona.

Nombre: Clave		
Descripción: Modelo matemático de calificación, ayuda a determinar la calificación de un usuario en un examen.		
Atributo	Tipo	Descripción
Id_Clave	Integer	Identificador de la clave.
Id_Examen	Integer	Examen al que hace referencia.
ValorMinimo	Integer	Rango en que oscilan los valores para otorgar la calificación
ValorMaximo	Integer	
Calificacion	String	Puntuación que se determina, nota.
EquivalenteValor(5)	Integer	Equivalente del aprobado en base a 5 puntos.

Nombre: Dinámico		
Descripción: Especificación de un examen que es generado por el CATs.		
Atributo	Tipo	Descripción
Id_Examen	Integer	Identificador del examen.

Nombre: Dirigido		
Descripción: Especificación de un examen programado para ser visualizado a una hora determinada.		
Atributo	Tipo	Descripción
Id_Examen	Integer	Identificador del examen.

Fecha/Hora_Inicio	DateTime	Fecha de inicio en la que se debe comenzar a responder el examen.
Fecha/Hora_Final	DateTime	Fecha de finalización del examen.

Nombre: Estático		
Descripción: Especificación de un examen.		
Atributo	Tipo	Descripción
Id_Examen	Integer	Identificador del examen.
TiempoAdicional	Integer	Tiempo adicional que se le otorga para responder el examen.

Nombre: Examen		
Descripción: Contiene los datos perteneciente al examen.		
Atributo	Tipo	Descripción
IdExamen	Integer	Identificador del examen.
Titulo	String	Nombre o descripción del Examen.
Disponibilidad	Boolean	Si el examen esta disponible o no.
Objetivos	String	Objetivos que se deben cumplir con el examen.
CantPreguntas	Integer	Cantidad de preguntas que contiene el examen.
Id_TipoExamen	Integer	Tipo de examen al que hace referencia.
Id_Persona	Integer	Propietario del Examen

Nombre: Examen_Pregunta		
Descripción: Preguntas que son asignadas a un examen.		
Atributo	Tipo	Descripción
Id_Examen	Integer	Examen al que hace referencia.
Id_Pregunta	Integer	Pregunta a la que hace referencia.
Valor	Integer	Valor que se le va a otorgar a la pregunta en el examen, par su calificación.

Nombre: Perfil		
Descripción: Archivo correspondiente a un alumno registrado en el que se guardan los resultados por examen, así como el resumen de las deficiencias y fortalezas en el mismo.		
Atributo	Tipo	Descripción
Id_Persona	Integer	Persona a la que hace referencia.
Id_Examen	Integer	Examen al que hace referencia.
FechaExamen	DateTime	Fecha en el que se realiza en examen.
TiempoRespuestaPregunta	Integer	Tiempo promedio que demoró por pregunta.
TiempoRespuestaExamen	Integer	Tiempo promedio que demoró en responder el examen.
Calificacion	Integer	Calificación obtenida
Rango	Integer	Ubicación que se encuentra el usuario en comparación con otros que hayan realizado este mismo tipo de examen.

Nombre: Pregunta		
Descripción: Contiene los datos pertenecientes a la pregunta.		
Atributo	Tipo	Descripción
Id_Pregunta	Integer	Identificador de la pregunta.
Disponibilidad	Boolean	Si esta disponible o no.
Texto	String	Pregunta en cuestión.
Multimedia	String	Imagen o multimedia que se utilice para explicar la pregunta.
AclaracionRespuesta	String	Explicación adicional de la pregunta.
TiempoRespuesta	String	Tiempo en el que se le debe dar respuesta a la pregunta.
Objetivos	String	Objetivos que se persiguen con la pregunta.

CantExpertosEvaluadores	Integer	Cantidad de expertos que han emitido criterio o evaluado la pregunta.
Id_Tematica	Integer	Temática a la que hace referencia.
Id_Persona	Integer	Propietario. Persona a la que hace referencia.
Id_Prioridad	Integer	Prioridad a la que hace referencia.
Id_Complejidad	Integer	Complejidad a la que hace referencia.
Id_Longitud	Integer	Longitud a la que hace referencia.
Id_GComprension	Integer	Grado de Comprensión al que hace referencia.
Id_TipoPregunta	Integer	Tipo de pregunta a la que hace referencia.
Id_TipoRespuesta	Integer	Tipo de respuesta al que hace referencia.
Id_Peso	Integer	Peso al que hace referencia.

Nombre: Profesor		
Descripción: Almacena los profesores responsables.		
Atributo	Tipo	Descripción
Id_Persona	Integer	Identificador del profesor responsable.

Nombre: Respuesta		
Descripción: Almacena las respuestas q pertenecen a una pregunta.		
Atributo	Tipo	Descripción
Id_Respuesta	Integer	Identificador de la respuesta
Id_Pregunta	Integer	Pregunta a la que hace referencia
Respuesta	String	Texto de la respuesta.
Correcta	Boolean	Si la respuesta es correcta o no.

Nombre: Resultado		
Descripción: Tabla temporal, donde se almacena la nota de un usuario en una pregunta de un examen, y en caso de que la pregunta fuera abierta se guarda la respuesta del usuario en espera que el profesor la califique y le otorgue nota.		
Atributo	Tipo	Descripción
Id_Pregunta	Integer	Pregunta a la que hace referencia.
Id_Examen	Integer	Examen al que hace referencia.
Id_Persona	Integer	Persona a la que hace referencia.
Respuesta	String	Respuesta que da el usuario a la pregunta.
Nota	Integer	Nota parcial que se le otorga a la respuesta

Nombre: TipoPregunta		
Descripción: Codificador. Almacena el tipo de pregunta, puede ser teórica o práctica.		
Atributo	Tipo	Descripción
Id_TipoPregunta	Integer	Identificador del tipo de pregunta.
TipoPregunta	String	Nombre o descripción del tipo de pregunta.

Nombre: Usuario		
Descripción: Almacena los datos correspondientes a un usuario.		
Atributo	Tipo	Descripción
Id_Persona	Integer	Persona a la que hace referencia.
Nombre	String	Nombre de la persona.
Apellidos	String	Apellidos de la persona.
Login	String	Identificador asociado a los niveles de acceso dentro del sistema.
Password	String	Palabra clave para entrar al sistema.

Glosario de Términos.

ASP: (Active Server Pages) Páginas Activas del Servidor.

Base de Datos: (DataBase). Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc. Las bases de datos son uno de los grupos de aplicaciones de productividad personal más extendidos. Entre las más conocidas pueden citarse dBase, Paradox, Access y Approach, para entornos PC, y Oracle, ADABAS, DB/2, Informix o Ingres, para sistemas medios y grandes.

Browser: Navegador. Aplicación para visualizar documentos WWW y navegar por Internet. En su forma más básica son aplicaciones hipertexto que facilitan la navegación por los servidores de navegación de Internet. Los más avanzados, cuentan con funcionalidades plenamente multimedia y permiten indistintamente la navegación por servidores WWW, FTP, Gopher, acceso a grupos de noticias, la gestión del correo electrónico, etc.

CASE: (Computer Aided Software Engineering). Bajo el término de Ingeniería de Software Asistida por Ordenador se incluyen una serie de herramientas, lenguajes y técnicas de programación que permiten la generación de aplicaciones de manera semiautomática. Las herramientas CASE liberan al programador de parte de su trabajo y aumentan la calidad del programa a la vez que disminuyen sus posibles errores.

Casos de Uso: Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

CAT: (Computerized Adaptive Testing) Test Adaptativo Computarizado o Computadorizado.

Click: En algunos manuales de programas que utilizan intensivamente el ratón, se especifican instrucciones como «doble clic» que se refieren a apretar dos veces el botón izquierdo del ratón, acción que genera el ruido al que hace referencia esta definición.

Cliente: Cualquier elemento de un sistema de información que requiere un servicio mediante el envío de solicitudes al servidor.

Cuando dos programas se comunican por una red, el cliente es el que inicia la comunicación, mientras que el programa que espera ser contactado es el servidor. Cualquier programa puede actuar como servidor para un servicio y como cliente para otro.

Cliente Servidor: Modelo lógico de una forma de proceso cooperativo, independiente de plataformas hardware y sistemas operativos. El concepto se refiere más a una filosofía que a un conjunto determinado de productos. Generalmente, el modelo se refiere a un puesto de trabajo o cliente que accede mediante una combinación de hardware y software a los recursos situados en un ordenador denominado servidor.

C Sharp o C#: El lenguaje C# es una de las bases de la plataforma Microsoft.NET, que básicamente consiste en un C++ simplificado, pero intentando solucionar algunos problemas de Java.

C++: Versión de C orientada a objetos creada por Bjarne Stroustrup. C++ se ha popularizado porque combina la programación tradicional en C con programación orientada a objetos. Smalltalk y otros lenguajes originales de programación orientada a objetos no suministraban las estructuras familiares de lenguajes convencionales como C y Pascal.

DHTML: Dynamic HTML, son aplicaciones que contienen objetos y eventos y se procesan en el lado del cliente dentro del navegador Web.

Hardware: Conjunto de componentes materiales de un sistema informático. Cada una de las partes físicas que forman un ordenador, incluidos sus periféricos. Maquinaria y equipos (CPU, discos, cintas, modem, cables, etc.). En operación, un computador es tanto hardware como software. Uno es inútil sin el otro. El diseño del hardware especifica los comandos que puede seguir, y las instrucciones le dicen qué hacer. El hardware es "almacenamiento y transmisión".

HTML: (Hiper Text Transfer Protocol). Protocola de transferencia de HiperTexto. Es el protocolo de Internet que permite que los exploradores del WWW recuperen información distribuidos, colaborativos y de diferentes medios, de los servidores.

HTTP: (HyperText Markup Language). Lenguaje de marcado de Hipertexto. Es el lenguaje estándar para describir el contenido y la apariencia de las páginas en el WWW, es decir, es un conjunto de especificaciones para el intercambio de ficheros (texto, gráfico, imagen, sonido, vídeo) en la Web.

Internet: Red de ordenadores mundial que permite comunicación y transferencia de datos, noticias y opiniones entre personas y usuarios conectadas a ella.

Intranet: Es una red privada que pertenece a una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet, la cual puede ser accesible solo por sus miembros o empleados. Puede tratarse de una red aislada, es decir no conectada a Internet. La mayoría de las Intranet, están configuradas de forma que sus usuarios puedan tener acceso a Internet sin permitir que los usuarios de Internet tengan acceso a los equipos de la Intranet.

ITS: (Intelligent Tutoring Systems) Sistemas Inteligentes de Enseñanza.

Jscript o JavaScript: JavaScript, al igual que Java o VRLM, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML. JavaScript no es un lenguaje de programación propiamente dicho. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto. Nunca podrás hacer un programa con JavaScript, tan sólo podrás mejorar tu página Web con algunas cosas sencillas (revisión de formularios, efectos en la barra de estado, etc...) y, ahora, no tan sencillas (animaciones usando HTML dinámico, por ejemplo). JavaScript y Java son dos cosas distintas. Principalmente porque Java sí que es un lenguaje de programación completo. Lo único que comparten es la misma sintaxis.

Link: Apuntadores de Hipertexto que sirven para saltar de una información a otra, o de un servidor a otro, cuando se navega por Internet.

MAP: (Manufacturing Automation Protocol) Protocolo de automatización de manufactura. Protocolo de comunicaciones introducido por General Motors en 1982. MAP provee estándares comunes para la interconexión de computadores y máquinas herramientas programables usadas en la automatización de fábricas. En el nivel físico más bajo, MAP usa el protocolo IEEE 802.4 de bus de señales.

Con frecuencia, MAP se usa junto con TOP, un protocolo de oficina desarrollado por Boeing Computer Services. TOP se utiliza en la oficina y MAP, en la fábrica.

Microsoft: (Microsoft Corporation, Redmond, WA) Compañía de software más grande del mundo. Microsoft fue fundada en 1975 por Paul Allen y Bill Gates, dos estudiantes universitarios que escribieron el primer intérprete BASIC para el microprocesador 8080 de Intel. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

MVC: (Model-View-Controller) Modelo Vista Controlador.

NET: Red.

Oracle: Empresa especializada en la fabricación de programas de base de datos (en ordenadores).

PHP: (Preprocessed Hypertext Pages) Páginas de Hipertexto Preprocesadas. Fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, Personal Home Page Tools. Al principio, PHP sólo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico y se llamó PHP/F1 Versión 2, y sólo reconocía el texto HTML y algunas directivas de mSQL. A partir de este momento, la contribución al código fue pública.

RDBMS: (Relational Data Base) Administrador de bases de datos relacional.

Red: Se ha dicho muchas veces que el futuro de la informática está en las comunicaciones. Es una afirmación bastante obvia que hoy tiene ya sentido pleno. La intercomunicación entre ordenadores permite no sólo el intercambio de datos, sino también compartir recursos de todo tipo, optimizando así elevadas inversiones. Las redes son el soporte para estas conexiones y (aparte la diferenciación más genérica entre redes públicas y privadas), según el objeto de definición, la terminología es variada.

RUP: (Rational Unified Process) Proceso Unificado de Rational.

Servidor: Genéricamente, dispositivo de un sistema que resuelve las peticiones de otros elementos del sistema, denominados clientes. (Ver: Cliente/servidor).

SGBD: Sistema de Gestión de Bases de Datos.

Sistema: En informática, este término utilizado sin otra palabra que lo adjetive designa un conjunto de hardware y software específico.

Software: " El software se ocupa de los detalles de un negocio en constante cambio y debe procesar transacciones en una forma lógica. Los lenguajes se utilizan para programar el software. La lógica y el lenguaje involucrados en el análisis y la programación son por lo general mucho más complejos que especificar un requerimiento de almacenamiento y de transmisión. El software es "lógica y lenguaje.

SQL: (Structured query language) Lenguaje de preguntas estructurado, lenguaje que utiliza bases de datos para pedir información de las mismas.

Sun: Sun Microsystem Inc. Una de las firmas norteamericanas más importantes en la fabricación y comercialización de estaciones de trabajo, entre otros productos.

TADI: Tests Autoadaptados Informatizados

TAIM: Tests Adaptativos Informatizados multidimensional.

TAI: Tests Adaptativos Informatizados.

TRI: Teoría de la Respuesta al Ítem.

UML: (Unified Modeling Language) Lenguaje de Modelación Unificado. Es una notación Standard para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

VB: Visual Basic. Versión de BASIC de Microsoft utilizado para desarrollar aplicaciones de Windows, que se ha vuelto popular. Es similar a QuickBASIC de Microsoft, pero no es 100% compatible con éste. Las interfaces de usuario se desarrollan llevando objetos de la caja de herramientas de Visual Basic hacia el formato de aplicación.

Web o WWW: (World Wide Web) Telaraña o malla mundial. Sistema de información con mecanismos de hipertexto creado por investigadores del CERN. Los usuarios pueden crear, editar y visualizar documentos de hipertexto. También llamado W3.

XML: (Extensible Markup Language) Es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. En la práctica corresponde a un estándar que permite a diferentes aplicaciones interactuar con facilidad a través de La Red. XML. Está orientado al almacenamiento, procesamiento y transmisión de mensajes. La primera definición que apareció fue: Sistema para definir validar y compartir formatos de documentos en la Web.