

**Universidad de Sancti-Spíritus “José Martí Pérez”**



**Facultad de Ingeniería  
Ingeniería Informática**

**Trabajo de Diploma para optar por el Título de  
Ingeniería en Informática**

**Título**

Tutorial para el aprendizaje del Sistema de Restricciones del tema  
Programación Lineal

**Autor**

Adrian Rodríguez Sanz.

**Tutor**

Msc. Aurelio Lorenzo Hernández Reyes

**Curso 2009-2010**

*“Todo esfuerzo por difundir la instrucción es vano, cuando no se acomoda la enseñanza a las necesidades, naturaleza y porvenir del que la recibe”.*

*José Martí*

## **Agradecimientos**

- A mi familia por su apoyo incondicional.
- A mis compañeros de aula por estos 5 años de experiencias.
- A mi tutor, por su amistad y sus sabios consejos.
- A mis amigos especialmente a Lisandra y Borges que han sido fundamentales para el éxito de este trabajo.
- A todos mis profesores por los conocimientos adquiridos.
- A mis suegros, quienes me han hecho sentir parte de su familia.
- A mi novia Danays que me ha enseñado que para alcanzar grandes cosas debemos no solo actuar sino soñar, no solo planear sino creer.
- A todos aquellos que de una manera u otra han contribuido con su apoyo.

## Resumen

La investigación que se presenta tiene como título ***“Tutorial para el aprendizaje del Sistema de Restricciones en el tema Programación Lineal”*** y surge como respuesta al déficit de materiales educativos computarizados que traten el tema de la Programación Lineal. El trabajo tiene como principal objetivo contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones del tema Programación Lineal de los estudiantes de Contabilidad y Finanzas de la UNISS. El Tutorial tiene como base fundamental la Programación Lineal específicamente el Sistema de Restricciones. Cuenta con una interfaz gráfica donde intervienen un menú principal y una barra de herramientas desde las cuales se puede controlar toda la funcionalidad del Tutorial. Funcionará de manera interactiva, o sea, que el estudiante tendrá la posibilidad de revisar la teoría, ver ejercicios resueltos, resolver ejercicios propuestos, además tiene la posibilidad de resolver un autoexamen y obtener una calificación a la cual el estudiante y el profesor pueden acceder. La bibliografía revisada para la realización de esta investigación es escasa y solo existe de forma actualizada en Internet. Para los conceptos teóricos que se abordan se consultaron además Trabajos de Diploma, textos de metodología, revistas electrónicas y otros.

## **Abstract**

The investigation that appears next is entitled “ Tutorial for the education learning of the Restrictions System in the topic Linear Programming ” and arises as response to few existence of Digital Educational Materials, capable of achieving the improvement of the process of education learning of the students of Accounting and Finance of the UNISS. The work takes as a principal target to support the process of education learning of the Restrictions System in the topic Linear Programming for the students of Accounting and Finance of the UNISS. It is provided with a graphic interface which consist of a principal menu and a toolbar from which it is possible to gain access to all the functionalities of the same one, it will work in an interactive way, meaning this, that the student will have the possibility of checking the theory, seeing solved exercises, solving proposed exercises, also it has the possibility of solving an autoexamination and of obtaining a qualification to which the user and the teacher can gain access to see their performance. The bibliography checked for the achievement of this investigation is scarce and only it exists in updated form in Internet. For the theoretical concepts that are tackled there were consulted also Works of Diploma, texts of methodology, electronic magazines among others.

## Indice

Introducción.....	1
Capítulo 1: Fundamentación Teórica .....	5
1.1 Introducción.....	5
1.2 Las Tecnologías de la Información y las Comunicaciones (TIC).....	5
1.3 Las Tecnologías de la Información y la Comunicación en la educación superior .....	6
1.4 El Proceso de enseñanza aprendizaje .....	8
1.5 El Software en la educación .....	11
1.5.1 Sistemas Tutoriales .....	11
1.5.3 Libros electrónicos.....	12
1.5.4 Sistemas Tutoriales Inteligentes .....	13
1.5.5 Sistemas Hipermedia.....	14
1.6 Breve historia de la Programación Lineal.....	15
1.7 Concepto de Programación Lineal .....	16
1.8 Construcción del sistema de restricciones .....	16
1.9 El proceso de enseñanza de la Programación Lineal en la Universidad de Sancti Spíritus .....	17
1.10 Causas que originan la situación problemática y las consecuencias .....	18
1.11 Análisis comparativo de otras soluciones existentes con la propuesta .....	20
1.12 Fundamentación de la metodología utilizada .....	21
1.12.1 Proceso Unificado de Desarrollo (RUP).....	21
1.12.2 Lenguaje de Modelamiento Unificado (UML).....	23
1.12.3 Arquitectura de desarrollo de N Capas.....	25
1.13 Tecnologías y Herramientas.....	27
1.13.1 Borland Delphi 7.0 .....	27
1.13.2 Sistemas Gestores de Base de Datos (SGDB).....	27
1.13.3 SQL (Structured Query Language) .....	28
1.13.4 SQL (Structured Query Language) Server 2000.....	28
1.13.5 Hiper Text Markup Lenguaje (HTML).....	29
1.14 Conclusiones.....	30
Capítulo 2: Modelo del Dominio .....	31
2.1 Introducción.....	31
2.2 Descripción del modelo de dominio.....	31
2.2.1 Principales conceptos asociados al dominio del problema. ....	32
2.2.2 Modelo de objetos del dominio .....	32
2.3 Reglas a considerar .....	33
2.4 Requisitos funcionales.....	33
2.5 Requisitos no funcionales.....	35
2.6 Descripción del sistema propuesto.....	38
2.7 Modelo de Casos de Uso del Sistema .....	39
2.7.1 Actores del sistema.....	39
2.7.2 Casos de Uso del sistema .....	40
2.7.3 Descripción de los casos de usos del sistema.....	43
2.8 Conclusiones.....	58
Capítulo 3: Construcción de la Solución Propuesta .....	59

3.1	Introducción.....	59
3.2	Diagrama de clases del diseño .....	59
3.3	Diseño de la base de datos .....	61
3.3.1	Diagrama de clases persistentes.....	61
3.3.2	Modelos de datos .....	62
3.4	Principios del diseño .....	62
3.4.1	Estándares en la interfaz de la aplicación.....	62
3.4.2	Formatos de reportes.....	63
3.4.3	Concepción general de la ayuda .....	64
3.4.4	Tratamiento de excepciones.....	64
3.4.5	Seguridad .....	65
3.4.6	Estándares de codificación .....	65
3.5.	Diagrama de Componentes.....	66
3.6	Modelo de despliegue .....	67
3.7	Conclusiones .....	68
	Capítulo 4: Estudio de la Factibilidad .....	69
4.1	Introducción.....	69
4.2	Planificación .....	69
4.3	Costos .....	72
4.4	Beneficios intangibles.....	74
4.5	Análisis de costos y beneficios.....	75
4.6	Conclusiones .....	76
	<b>Conclusiones</b> .....	77
	Recomendaciones.....	78
	Bibliografía .....	79
	<b>Anexos</b> .....	83

## **Introducción**

El nombre de programación lineal no procede de la creación de programas de ordenador, sino de un término militar, programar, que significa 'realizar planes o propuestas de tiempo para el entrenamiento, la logística o el despliegue de las unidades de combate'.

La programación lineal surge en el período de la segunda guerra mundial. Este se origina como un modelo matemático creado con el objetivo de planificar los gastos de guerra y maximizar las pérdidas del enemigo.

Aunque parece ser que la programación lineal fue utilizada por G. Monge en 1776, se considera a L. V. Kantoróvich uno de sus creadores. La presentó en su libro Métodos matemáticos para la organización y la producción (1939) y la desarrolló en su trabajo Sobre la transferencia de masas (1942). Kantoróvich recibió el premio Nobel de economía en 1975 por sus aportaciones al problema de la asignación óptima de recursos humanos.

En años posteriores a la Segunda Guerra Mundial, en Estados Unidos se asumió que la eficaz coordinación de todas las energías y recursos de la nación era un problema de tal complejidad, que su resolución y simplificación pasaba necesariamente por los modelos de optimización que resuelve la programación lineal.

Paralelamente a los hechos descritos se desarrollan las técnicas de computación y los ordenadores, instrumentos que harían posible la resolución y simplificación de los problemas que se estaban gestando.

Se ha estimado, de una manera general, que si un país subdesarrollado utilizase los métodos de la programación lineal, su producto interior bruto (PIB) aumentaría entre un 10 y un 15% en tan sólo un año.

### **Situación Problemática:**

Durante años los estudiantes de Contabilidad y Finanzas de la Universidad “José Martí Pérez” de Sancti Spíritus (UNISS) reciben la Programación Lineal; tanto en el Curso



Regular Diurno como en el Curso para Trabajadores. Estos estudiantes para el aprendizaje de la misma sólo cuentan con la bibliografía existente, la cual suele ser escasa y a veces se dificulta su consulta. Los estudiantes no tienen un medio de apoyo al aprendizaje de la Programación Lineal que no sea una Multimedia de la CUJAE, la bibliografía existente y otros artículos en la red. Esta multimedia aborda el tema de manera precisa; pero no permite la actualización de los contenidos. Además se dificulta mucho la revisión en Internet para los estudiantes. También podemos decir que desde el año 1998 hasta el año 2009 solo 7 estudiantes de 392, realizaron su trabajo para optar por el título de Licenciado en Contabilidad y Finanzas abordando temas sobre la Programación Lineal; lo cual representa el 1,8% de los estudiantes.

**Problema científico:**

¿Cómo contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones del tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la Universidad “José Martí Pérez” de Sancti Spíritus?

**Objeto de estudio:** Proceso de enseñanza aprendizaje de la Programación Lineal.

**Campo de acción:** Proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la UNISS.

**Objetivo general:** Diseñar e Implementar un Tutorial para contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones del tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la Universidad “José Martí Pérez” de Sancti Spíritus.

Para alcanzar el objetivo general propuesto se formularon las siguientes **Preguntas Científicas:**

1. ¿Cuáles son los fundamentos teóricos-metodológicos que sustentan el uso del Tutorial para contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal?

2. ¿Cómo diseñar un Tutorial basado en una metodología para el proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal y contribuir al aprendizaje de los estudiantes de Contabilidad y Finanzas del de la UNISS?
3. ¿Cómo realizar un análisis de los costos-beneficios del Tutorial?
4. ¿Cómo implementar un Tutorial para contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la UNISS?

Para dar cumplimiento a la investigación se tienen en cuenta las siguientes **Tareas de Investigación:**

2. Determinación de los fundamentos teóricos-metodológicos que sustentan el uso del Tutorial para contribuir al proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal.
3. Diseño de un Tutorial basado en una metodología para el proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la UNISS.
4. Realización de un análisis de los costos y beneficios del Tutorial.
5. Implementación un Tutorial para apoyar el proceso de enseñanza aprendizaje del Sistema de Restricciones en el tema Programación Lineal para los estudiantes de Contabilidad y Finanzas de la UNISS.

El presente trabajo está estructurado en cuatro capítulos.

**Capítulo 1:** Se abordan la fundamentación teórica del tema y los conceptos asociados al dominio del problema. Se explica en detalles el problema a resolver y se describe el

campo de acción donde se desarrolla. Así como las tecnologías y metodologías utilizadas para su desarrollo teniendo en cuenta las tendencias actuales en el desarrollo de las NTIC.

**Capítulo 2:** Se centra fundamentalmente en analizar el modelo de dominio, así como la descripción de dicho proceso utilizando los artefactos de UML y tomando como guía la metodología RUP. Se describen los requerimientos funcionales y no funcionales, el diagrama de casos de uso y la descripción de cada uno, los cuales ayudan a modelar y describir la solución propuesta.

**Capítulo 3:** Se muestran los diagramas de clases del sistema a construir, con su correspondiente descripción. Se describe el diseño de la solución propuesta con sus correspondientes modelos y/o diagramas: modelo de clases, modelo de clases persistentes, modelo de implementación y los principios que cumple el diseño.

**Capítulo 4:** Se describe lo relacionado con la planificación, costo, beneficios intangibles, análisis de costo y beneficios en el desarrollo de la aplicación a desarrollar.

## **Capítulo 1: Fundamentación Teórica**

### **1.1 Introducción**

En el presente capítulo se brinda una fundamentación teórica acerca del tema a desarrollar. Se expone, fundamentalmente una descripción del tema que trata el Tutorial: “El Sistema de Restricciones del tema de Programación Lineal”. También se describen las características fundamentales de algunas de las tecnologías utilizadas para el desarrollo del Tutorial exponiendo los principales conceptos asociados al dominio del mismo.

### **1.2 Las Tecnologías de la Información y las Comunicaciones (TIC)**

La utilización de tecnología, nuevas tecnologías, nuevas tecnologías de la información y la comunicación o, simplemente, tecnologías de la información y la comunicación se ha hecho de forma indistinta, y centrándose más en el espacio temporal que en las características de los instrumentos utilizados, para referirse a concepciones o ámbitos de estudio similares. (Baelo Álvarez & Cantón Mayo, 2009)

“Podemos entender por nuevas tecnologías a todos aquellos medios de comunicación y de tratamiento de la información que van surgiendo de la unión de los avances propiciados por el desarrollo de la tecnología electrónica y las herramientas conceptuales, tanto conocidas como aquellas otras que vayan siendo desarrolladas como consecuencia de la utilización de estas mismas nuevas tecnologías y del avance del conocimiento humano”. (Tejedor & García Valcárcel)

(Majó & Marqués, 2002), profundizaban en la composición de las TIC haciendo referencia a tres campos; la informática, las telecomunicaciones y las tecnologías de la imagen y el sonido.

De forma similar Haag, Cummings y McCubbrey (2004), consideraban que las tecnologías de información están compuestas de “cualquier herramienta basada en los ordenadores y que la gente utiliza para trabajar con la información, apoyar a la información y procesar las necesidades de información”. (Haag, Cummings, & McCubbrey, 2004)

Desde una perspectiva institucional la OCDE (2002) define las TIC como “aquellos dispositivos que capturan, transmiten y despliegan datos e información electrónica y que apoyan el crecimiento y desarrollo económico de la industria manufacturera y de servicios”. (OCDE(Organisation for Economic Co-operation and Development, 2002)

Las TIC son una realización social que facilitan los procesos de información y comunicación, gracias a los diversos desarrollos tecnológicos, en aras de una construcción y extensión del conocimiento que derive en la satisfacción de las necesidades de los integrantes de una determinada organización social. (Baelo Álvarez & Cantón Mayo, 2009)

### **1.3 Las Tecnologías de la Información y la Comunicación en la educación superior**

Los acelerados procesos de cambio tecnológico y cultural, principalmente del último medio siglo, así como la exigencia del desarrollo de una nueva configuración de la educación superior, han llevado a las universidades a un proceso crítico de reestructuración y revisión general que apuesta por la flexibilidad en los conocimientos impartidos, la formación continuada y la diversificación de metodologías y formas de desarrollo de las mismas.

Dentro de este proceso de reinención de la universidad las TIC cobran un importante protagonismo, conformándose como elementos esenciales para el desarrollo de la flexibilidad organizativa de las enseñanzas y el desarrollo de nuevas sinergias que inserten plenamente a la universidad en el actual entramado de las sociedades del conocimiento. Las experiencias desarrolladas en este sentido auguran un futuro prometedor para aquellas instituciones de educación superior que integren, de forma efectiva, las TIC en sus procesos y estructuras, puesto que de forma general podemos indicar que:

- La introducción y el uso de las TIC en los procesos de enseñanza-aprendizaje y de gestión en la educación superior ha supuesto una transformación institucional orientada a dar respuesta a las necesidades demandadas por las sociedades del conocimiento.
- Las TIC han incidido en todos los campos relacionados con la educación, facilitando la transformación y la optimización de la mayoría de los procesos administrativos, el

desarrollo de metodologías innovadoras en los procesos de enseñanza-aprendizaje, el acceso a la formación superior de nuevos grupos de personas, así como una incipiente transformación del sistema organizativo de las universidades (Gayle, Tewarie y White, 2003).

- Las nuevas exigencias en la educación superior se centran en la mejora del proceso educativo y, en este sentido, la integración de las TIC facilita aspectos relacionados con la mejora del trabajo individual, la autonomía del alumnado, la facilidad para el desarrollo de trabajos en equipo y colaborativos, la posibilidad de modificar y adaptar los métodos de evaluación y la interacción bidireccional entre el profesorado y el alumnado.

De igual forma, las experiencias desarrolladas nos permiten señalar como las TIC están cobrando un mayor protagonismo en el desarrollo de la educación superior, convirtiéndose en elementos esenciales para el desarrollo de la actividad docente e investigadora. A este respecto se hace necesario el desarrollo, dentro de las instituciones universitarias, de medidas que, cómo indican Nieto y Rodríguez (2007) y Barro et al. (2004), favorezcan la integración efectiva de las TIC, unas medidas que se deben de desarrollar en torno a los siguientes aspectos:

- La formación, capacitación técnica y pedagógica del profesorado, desarrollando políticas de alfabetización digital entre los docentes para evitar la profundización de la brecha digital docente.
- La motivación y el reconocimiento. Es necesaria una apuesta clara y decidida por unos sistemas de recompensa que permitan reconocer el esfuerzo desarrollado por el profesorado que lleva a cabo actividades formativas y de integración de las TIC en su práctica habitual.
- La inversión; tanto para la adquisición de los medios tecnológicos en la cantidad, calidad y grado de accesibilidad adecuado, como para el desarrollo de centros de apoyo especializados, que den respuesta a las necesidades del docente, además de desarrollar y fomentar actividades formativas y de innovación docente. (Baelo Álvarez & Cantón Mayo, 2009)

#### **1.4 El Proceso de enseñanza aprendizaje**

“Las reformas educativas, iniciadas en la última década, han insistido en considerar a los alumnos como sujetos activos en la construcción de conocimientos, en la necesidad de promover aprendizajes en sentido amplio y en asignar un nuevo rol al docente como mediador y facilitador del aprendizaje.” (UNESCO, 2003)

Así el aprendizaje lo definen como:

“aquel que garantiza en el individuo la apropiación activa y creadora de la cultura, propiciando el desarrollo de su auto-perfeccionamiento constante, de su autonomía y autodeterminación, en íntima conexión con los necesarios procesos de socialización, compromiso y responsabilidad social.” (Castellanos & otros, El proceso de enseñanza-aprendizaje desarrollador en la Secundaria Básica, 2001)

Así se recoge en “La Enseñanza Problémica” de Majmutov un concepto de aprendizaje enfocado a través de los procesos psíquicos que transcurren: “Según nuestra opinión, la definición más plena del concepto de aprendizaje del hombre, la formuló el psicólogo checo I.Lingart ‘El aprendizaje en el sentido más amplio de esta palabra, lo consideramos un tipo de actividad en la que el sujeto modifica, en una situación dada y bajo la influencia de las condiciones externas, así como en dependencia de los resultados de la actividad propia, su conducta y sus procesos psíquicos, de forma tal que las nuevas informaciones disminuyen el grado de su inseguridad y halla una respuesta correcta o una regla adecuada de conducta’. (Majmutov, 1983)

Consecuente con su definición Lingart distingue varios tipos de aprendizaje: verbal, cognoscitivo y durante la resolución de problemas. Más adelante el propio Majmutov da su criterio pero más próximo de la actividad docente al decir: “La esencia del aprendizaje desde el punto de vista de los objetivos instructivos-educativos, la vemos en la asimilación activa de la experiencia social por el alumno, en la adquisición de conocimientos científicos y de modos de actividad práctica y teórico-intelectual, que condiciona el desarrollo multilateral de la personalidad”

Muy relacionada con esta definición, Danilov declara que: "Los didactas soviéticos (...) han elaborado la concepción del aprendizaje como desarrollo mental e intelectual ininterrumpido de los educandos durante el proceso de aprendizaje, durante la asimilación (apropiación) activa y consciente de los conocimientos sobre los fundamentos de las ciencias y su aplicación práctica" (Danilov, 2003)

El papel activo del estudiante se ratifica por Labarrere al señalar que: "(...) Un tercer componente lo encontramos en el sistema de aprendizaje, es decir, en el alumno. El escolar resulta al mismo tiempo objeto y sujeto de la educación. Lo primero está dado por constituir el elemento sobre el cual recae la acción educadora del maestro. Lo segundo se explica al considerar que el alumno es a la vez sujeto activo del aprendizaje" (Labarrere, 1988)

En un sentido más amplio, Álvarez de Zayas, define que: "El aprendizaje es la actividad que desarrolla el estudiante para aprender, para asimilar la materia de estudio (...)el estudiante es el sujeto de su propio aprendizaje". (Álvarez de Zayas, 1999)

En este enfoque, como actividad que realiza el estudiante, Suanes señala que "aprendizaje es la actividad reflejo-transformadora, motivada y autorregulada, orientada al logro de un objetivo, mediante la cual el sujeto en una comunicación variada se apropia de la experiencia histórico social".

Resultó muy valioso, para este autor, el análisis que realiza González sobre el aprendizaje al ubicarlo como un proceso humano complejo" (...) el cual se expresa como configuración subjetiva y proceso interactivo". Al describir a continuación el contenido de estos dos conceptos, los cuales son compartidos por el autor, se señala que:

Ubicar el aprendizaje en la subjetividad, implica explicarlo como:

- Proceso diferenciado a escala individual.
- Expresión integral del sujeto resultante de la configuración de un conjunto de elementos psicológicos diferentes.
- Resultado de una integración funcional de lo cognitivo y lo afectivo.
- Expresión intencional de un sujeto interesado en aprender.



Ubicar el aprendizaje en un marco interactivo, implica reconocer el:

- Carácter necesario de la comunicación en el proceso de construcción del conocimiento.
- Papel de una atmósfera institucional sana en el desarrollo de las potencialidades del sujeto para el aprendizaje.
- Aprendizaje como un proceso de cooperación, de integración.
- Sentido que para el aprendizaje tiene el bienestar emocional del sujeto en sus distintos sistemas de relaciones.

Expresa finalmente sobre el aprendizaje que: “El aprendizaje, entonces, tenemos que representárnoslo como un proceso activo e integral del sujeto en la construcción del conocimiento”. Sobre esta construcción advierte “ni tampoco como construcción solo cognitiva”.

“Por otra parte, desde el punto de vista didáctico, la consideración del grupo como un espacio de aprendizaje, supone una visión diferente y cualitativamente superior del diseño de las tareas de aprendizaje, pues no se trata ya de la limitada relación dicotómica entre la atención a *todos* los alumnos del grupo y la atención diferenciada a *cada* miembro del mismo. Se trata de utilizar este espacio grupal como un componente del proceso que debe ser tenido en cuenta en su diseño y ejecución, como una *herramienta* para la atención a la diversidad. Este principio pedagógico se constituye en eje estructurador de la organización del proceso.”

En resumen, para la creación de situaciones de aprendizaje se deben tener en cuenta los siguientes principios:

- La promoción de una construcción activa y personal del conocimiento por parte de los/las estudiantes.
- La unidad de afecto y cognición a través de un aprendizaje racional y afectivo-vivencial.
- Las oportunidades para trabajar en grupo y realizar un aprendizaje cooperativo.
- El respeto a la individualidad, a los intereses, particularidades y necesidades de los/las estudiantes desde la flexibilidad y diversidad en objetivos específicos, contenidos, métodos, estrategias y situaciones educativas-
- La posibilidad de aprender a través de actividades desafiantes que despierten las motivaciones intrínsecas.

- La participación y solución en problemas reales, contextualizados, que permitan explorar, descubrir y hacer por transformar la realidad.
- La transformación del (de la) estudiante de receptor en investigador y productor de la información
- La promoción del autoconocimiento, de la autovaloración y de la reflexión acerca del proceso de aprendizaje.
- La valoración de la autodirectividad y la autoeducación como meta.

El centro en los cuatro planes básicos de la educación: aprender a conocer, a hacer, a convivir, y a ser”. (Castellanos & otros, Aprender y enseñar en la escuela: una concepción desarrolladora, 2002)

## **1.5 El Software en la educación**

### **1.5.1 Sistemas Tutoriales**

García D. plantea que “constituye un programa especializado en la enseñanza de un dominio específico del conocimiento, apoyándose para ello en el diálogo con el estudiante, en la consolidación de un conjunto de aspectos esenciales que por su complejidad requieren de un nivel de abstracción que permita la representación adecuada del conocimiento” (García & López)

Esta definición es retomada por Rodríguez, R. para puntualizar que: “el tutorial es un programa especializado en un área del conocimiento, que establece una estrategia basada en el diálogo, está de acuerdo a las características del estudiante y además, existe una estrategia pedagógica para guiar a este estudiante”. (Rodríguez, 1991)

Los programas tutoriales están diseñados para la enseñanza, a través del uso de la computadora, es por eso que se utiliza el término de formación basada en ordenador. En este tipo de tutoriales se utilizan colores, gráficos y otros elementos para atraer la atención y mantener el nivel de interés. Sus usos son variados, por ejemplo, un programador de software puede incluir una serie de lecciones dentro de una aplicación para que los usuarios se familiaricen con el programa; un asesor puede utilizar un tutorial más largo y detallado como herramienta para un seminario de formación.

Los tutoriales pueden contener ejercicios y sesiones de preguntas y respuestas para presentar un tema y verificar su comprensión por parte del estudiante, permitiéndole también estudiar a su propio ritmo. Los tutoriales actuales, no sólo admiten respuestas más abiertas, sino que su secuenciación es tal que permiten distintos desarrollos en función del aprendizaje del alumno; en determinados momentos del programa se introducen elementos de evaluación cuyo resultado orienta al alumno a pasar a una lección siguiente o a repasar una anterior, proponiéndole, incluso, ejercicios de refuerzo. (Microsoft Corporation, 1993-2005)

### **1.5.2 Sistemas Entrenadores**

Pérez y otros (2000), los definen como “software diseñado con el propósito de desarrollar una determinada habilidad, específicamente una habilidad manual o motora, en el estudiante que lo emplea”. Para otros autores como Rodríguez, se definen los entrenadores como “un software educativo diseñado con el propósito de contribuir al desarrollo de una determinada habilidad, intelectual, manual o motora, en el estudiante que lo utiliza, por lo que profundizan en las dos fases finales del aprendizaje: aplicación y retroalimentación” (Rodríguez y otros, 2000). (REDVET, 2009)

### **1.5.3 Libros electrónicos**

Es un mecanismo electrónico que permite almacenar una gran cantidad de información de tipología distinta. Esta información está organizada conceptualmente del mismo modo que las de uno de papel, pero éste permite interactuar. Las herramientas que incorpora el aparato electrónico aumentan notablemente sus funciones y sus posibilidades al permitir integrar a la vez texto, sonido, imágenes, permiten realizar búsquedas, unir textos, etc. Permiten incluso subrayar párrafos o hacer anotaciones al margen, con lo que permiten más funciones que permite un libro tradicional y además potenciadas y mejoradas.

Estos mecanismos electrónicos tienen el potencial de ampliar las posibilidades que nos ofrece el libro tradicional. Además la relación, o mejor dicho, interacción que tendremos con él será tan diferente que nos cambiará el comportamiento en el tan humano acto de leer y de cómo accedemos a la cultura. Esto es debido a que:

1. Tiene una capacidad de almacenamiento enorme. Permite transportar en ejemplar electrónico el equivalente a estanterías completas de sus análogos de papel.
2. Se trata de un documento electrónico, con lo que permite incluir junto al texto sonidos, imágenes, incluso en movimiento.
3. Además, como en los otros libros se pueden subrayar palabras, párrafos e incluso hacer anotaciones en los márgenes.
4. Y algo que es muy novedoso y que nos demuestra que no son meras simulaciones de los libros impresos es que; gracias a los hiperenlaces, el texto pierde la linealidad del libro tradicional. Al encontrarnos en el texto una palabra resaltada, con un click nos transportará a otra parte del texto o a otro documento con el que tenga relación.

Los documentos electrónicos permiten mantener la información que contienen permanentemente actualizada de un modo fácil; esto es especialmente útil en la comunicación científica. En la actualidad existen revistas que son editadas a la vez en papel y en formato digital. Por supuesto las segundas permiten un acceso mucho más rápido. En el futuro, el tipo de documentos de corta vida como los periódicos podrían enviar la edición diaria a sus suscriptores de manera inalámbrica. (Pérez Arranz, 2001)

#### **1.5.4 Sistemas Tutoriales Inteligentes**

Sistemas capaces de enseñar a un usuario un determinado asunto a través de la evaluación más frecuente del alumno, estos sistemas surgieron de la aplicación de la Inteligencia Artificial en la Educación y vinieron a sustituir a la tradicional instrucción asistida por computadora. Los STI deben presentar tres características que denotan inteligencia. El sistema debe:

- Ser capaz de hacer inferencias sobre el dominio y resolver problemas que estén en su esfera de actuación.
- Ser capaz de ayudar al alumno.
- Favorecer estrategias pedagógicas que minimicen la diferencia entre el aprendiz y el especialista.

(Pacios Fernández, Arenas Gutiérrez, & Pérez Llanes, 2010)

La arquitectura clásica de un Sistema Tutorial Inteligente (STI), está compuesta principalmente de cuatro módulos: tutor, dominio de aplicación, interfaz y modelo del estudiante.

El módulo del tutor es el encargado de determinar a quién, cómo, cuándo y qué se enseña o se evalúa. El módulo del dominio de aplicación se refiere al tema o curso en el que se requiere la capacitación. El módulo interfaz es en el que se lleva a cabo la interacción estudiante-STI, debe ser atractivo y capaz de retener la atención del usuario. Finalmente, el módulo del modelo del estudiante es el encargado de determinar en todo momento el grado de conocimiento que tiene el estudiante sobre el tema que está aprendiendo. (Romero J & Martínez R)

### **1.5.5 Sistemas Hipermedia**

Sistemas capaces de almacenar un documento y permitir la lectura de este en forma no convencional o no lineal.

Los sistemas hipermedia no pueden ser considerados como educacionales en el sentido propio de la palabra, pues aunque transmiten información no poseen métodos para determinar el grado de instrucción alcanzado, y tampoco permiten la individualización de la instrucción. Otra posibilidad es que un usuario inexperto, se pierda en el hiperespacio, o se desvíe de su área de interés.

El uso de hipermedia permite grandes ventajas como son:

- La libertad de navegación para el alumno, lo que le permite decidir el flujo de su aprendizaje, obteniendo información que desea en el orden que prefiera mientras navega por la red de información. El alumno puede dirigir su aprendizaje hacia los conceptos y aspectos de mayor interés para él, profundizando en las materias que necesite dominar mientras sobrevuela otras materias secundarias.
- La presentación de información hipermedia (texto, audio, video, gráfico) resulta más enriquecedora y amena, lo que incrementa la usabilidad y atractivo del

sistema de cara a los alumnos que lo utilizan. Aunque para ello se necesita un diseño adecuado a fin de evitar que una presentación excesivamente colorista o llamativa desvíe la atención del alumno desde el mensaje al medio.

- El concepto hipermedia permite, mediante relaciones y la organización no secuencial de informaciones, una implementación sencilla de las estructuras conceptuales, generalmente de semántica compleja, de los dominios a enseñar.

Pero no dejan de ver las desventajas:

- El riesgo de que el alumno se pierda en la red de informaciones. Si el dominio es demasiado extenso o detallado, la libertad de navegación puede hacer que el alumno no alcance las materias que le interesan, o que deje de estudiar otras debido a que no sabe siquiera que existan en el hiperespacio, o que desespere por no saber en qué momento ha aprendido ya todo lo que necesita.
- Desde el punto de vista pedagógico, no hay posibilidad de evaluar la instrucción recibida por el alumno, ni de adaptar la información al nivel de conocimientos del mismo, lo que disminuye el potencial didáctico del uso de las hipermedias. La información que contienen las hipermedias tradicionales es estática. (Pacios Fernández, Arenas Gutiérrez, & Pérez Llanes, 2010)

## **1.6 Breve historia de la Programación Lineal**

El problema de la resolución de un sistema lineal de inecuaciones se remonta, al menos, a Fourier, después de quien nace el método de eliminación de Fourier-Motzkin. La programación lineal se plantea como un modelo matemático desarrollado durante la segunda guerra mundial para planificar los gastos y los retornos, a fin de reducir los costos al ejército y aumentar las pérdidas del enemigo. Se mantuvo en secreto hasta 1947. En la posguerra, muchas industrias lo usaron en su planificación diaria.

Los fundadores de la técnica son George Dantzig, quien publicó el algoritmo simplex, en 1947, John von Neumann, que desarrolló la teoría de la dualidad en el mismo año, y Leonid Kantorovich, un matemático ruso, que utiliza técnicas similares en la economía antes de Dantzig y ganó el premio Nobel en economía en 1975. Leonid Khachiyan en 1979 fue el primero en demostrar que el problema de la programación lineal se solucionaba en tiempo polinomial, sin embargo, el mejor avance en los principios teóricos

y prácticos en el campo se produjo en 1984, cuando Narendra Karmarkar introduce un nuevo método del punto interior para resolver problemas de programación lineal.

El ejemplo original de Dantzig de la búsqueda de la mejor asignación de 70 personas a 70 puestos de trabajo es un ejemplo de la utilidad de la programación lineal. La potencia de computación necesaria para examinar todas las permutaciones a fin de seleccionar la mejor asignación es inmensa; el número de posibles configuraciones excede al número de partículas en el universo. Sin embargo, toma sólo un momento encontrar la solución óptima mediante el planteamiento del problema como una programación lineal y la aplicación del algoritmo simplex. La teoría de la programación lineal reduce drásticamente el número de posibles soluciones óptimas que deberán ser revisadas.

### **1.7 Concepto de Programación Lineal**

Procedimiento o algoritmo matemático mediante el cual se resuelve un problema indeterminado, formulado a través de ecuaciones lineales, optimizando la función objetivo, también lineal.

La programación lineal consiste en optimizar (minimizar o maximizar) una función lineal, que denominaremos función objetivo, de tal forma que las variables de dicha función estén sujetas a una serie de restricciones que expresamos mediante un sistema de inecuaciones lineales.

La programación lineal es un procedimiento que encuentra su aplicación práctica en casi todas las facetas de los negocios, desde la publicidad hasta la planificación de la producción. Problemas de transporte, distribución, y planificación global de la producción son los objetos más comunes del análisis de programación lineal.

### **1.8 Construcción del sistema de restricciones**

El sistema de ecuaciones y/o inecuaciones junto con la condición constituyen las limitaciones que forman el conjunto posibles de decisiones a tomar, ya que, en la programación lineal no se optimiza la función objetivo en general, sino sujeta a

restricciones que hay que respetar. Dado que el modelo es lineal, todas y cada una de las restricciones tienen que cumplir los requisitos de la linealidad.

### **1.9 El proceso de enseñanza de la Programación Lineal en la Universidad de Sancti Spíritus**

La UNISS, como universidad territorial universalizada sustentable tiene como misión formar de manera continua profesionales integrados, revolucionarios, tecnológicos, artísticos; fieles a las tradiciones e historia y con voluntad de innovación.

La Universidad de Sancti-Spíritus tiene como visión consolidar la comunidad universitaria con un claustro altamente categorizado, comprometido con la revolución y con los valores que comparten, asumiendo la prevención y el control de ilegalidades dentro de la cultura organizacional. Satisfacer las necesidades de las Sedes Universitarias Municipales, logrando perfeccionamiento y consolidación. Aumentar la calidad de la formación integral de los estudiantes con la acreditación y crecimiento del número de carreras. La gestión de postgrado satisface las necesidades del territorio. Se amplía la efectividad de la colaboración internacional. Se incrementa el trabajo por proyectos y la comercialización de producciones científicas, lo que permite el autofinanciamiento y desarrollo de los grupos científicos. Se optimizan las estructuras y métodos de dirección. Se incrementa el uso de la informatización. Las redes aseguran el empleo de las TIC.

En la actualidad la Universidad de Sancti-Spíritus para impartir la Programación Lineal sigue los métodos tradicionales, por medio de un profesor empleando conferencias, clases prácticas, seminarios, y apoyados de cierta forma en la bibliografía existente como es el caso de los libros de textos y el uso de la Internet. Esta materia es impartida a estudiantes que cursan en tercer año de contabilidad y finanzas como un tema de la asignatura de investigación de operaciones. Los estudiantes para poder profundizar en el tema deben hacer búsquedas a ciegas en Internet, así como visitar la biblioteca buscando algún documento que trate el tema. Para muchos estudiantes de la carrera de contabilidad y finanzas se torna difícil hacer búsquedas en Internet y en alguno de los casos estas búsquedas no muestran los resultados deseados. Se cuenta además con un software que permite trabajar con problemas de esta índole, el mismo solo comprende lo relacionado con la resolución de ejercicios, no explica teoría alguna y de sentarse un



estudiante a trabajar sin conocer los conocimientos teóricos necesarios le sería imposible dar solución a un ejercicio.

### 1.10 Causas que originan la situación problemática y las consecuencias

La programación lineal es un arma muy poderosa para mejorar la economía de cualquier país, ya que por mediación de la misma podemos minimizar gastos, maximizar ganancias entre otras tantas aplicaciones. Cuba por ser un país bloqueado económicamente y subdesarrollado le sería de mucha utilidad el empleo de este tipo de técnicas, para mejorar su economía haciendo una mejor planificación de sus recursos. Siendo los estudiantes de contabilidad y finanzas los futuros profesionales que trabajarán la parte económica de cualquier entidad del país, se convierte en un hecho la responsabilidad que tiene la Universidad de Sancti Spíritus en formar profesionales capaces de ayudar a mejorar la economía y tomar las decisiones correctas antes las posibles dificultades. Conociendo todo lo que se puede lograr por mediación de la programación lineal es de vital importancia que los estudiantes adquieran la mayor cantidad de conocimientos con respecto a esta temática. De un estudio realizado en la biblioteca perteneciente a la Universidad de Sancti Spíritus se pudo comprobar que de los estudiantes graduados de contabilidad y finanzas en Sancti Spíritus desde el año 1998 hasta el año 2009 solo 7 estudiantes de 392 hicieron su trabajo de diploma para optar por el título de licenciado en contabilidad y finanzas con temas en los que se hacía empleo de la programación lineal. Representando el 1.8 % de los graduados en todos estos años.

Haciendo un análisis por año vemos que:

Curso	Número de Graduados con temas en los que se emplea la programación lineal	Trabajos Vinculados
1998-1999	6	0
1999-2000	12	0
2000-2001	8	0
2001-2002	22	1
2002-2003	42	0
2003-2004	13	2
2004-2005	43	4
2005-2006	54	0
2006-2007	53	0
2007-2008	81	0
2008-2009	58	0

Todo esto muestra el poco interés y conocimiento que existe actualmente con respecto a la materia abordada. En la actualidad existen pocos software que faciliten a los estudiantes consultar esta temática, que se ajusten a los contenidos que necesitan estudiar, y de una manera amena, que causen interés. Es por ello que se quiere lograr la implementación de un software tutorial en el cual aparezca todo lo vinculado con la parte teórica-explicativa de la temática, así como una parte evaluativa para conocer donde radican las principales dificultades.

Con la implementación del tutorial se quiere automatizar distintas partes: explicativa, evaluativa, investigativa, de preparación, administrativa y de autenticación.

➤ Parte explicativa

En esta aparecen las teorías que comprenden la construcción del sistema de restricciones. Aquí el estudiantado encontrará todas aquellas teorías existentes en el software de diferentes investigadores. Concluyendo esta parte los estudiantes deben ser capaces a partir de un problema de programación lineal de construir el sistema de restricciones.

➤ Parte de preparación

Los estudiantes contarán con ejercicios resueltos y ejercicios propuestos para poder tener una mejor preparación, así como para aplicar todo lo estudiado y aprendido en la parte explicativa.

➤ Parte investigativa

Se mostrarán sitios Web y documentos digitales vinculados con el tema para adquirir una mejor preparación.

➤ Parte evaluativa

Los estudiantes podrán medir los conocimientos adquiridos a través de autoexámenes que otorgan una calificación. De no ser la máxima, estos pueden consultar los errores cometidos.

➤ Parte administrativa

En esta parte juega un importante papel el profesor o administrador pues es el encargado de mantener actualizada la información del sistema y de eliminar la información que no sea necesaria. El profesor cuenta además con la posibilidad de monitorear la actividad de los estudiantes en el sistema.

➤ Parte de autenticación

El software brinda la posibilidad de que toda aquella persona que no se encuentre registrada en el sistema pueda crear su propia cuenta de usuario y tener acceso a todos los contenidos y herramientas existentes en el mismo. Estas personas no contarán con los privilegios de un administrador por lo que no podrán modificar el contenido del tutorial.

### **1.11 Análisis comparativo de otras soluciones existentes con la propuesta**

Existen disímiles soluciones para enseñar la programación lineal como son los casos de los libros de textos en los cuales aparecen por lo general las principales teorías para la construcción de un modelo de programación lineal, así como ejercicios resueltos y hasta explicaciones de los métodos de resolución como es el caso del método simplex. En Internet se realizó una búsqueda minuciosa y se pudo llegar a la conclusión de que no se encontró un software tutorial de programación lineal, solo aparecen textos que explican las teorías llamados también por el nombre de tutorial en formato pdf, pero sin lograr ningún intercambio con los que acuden al mismo en búsqueda de conocimientos.

El presente tutorial de programación lineal es un software que explica, motiva, evalúa, entre otras funcionalidades, por lo que se puede considerar superior a cualquier texto o tutorial que pueda ser encontrado en Internet, no obstante estos tutoriales que existen en Internet se emplearán como materiales de apoyo pues el software tiene entre sus

funcionalidades almacenado en la base de datos del mismo todas aquellos sitios, tutoriales o libros que traten el tema.

Este software es por definición propia un Software Educativo, que será utilizado únicamente con el fin de proporcionarle al estudiante y a los profesionales del tema una herramienta a través de la cual poder expandir sus conocimientos, y porque no, una vía de autoevaluar sus propios conocimientos, siempre con información actualizada y en constante transformación de ahí que durante su creación nos planteamos los siguientes objetivos:

- 1.- Desarrollar una herramienta para la enseñanza y comprobación de los conocimientos de la Programación Lineal.
- 2.- Garantizar la continuidad del estudio de la asignatura para los cursos continuantes de Ingeniería y cualquier otra persona interesada en el tema.
- 3.- Proporcionar a los compañeros interesados de un modelo adicional de interactuar con las temáticas que aquí les presentamos, y de esta forma fortalecer sus conocimientos considerablemente, a través de las palabras técnicas y su significado.

## **1.12 Fundamentación de la metodología utilizada**

### **1.12.1 Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado de Desarrollo, fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos.

El Rational Unified Process unifica todo el equipo de desarrollo de software y mejora la comunicación del equipo al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar el software.(GUERRERO, LUIS A.)

RUP pretende implementar las mejores prácticas actuales en ingeniería de software: desarrollo iterativo del software, administración de requerimientos, uso de arquitecturas basadas en componentes, modelamiento visual del software, verificación de la calidad del software y control de cambios.

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, los cuales se muestran en la Figura 2.1, que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

A continuación se muestran estas prácticas:

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente
- Verifica la calidad del software.
- Controla los cambios.

Para apoyar el trabajo con esta metodología ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose en el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto. Después del análisis realizado a ambas metodologías se decidió, por parte de los autores, utilizar para la elaboración del presente documento y para llevar a cabo paso a paso todo el proceso de desarrollo del software propuesto la metodología RUP. Esto responde fundamentalmente a que esta metodología se ha convertido en un estándar internacional para guiar el proceso de desarrollo de software, al igual que en nuestro país y además porque se cuenta también con la herramienta CASE Rational Rose del 2003, con la que se han elaborado todos los diagramas incluidos en este documento.

### 1.12.2 Lenguaje de Modelamiento Unificado (UML)

El Lenguaje de Modelado Unificado o UML es una notación estándar con carácter universal utilizado para escribir modelos de sistemas, ante todo, de sistemas de software, que utiliza una serie de diagramas y una semántica bien definida con el propósito de elaborar los artefactos de un sistema a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo. (Salazar & Antonio)

Los objetivos que se fijaron al desarrollar el UML fueron los siguientes:

- Proporcionar a los usuarios un Lenguaje de Modelado Visual de tal forma que sea posible intercambiar información de los modelos.
- Proporcionar mecanismos de extensibilidad y especialización para ampliar los conceptos básicos.
- Ser independiente de un lenguaje en particular y del proceso de desarrollo.
- Proporcionar bases formales para la comprensión del Lenguaje de Modelado.
- Integración en una mejor práctica.
- El UML es un lenguaje de modelado que incorpora a la comunidad orientada a objetos el consenso de los conceptos de modelado básico y permite desviaciones, las cuales se expresan en términos de mecanismos de extensión. Es un conjunto preciso que consiste en la definición de la semántica y notación del UML, definiendo también como se maneja el Lenguaje de Especificación de Objetos
- El UML es un Lenguaje de Modelado Visual riguroso, y ya convertido en un estándar, es la herramienta ideal para atacar el ciclo de vida de un proyecto de software utilizando la tecnología Orientada a Objetos. . (Romero)

Los tres elementos que forman el modelo conceptual de UML son: los bloques básicos de construcción del lenguaje, las reglas que se aplican sobre esos bloques y los mecanismos comunes de UML. (DR. JIE ZHAO).

Bloques de construcción

Existen tres tipos de bloques de construcción:

**Elementos:** Son los modelos UML.

**Relaciones:** Ligan elementos entre sí, establecen la forma en que interactúan.

**Diagramas:** Representación gráfica de un grupo de elementos y sus relaciones.

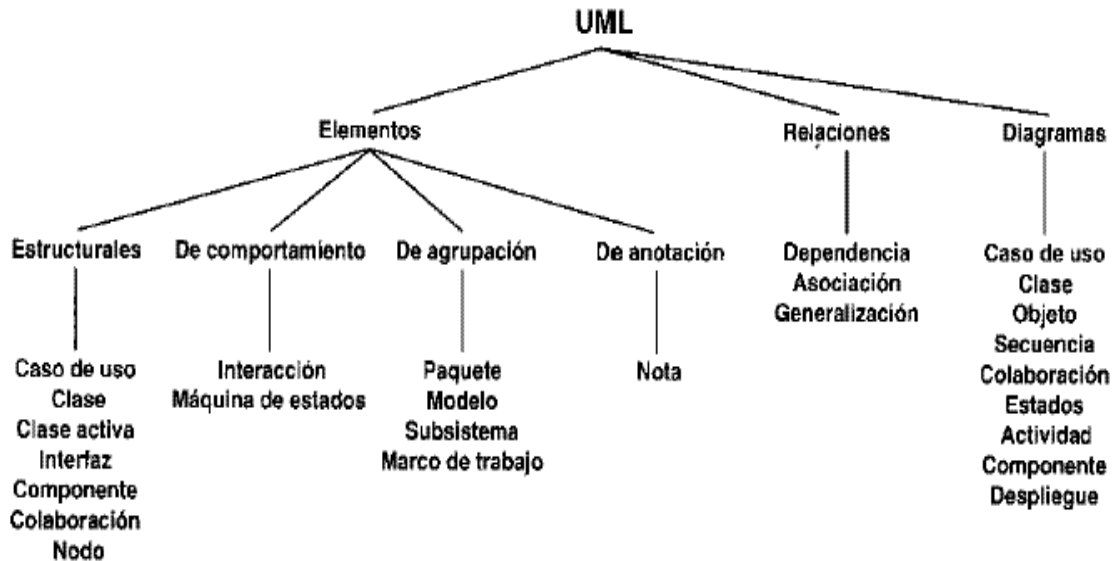


Figura 1. UML. Elementos fundamentales

## Reglas

Los bloques de construcción se deben combinar siguiendo las normas que establece UML. UML establece una serie de normas sobre cómo nombrar a los elementos, relaciones y diagramas; la visibilidad y alcance de dichos nombres y sobre su integridad (cómo diseñar relaciones consistentes). Podremos decir que un modelo está bien formado cuando cumpla estas reglas.

## Mecanismos comunes.

Bajo una serie de mecanismos que se aplican durante todo el proceso de desarrollo de modelos en UML, se consiguen diseños simples y eficientes. Estos mecanismos son los siguientes:

**Especificaciones:** UML no es simplemente un lenguaje que proporciona elementos gráficos para modelado, tras cada elemento existe una especificación que nos va a permitir detallar textualmente el comportamiento de los mismos; de esta forma con los elementos gráficos conseguimos una visualización global del sistema y con la especificación conseguimos una visualización detallada del mismo.

**Adornos:** Todos los elementos gráficos en UML poseen una notación gráfica que

representa los aspectos más importantes de éstos; aún así existe una notación que permite representar los aspectos más específicos a través de adornos, consiguiendo así una especificación aún más detallada.

Divisiones comunes: UML permite representar las distintas divisiones que presenta el diseño orientado a objetos. Por ejemplo la división clase/objeto donde un objeto es la manifestación concreta de la clase. UML permite el modelado conjunto de los mismos a través de una notación específica.

Mecanismos de extensibilidad: UML proporciona un lenguaje abierto en el que se pueden expresar los matices de los distintos sistemas, para ello hace uso de tres mecanismos que permiten extender el lenguaje:

Estereotipo: Permite crear nuevos bloques de construcción partiendo de los ya existentes.

Valor etiquetado: Permite añadir nuevas propiedades a un bloque de construcción.

Restricción: Permite añadir o modificar las reglas referidas a los bloques de construcción. (LAGUNA 2004)

### **1.12.3 Arquitectura de desarrollo de N Capas**

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se modifica el nivel requerido sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la Interfaz de Programación de Aplicaciones (Application Programming Interface - API por sus siglas en Inglés) que existe entre los niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

El diseño más usado en la actualidad es el de tres capas



1. Capa de presentación
2. Capa de lógica de negocio
3. Capa de datos

**1.- Capa de presentación:** Esta capa provee la interfaz de usuario (IU). Aquí es donde la aplicación presenta información a los usuarios y acepta entradas o visualiza respuestas. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios. Por el contrario, la IU debería relegar sobre la capa de negocios para manipular estos asuntos. Esto es importante, especialmente hoy en día, debido a que es muy común para una aplicación tener múltiples IU.

**2.- Capa de negocio:** es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio - LN) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento o la recuperación de los mismos.

**3.- Capa de datos:** La capa de acceso a datos (AD) está formada por uno o más Sistemas Gestores de Bases de Datos (SGBD) que se encargan del almacenamiento y recuperación de los datos que necesita la capa de negocio.

Todas estas capas pueden residir en un mismo equipo. Lo más usual es que haya una multitud de computadoras donde resida la capa de presentación, es decir los clientes de la arquitectura cliente/servidor, y una computadora central que ejerce la función de servidor en dicha arquitectura. Las capas de negocio y de datos pueden residir en un mismo servidor y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más servidores. Así mismo, si el tamaño o la complejidad de la base de datos aumentaran, se puede separar en varios servidores de datos, los cuales recibirán las peticiones del servidor donde reside la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en una o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de

ordenadores sobre los cuales corre la base de datos. En este caso se está haciendo referencia a Sistemas de Aplicaciones Distribuidas.

En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

### **1.13 Tecnologías y Herramientas**

Para poder garantizar calidad en el desarrollo de una herramienta de software es necesario seguir las indicaciones de alguna metodología. Es importante antes de llevar a cabo el proceso de desarrollo, hacer un estudio de cuáles son las tecnologías actuales, conocidas o no, con el fin de seleccionar y utilizar la más conveniente. Se describen las características fundamentales de algunas de las tecnologías existentes para el desarrollo de software justificando la seleccionada para la implementación de dicha propuesta.

#### **1.13.1 Borland Delphi 7.0**

El Object Pascal es el lenguaje que Delphi utiliza para crear las aplicaciones orientadas a objetos. Debido a que Delphi pertenece a la empresa Borland, la potencia de éste puede compararse con el compilador de C++. Borland Delphi es un ambiente de desarrollo rápido de aplicaciones (RAD) muy flexible y fácil de usar. Estos últimos años ha tenido una gran repercusión dentro del mundo de la programación visual. Presenta un ambiente visual de desarrollo para aplicaciones controlados por eventos de usuario sobre interfaces gráficas. Proporciona una jerarquía muy extensa de clases de objetos reutilizables. es una herramienta de propósito general, se puede programar tanto a bajo nivel, como a alto nivel (simplemente usando controles y ajustando propiedades) y tiene buenas capacidades gráficas. Las aplicaciones creadas en Delphi solo funcionan sobre la plataforma de trabajo Windows.

#### **1.13.2 Sistemas Gestores de Base de Datos (SGDB)**

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

Un Sistema de Gestión de Base de Datos (SGBD) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

Un SGBD tiene los siguientes objetivos específicos:

- Independencia de los datos y los programas de aplicación
- Minimización de la redundancia
- Integración y sincronización de las bases de datos
- Integridad de los datos
- Seguridad y protección de los datos
- Facilidad de manipulación de la información
- Control centralizado

En los SGBD basados en el modelo relacional, la información es representada a través de tuplas, las cuales describen el fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas. (Matos)

### **1.13.3 SQL (Structured Query Language)**

Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella.

### **1.13.4 SQL (Structured Query Language) Server 2000**

Microsoft SQL Server, propietario de Microsoft, pertenece a la familia de los sistemas de administración de base de datos, operando en una arquitectura cliente/servidor de gran rendimiento. Su desarrollo fue orientado para hacer posible manejar grandes volúmenes de información, y un elevado número de transacciones. SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor sólo tiene que

enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos. (SQL)

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones SQL que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL, se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes. Microsoft SQL Server constituye un fuerte gestor de base de datos, que puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos, sólo depende de la potencia del hardware del equipo en el que esté instalado y solamente corre sobre Windows NT- 2000 Server o superior. Además que permite la ejecución de procedimientos almacenado o Stored Procedures. (Manual de SQL)

#### **1.13.5 Hiper Text Markup Lenguaje (HTML)**

Lenguaje de marcas de hipertexto que se utiliza desde 1989. Los documentos HTML contienen dos tipos de información: la que se muestra en la pantalla (texto, imágenes...) y los códigos (tags o etiquetas), transparentes al usuario, que indican cómo se debe mostrar esa información. A medida que se ha ido avanzando, se han estandarizado distintas versiones del lenguaje HTML. Cada una de ellas amplía el número de etiquetas, lo que permite nuevas posibilidades para los documentos; así, se le ha dotado de marcas para rellenar formularios (forms) de manera interactiva, que permiten al usuario enviar la información necesaria para realizar consultas en bases de datos, comprar o solicitar un servicio.

La versión HTML 4.0 se utiliza en combinación con el XML 1.0, otro subconjunto de SGML que permite al desarrollador definir sus propias etiquetas; el resultado es un nuevo formato denominado XHTML, que se espera que constituya un nuevo estándar de formato para páginas Web. La evolución de las distintas versiones del lenguaje HTML ha estado determinada por un desarrollo paralelo de los navegadores, que permitiese interpretar las nuevas etiquetas. El avance de unos y otros ha contribuido al crecimiento exponencial que ha experimentado WWW.

(Microsoft Corporation, 2005)

## 1.14 Conclusiones

En el presente capítulo se han definido las condiciones que rodean el objeto de estudio de la investigación. Así como las tecnologías usadas para su confección.

Se han justificado las razones por las cuales ha sido seleccionada como metodología a seguir en la documentación del software propuesto para la implementación el lenguaje Object Pascal utilizando como herramienta de programación el **Borland Delphi v 7.0** y el **SQL Server 2000**.

## **Capítulo 2: Modelo del Dominio**

### **2.1 Introducción**

Teniendo en cuenta la descripción de los procesos que se trataron anteriormente, puede afirmarse que la situación tratada en este trabajo tiene muy bajo nivel de estructuración para ser modelada como un negocio, por lo que en el presente capítulo tomando como guía la Metodología RUP, se utiliza uno de los artefactos que brinda dicha metodología: el modelo de dominio, los requerimientos funcionales y no funcionales, el diagrama de casos de uso y la descripción de cada uno, los cuales ayudan a modelar y describir la solución propuesta. Además, se presenta una descripción detallada de las reglas de negocio con el objetivo de asegurar el cumplimiento de las restricciones que existen en el dominio.

### **2.2 Descripción del modelo de dominio**

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos (Jacobson I. , 2000). La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema. (Bravo, 2004).

En el modelo del dominio empleado en este trabajo se definen las siguientes entidades: Administrador y Usuario; y clases principales: Teoría y Autoexamen.

La esencia del funcionamiento de este dominio consiste en el constante acceso de los usuarios al tutorial con el objetivo de adquirir conocimientos teóricos y prácticos acerca del Sistema de Restricciones de la Programación Lineal, haciendo uso para esto de distintas fuentes teóricas que han descrito este tema, de ejercicios resueltos y ejemplos que ayuden a desarrollar las habilidades de los usuarios. Con la posibilidad además de evaluar los conocimientos adquiridos mediante un autoexamen.

### 2.2.1 Principales conceptos asociados al dominio del problema.

**Usuario:** Se refiere a los usuarios registrados en el Tutorial y que interactúan con este. Los mismos son los estudiantes de Contabilidad y Finanzas de la UNISS.

**Administrador:** Se refiere a la persona con total acceso al tutorial, que puede realizar modificaciones en este con el objetivo de actualizarlo. Además puede controlar la actividad de los usuarios en el tutorial.

**Autoexamen:** Se refiere a la herramienta que genera el tutorial con el objetivo de evaluar los conocimientos adquiridos por los usuarios.

**Teoría:** Es la clase principal que se refiere a las teorías, ejercicios resueltos y ejercicios propuestos los cuales son gestionados por el administrador y utilizados por los usuarios.

### 2.2.2 Modelo de objetos del dominio

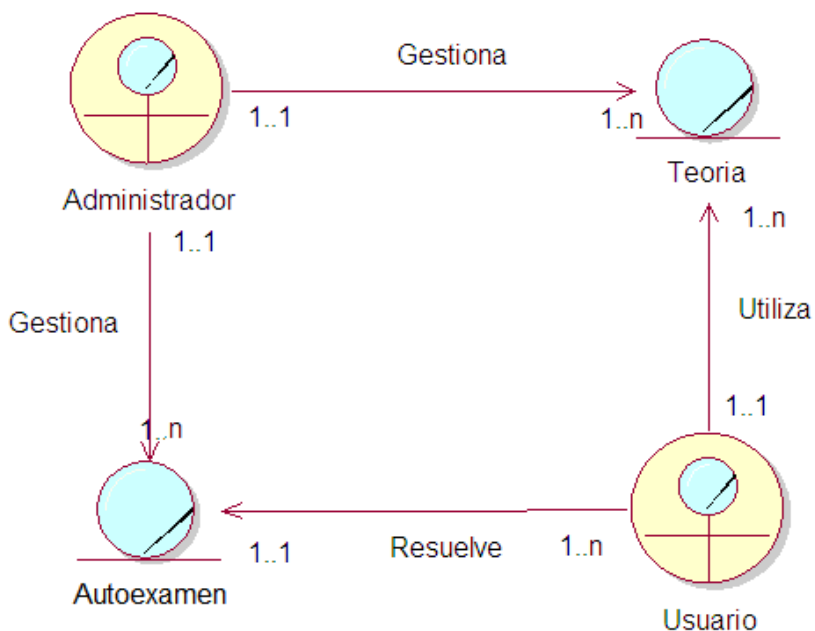


Figura 2.1. Modelo de Dominio.

## 2.3 Reglas a considerar

Las reglas de negocio regulan y describen las principales políticas que deben cumplirse para el adecuado funcionamiento del negocio. A continuación se presentan las que fueron identificadas.

- Para tener acceso a la información y herramientas que ofrece el sistema es necesario que la persona se encuentre registrada como usuario del mismo.
- Para la inserción de datos en el sistema por el administrador es necesario cumplir la estructura para el tipo de información que se desea insertar, ya sean problemas propuestos, problemas resueltos, teorías o preguntas de autoexamen.
- La información necesaria para el funcionamiento del sistema debe ser lo más real posible y adaptarse a las necesidades del usuario..
- La única persona que tiene autorización para eliminar contenido o cuentas de usuario del sistema así como de crear un usuario con privilegios administrativos es el administrador del sistema.
- Las cuentas de usuario son personales e intransferibles, de manera que el administrador pueda tener acceso a la información de cada usuario para poder evaluar su trayectoria por el sistema.

## 2.4 Requisitos funcionales

Para la realización de cualquier sistema es necesario establecer una serie de objetivos y metas que tienen que estar presentes para que el cliente esté satisfecho; es a esto a lo que le llamamos requisitos funcionales del sistema.

En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. (Soto, 2006)

1. Gestionar usuario administrador
  - 1.1 Crear nuevo usuario
  - 1.2 Modificar usuario
  - 1.3 Eliminar usuario
  - 1.4 Cambiar contraseña
2. Gestionar usuario
  - 2.1 Crear Cuenta usuario



- 2.2 Cambiar contraseña
- 3. Ver usuarios conectados
- 4. Gestionar teoría
  - 4.1 Insertar teoría
  - 4.2 Modificar teoría
  - 4.3 Eliminar teoría
- 5. Ver teoría
- 6. Gestionar problemas propuestos
  - 6.1 Insertar problema propuesto
  - 6.2 Modificar problema propuesto
  - 6.3 Eliminar problema propuesto
- 7. Ver problemas propuesto
- 8. Gestionar problema resuelto
  - 8.1 Insertar problema resuelto
  - 8.2 Modificar problema resuelto
  - 8.3 Eliminar problema resuelto
- 9. Ver problemas resueltos
- 10. Eliminar traza
- 11. Ver traza
- 12. Gestionar Web
  - 12.1 Insertar Web
  - 12.2 Modificar Web
  - 12.3 Eliminar Web
- 13. Ver Web
- 14. Gestionar ejercicios de selección
  - 14.1 Insertar ejercicios de selección
  - 14.2 Modificar ejercicios de selección
  - 14.3 Eliminar ejercicios de selección
- 15. Gestionar ejercicios de enlazar
  - 15.1 Insertar ejercicios de enlazar
  - 15.2 Modificar ejercicios de enlazar
  - 15.3 Eliminar ejercicios de enlazar
- 16. Gestionar ejercicios de verdaderos y falsos
  - 16.1 Insertar ejercicios de verdadero o falso

- 16.2 Modificar ejercicios de verdadero o falso
- 16.3 Eliminar ejercicios de verdadero o falso
- 17. Resolver autoexamen
- 18. Reporte de los Usuarios del Tutorial.
- 19. Reporte de los Usuarios que más se evalúan.
- 20. Reporte de los Usuarios evaluados en un período.
- 21. Reporte de Usuarios por Nota.
- 22. Reporte de la Teoría.

## **2.5 Requisitos no funcionales**

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema. (Soto, 2006)

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener. Son las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los son fundamentales en el éxito del producto. Están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer podemos, determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que los clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la funcionalidad requerida, las propiedades no funcionales, pueden marcar la diferencia entre un producto y otro. Los requerimientos no funcionales incluyen:

- ❖ Conjunto de facilidades.
- ❖ Capacidades.

❖ Seguridad.

**Apariencia o interfaz externa:** teniendo en cuenta las características de los usuarios hacia los cuales va dirigido, el sistema debe ser fácil de usar. Debe ser además autoritario para que los usuarios se sientan confiados al utilizarlo.

**Usabilidad:** se estima que será elevada, es decir que cuente con un elevado nivel de aceptación para los usuarios, debido a que constituye una forma más cómoda y rápida de realizar diferentes operaciones que en la actualidad resultan un tanto engorrosas. Además con él hay un aumento en la productividad del proceso al tornarse automatizado y por ende más rápido, exacto y eficiente. Se confeccionó de forma tal que puede ser utilizado por usuarios con conocimientos mínimos de informática, por lo que esto no constituye una limitación para la utilización del mismo.

**Rendimiento:** en cuanto al rendimiento la mayoría de las funcionalidades que están previstas necesitan de una respuesta instantánea, por lo que se tendrá que garantizar la no ocurrencia de demoras con la selección de la tecnología realizada.

La información deberá estar disponible las 24 horas del día, aunque una caída momentánea del sistema no provocará un error en las labores; sin embargo, el sistema deberá recuperarse en un corto período de tiempo ante una falla.

**Soporte:** se requiere que la base de datos sea configurable teniendo en cuenta el futuro crecimiento del sistema.

El sistema deberá ser compatible con el sistema operativo Windows. Esto es conveniente dado su internacionalización y amplia difusión dentro del mundo de aplicaciones profesionales para centros educativos.

**Portabilidad:** se garantiza la implementación del sistema en diferentes plataformas una vez que el cliente haya decidido actualizar la existente, en un tiempo acorde a los requerimientos del momento.

**Seguridad:** garantizar un control riguroso sobre la seguridad de la información teniendo en cuenta los niveles de acceso. No se permitirán accesos sin autorización al sistema,

para ello se definirá una política de usuarios con roles y privilegios diferentes para que la información pueda ser consultada por las personas autorizadas. Para ello el sistema contará con dos tipos de usuarios, el administrador y los usuarios, que según el rol que tengan asignado en el sistema será el acceso a la parte de la información que le corresponde. El administrador del sistema es el que asigna los roles de funcionalidad de cada módulo y controla todo lo referente a la seguridad del sistema.

La integridad de los datos que se almacenen en el servidor deberá ser consistente. Se validarán las entradas de datos irreales y mecanismos de vuelta atrás en procesos críticos que terminen abruptamente y produzcan estados inconsistentes de la información.

**Requerimientos de confiabilidad:** el sistema debe ser confiable.

**Ayuda y Documentación en Línea:** Cualquier usuario por muy poca habilidad informática que tenga, si domina los conocimientos mínimos de informática, puede hacer uso del sistema; además se implementó lo suficientemente claro y viable.

**Requerimientos de Software:**

- ❖ Servidor: SQL Server y cualquier plataforma de Windows superior a Windows 95.

**Requerimientos de Hardware:**

- ❖ Cliente: como mínimo un Pentium, con 32 MB de RAM y 100 MHz de velocidad.
- ❖ Servidor: como mínimo un Pentium II, con 128 MB de RAM y 300 MHz de velocidad.

**Restricciones en el diseño y la implementación:** Se desea desarrollar una herramienta dinámica para la automatización de los procesos de enseñanza y autoaprendizaje de la programación lineal en un ambiente de aplicación de desktop. Hemos escogido el lenguaje de programación Borland Delphi 2006, por las posibilidades multiplataforma que este brinda y lo fácil que resulta de aprender. Además su uso ha ido en ascenso en los últimos tiempos.

## 2.6 Descripción del sistema propuesto

Con la implementación de este sistema se pretende facilitar una herramienta a los estudiantes de tercer año de contabilidad y finanzas de la Universidad de Sancti Spíritus en forma de tutorial para el mejoramiento del aprendizaje de la programación lineal, automatizando los procesos principales que rigen la enseñanza de la programación lineal. A continuación se describirá en detalle las funcionalidades del sistema.

Para formar parte del sistema los usuarios deben crearse su propia cuenta de usuario. Esta información se refiere inicialmente, a que cuando un usuario solicita ver el sistema, se muestra todo lo referente a libros, teorías, ejercicios, autoexámenes, entre otras funcionalidades para fomentar sus conocimientos de programación lineal.

Cuando el estudiante se crea su propia cuenta ya pasa a ser miembro del sistema por lo que el administrador (profesor) podrá tener constancia de la participación del estudiante y su nivel alcanzado por mediación de los autoexámenes. El estudiante una vez usuario del sistema cuenta con la posibilidad de ver todas aquellas teorías referentes al tema Sistema de Restricciones de la Programación Lineal, estas teorías mostrarán además su autor.

También se cuenta con la dirección de sitios vinculados al tema donde pueden los estudiantes buscar información que le pueda servir de utilidad.

El sistema contará con un conjunto de ejercicios que se muestra al usuario separados en dos grupos:

- Ejercicios resueltos
- Ejercicios propuestos

Los ejercicios resueltos serán mostrados al usuario con el procedimiento que se fue llevando a cabo para su solución y los propuestos son para que el usuario sea quien le dé solución, aunque estos últimos, de querer el usuario, puede ver su respuesta ya sea para comprobarla con el resultado obtenido por el mismo o para conocerla si el usuario no fue capaz de resolverlo.

El tutorial cuenta con un módulo en el cual los usuarios pueden evaluarse mediante autoexámenes donde el estudiante puede realizar tres tipos de preguntas, verdaderos y

falsos para evaluar los principales conceptos que deben ser dominados por los estudiantes, enlazar y selección simple, estas dos últimas evaluarían los conocimientos prácticos. Una vez realizado el autoexamen los estudiantes conocerán la nota obtenida en cada ejercicio y su nota final, así como las preguntas contestadas incorrectamente con su respuesta. Los autoexámenes serán preguntas generadas aleatoriamente, una de cada tipo de pregunta.

Por parte de la administración del sistema:

El administrador es el único facultado para tener un control absoluto del sistema, es el encargado de entrar, modificar o eliminar el contenido del tutorial, como es el caso de los problemas resueltos, problemas propuestos, preguntas del autoexamen, libros, sitios webs y teorías. El mismo contará con la posibilidad de ver todas las actividades desarrolladas por los estudiantes en la aplicación así como los resultados obtenidos en los autoexámenes. El administrador es el único facultado además para crear otras cuentas con privilegios de administración, y de eliminar un usuario del sistema.

## **2.7 Modelo de Casos de Uso del Sistema**

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario (Jacobson I. , 2004)].

### **2.7.1 Actores del sistema**

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interactúan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema (Jacobson I. , 2004).

<b>Actores</b>	<b>Justificación</b>
Administrador	Es la persona con acceso total al sistema, el cuál posee mayores privilegios que el usuario. Este puede introducir

	<p>datos, modificar el contenido existente en el sistema, y además supervisar las cuentas de usuarios y su desempeño. El administrador puede crear cuentas con privilegios de administración.</p> <p><b>Requerimientos asociados:</b> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18.</p>
Usuario	<p>Es la persona que utiliza el sistema para adquirir conocimientos. Este tiene acceso limitado al sistema. Puede revisar toda la información teórica, realizar autoexámenes, cambiar su contraseña y revisar la ayuda y reportes del sistema en cualquier momento.</p> <p><b>Requerimientos asociados:</b> 1,4,6,8,12,16,17,18.</p>

**Tabla 2.1** Actores del sistema.

## 2.7.2 Casos de Uso del sistema

Cada forma en que los actores usan el sistema se representa con un caso de uso.

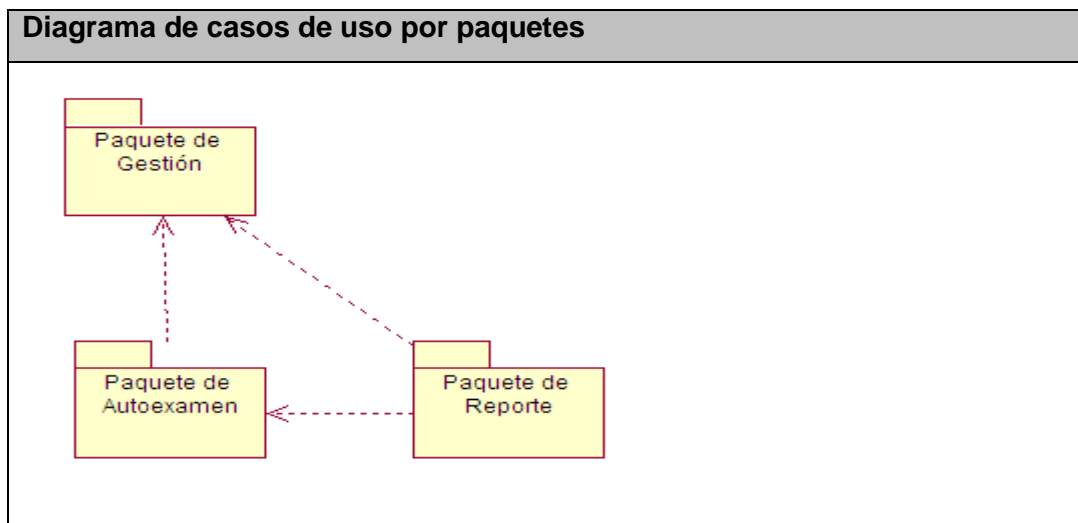
Los casos de uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Jacobson I. , 2004).

Se definieron los siguientes casos de uso para el sistema propuesto:

1. Gestionar usuario administrador
2. Gestionar usuario
3. Ver usuarios conectados
4. Gestionar teoría
5. Ver teoría
6. Gestionar problemas propuestos
7. Ver problemas propuestos
8. Gestionar problemas resueltos
9. Ver problemas resueltos
10. Eliminar traza

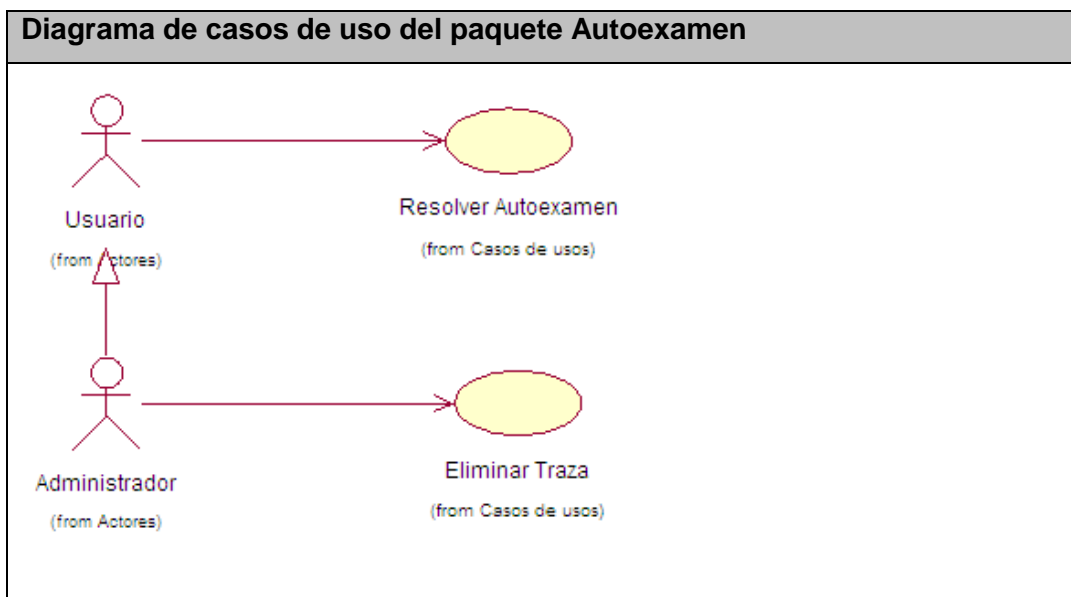
11. Ver traza
12. Gestionar Web
13. Ver Web
14. Gestionar ejercicios de selección
15. Gestionar ejercicios de enlazar
16. Gestionar ejercicios de verdaderos y falsos
17. Resolver autoexamen
18. Reporte Teoría
19. Reporte Usuarios del Sistema.
20. Reporte Usuarios Evaluados por Período
21. Reporte Usuarios por Nota
22. Reporte Usuarios que más se Evalúan

Se decide realizar el diagrama de casos de uso definiendo paquetes para subdividir éste, debido a que el diagrama es extenso y se hace difícil su comprensión. Se muestra un diagrama por cada paquete. Los paquetes de casos de uso son la forma de agrupar a estos últimos respondiendo a algún criterio. Se realizaron 3 paquetes: Autoexamen, Gestión y Reportes. El paquete de Reportes y el de Autoexamen dependen del paquete de Gestión y a su vez el de Reportes depende del paquete de Autoexamen.



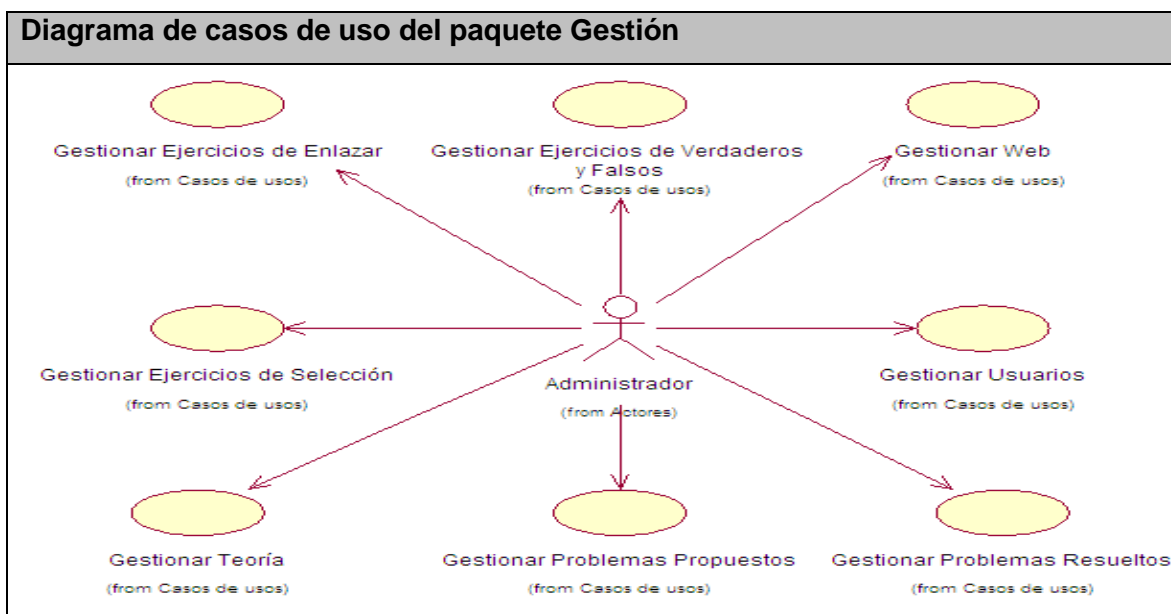
**Figura. 2.2** Diagrama de casos de uso por paquetes





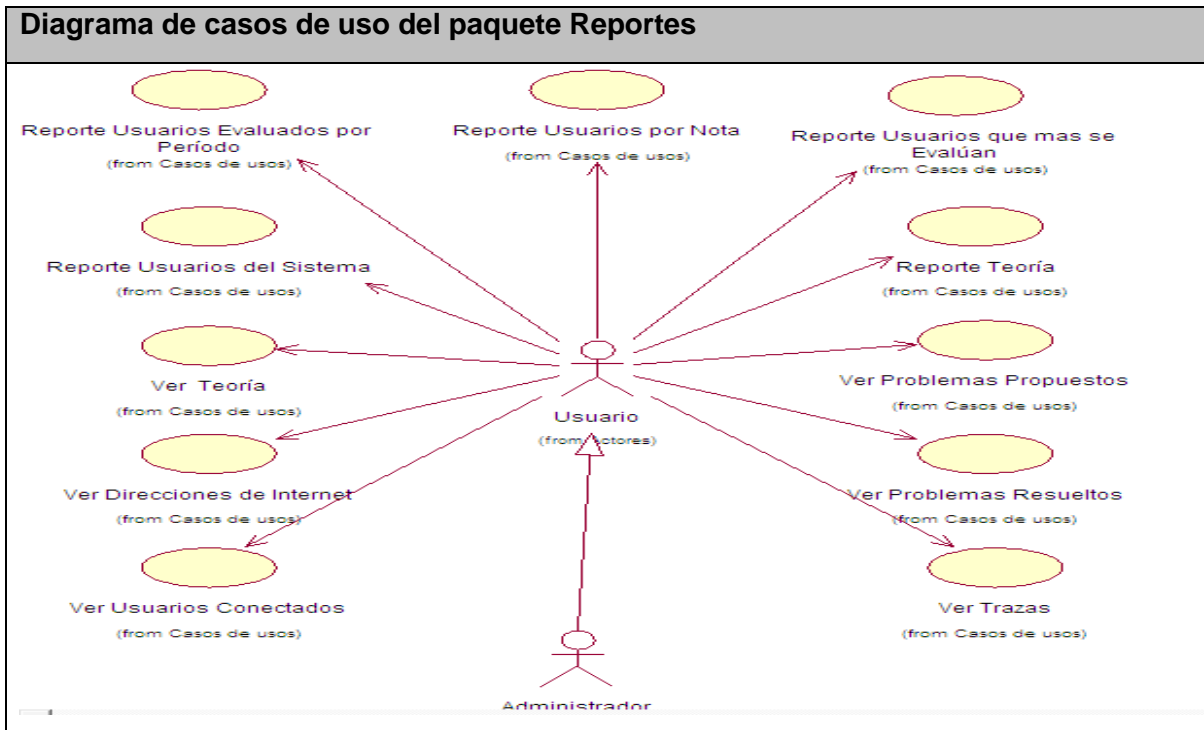
**Figura. 2.3** Diagrama de casos de uso del paquete Autoexamen

En este paquete tanto el Usuario como el Administrador pueden ejecutar un autoexamen para comprobar los conocimientos adquiridos en el Tutorial además permite al administrador eliminar las trazas de usuario que ya consultadas y analizadas no sean necesarias. De esta manera se mantiene el sistema con información actualizada.



**Figura. 2.4** Diagrama de casos de uso del paquete Gestión.

Este paquete recoge los casos de uso de gestión del Tutorial los cuales son ejecutados por el administrador con el objetivo de insertar, modificar o eliminar datos en el Tutorial. De esta manera se mantiene el sistema con información actualizada.



**Figura. 2.5** Diagrama de casos de uso del paquete Reportes

Este paquete recoge los casos de uso que ofrecen información acerca del Tutorial a los actores.

### 2.7.3 Descripción de los casos de usos del sistema

Caso de Uso	Gestionar usuario administrador.
Actores	Administrador. (inicia)
Propósito	Adicionar, modificar o eliminar un usuario al sistema. Cambiar contraseña
Resumen	

Se inicia por cuatro vías distintas:

1. Cuando el administrador desea insertar un usuario al sistema. Se muestra el formulario donde este inserta el usuario y sus datos.
2. Cuando el administrador desea modificar un usuario existente. Se muestran los usuarios registrados en el sistema, selecciona el que desea modificar y realiza los cambios.
3. Cuando el administrador desea eliminar un usuario del sistema. Se muestran los usuarios registrados en el sistema, selecciona el que desea eliminar y realiza los cambios.
4. Cuando el administrador desea cambiar su contraseña. Se le muestra el formulario donde este inserta los datos.

El caso de uso culmina con la actualización de los datos en el sistema.

Referencias	R 1.1, R 1.2, R 1.3, R 1.4
Precondiciones	-El usuario que se desea eliminar o modificar debe encontrarse en la base de datos del sistema.
Postcondiciones	-La información es almacenada en la base de datos del sistema. -Se muestra la forma principal.
Prototipo	
Ver anexo II.19, II.20, II.21	

**Tabla 2.2** Descripción del caso de uso de sistema Gestionar usuario administrador.

Caso de Uso	Gestionar usuario.
Actores	Usuario (inicia).
Propósito	Crear una nueva cuenta de usuario. Cambiar contraseña.
Resumen	
Se inicia por dos vías distintas:	
<ol style="list-style-type: none"> <li>1. Cuando el actor desea registrarse en el sistema. Se muestra el</li> </ol>	

<p>formulario donde este inserta sus datos personales y los datos necesarios para su cuenta.</p> <p><b>2.</b> Cuando un usuario del sistema desea cambiar su contraseña. Se le muestra el formulario donde este inserta los datos.</p> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 2.1, R 2.2
Precondiciones	-
Postcondiciones	-La información es almacenada en la base de datos del sistema. -Se muestra la forma principal.
Prototipo	
Ver anexo II.19	

**Tabla 2.3** Descripción del caso de uso de sistema Gestionar usuario

Caso de Uso	Ver usuarios conectados.
Actores	Usuario, Administrador.
Propósito	Ver los usuarios que se encuentran trabajando en el sistema en cierto momento.
Resumen	
<p>Se inicia cuando el actor desea ver los usuarios que se encuentran trabajando en el sistema.</p> <p>El caso de uso culmina cuando los datos son mostrados al actor.</p>	
Referencias	R 3
Precondiciones	-Debe existir al menos un usuario registrado en el sistema.
Postcondiciones	-La información es mostrada al actor.
Prototipo	
Ver anexo II.34	

**Tabla 2.4** Descripción del caso de uso de sistema Ver usuarios conectados.

Caso de Uso	Gestionar teoría.
Actores	Administrador
Propósito	Insertar, eliminar o modificar la teoría en el sistema.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar teoría en el sistema se muestra el formulario donde este inserta la teoría y sus autores.</li> <li>2. Cuando el administrador desea modificar una teoría existente. Se muestran las teorías que contiene el sistema, selecciona la teoría que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar una teoría existente. Se muestran las teorías que contiene el sistema, selecciona la teoría que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 4.1, R 4.2, R 4.3
Precondiciones	La teoría que se desea modificar o eliminar debe encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.16, II.17, II.18	

**Tabla 2.5** Descripción del caso de uso de sistema Gestionar teoría.

Caso de Uso	Ver teoría.
Actores	Administrador, Usuario.
Propósito	Acceder a las fuentes teóricas que brinda el sistema.
Resumen	
<p>Se inicia cuando el actor desea acceder a la teoría del sistema.</p> <p>El caso de uso termina cuando la teoría es mostrada.</p>	
Referencias	R 5.

Precondiciones	Debe estar insertada la teoría previamente en el sistema.
Postcondiciones	Se muestra al cliente la teoría.
Prototipo	
Ver anexo II.22	

**Tabla 2.6** Descripción del caso de uso de sistema Ver teoría.

Caso de Uso	Gestionar problemas propuestos.
Actores	Administrador.
Propósito	Insertar, eliminar o modificar los problemas propuestos en el sistema.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar problemas propuestos en el sistema. Se muestra el formulario donde este inserta los problemas propuestos.</li> <li>2. Cuando el administrador desea modificar problemas propuestos. Se muestran los problemas propuestos que contiene el sistema, selecciona el que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar problemas propuestos. Se muestran los problemas propuestos que contiene el sistema, selecciona el que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 6.1, R 6.2, R 6.3
Precondiciones	Los problemas propuestos que se desean modificar o eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.10, II.11, II.12	

**Tabla 2.7** Descripción del caso de uso de sistema Gestionar problemas propuestos.

Caso de Uso	Ver problemas propuestos.
Actores	Administrador, Usuario.
Propósito	Acceder a los problemas propuestos que brinda el sistema.
Resumen	
<p>Se inicia cuando el actor desea acceder a los problemas propuestos que brinda el Sistema.</p> <p>El caso de uso termina cuando los problemas propuestos son mostrados.</p>	
Referencias	R 7.
Precondiciones	Los problemas propuestos deben estar previamente insertados en el sistema.
Postcondiciones	Se muestra al cliente los problemas propuestos.
Prototipo	
Ver anexo II.31	

**Tabla 2.8** Descripción del caso de uso de sistema Ver problemas propuestos.

Caso de Uso	Gestionar problemas resueltos.
Actores	Administrador
Propósito	Insertar, eliminar o modificar los problemas resueltos en el sistema.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar problemas resueltos en el sistema. Se muestra el formulario donde este inserta los problemas resueltos.</li> <li>2. Cuando el administrador desea modificar problemas resueltos. Se muestran los problemas resueltos que contiene el sistema, selecciona el que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar problemas resueltos. Se muestran los problemas resueltos que contiene el sistema, selecciona el que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	

Referencias	R 8.1, R 8.2, R 8.3.
Precondiciones	Los problemas resueltos que se desean modificar o eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.7, II.8, II.9	

**Tabla 2.9** Descripción del caso de uso de sistema Gestionar problemas resueltos.

Caso de Uso	Ver problemas resueltos.
Actores	Administrador, Usuario.
Propósito	Acceder a los problemas resueltos que brinda el sistema.
Resumen	
<p>Se inicia cuando el actor desea acceder a los problemas resueltos que brinda el sistema.</p> <p>El caso de uso termina cuando los problemas resueltos son mostrados.</p>	
Referencias	R 9.
Precondiciones	Los problemas resueltos deben estar previamente insertados en el sistema.
Postcondiciones	Se muestra al cliente los problemas resueltos.
Prototipo	
Ver anexo II.32	

**Tabla 2.10** Descripción del caso de uso de sistema Ver problemas resueltos.

Caso de Uso	Eliminar traza.
Actores	Administrador.
Propósito	Eliminar una traza en el sistema.



Resumen	
Se inicia por dos vías distintas cuando el administrador desea eliminar una traza del sistema.	
El caso de uso culmina con la actualización de los datos en el sistema.	
Referencias	R 10
Precondiciones	La traza que se desea eliminar debe encontrarse en el sistema.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.36	

**Tabla 2.11** Descripción del caso de uso de sistema Eliminar traza.

Caso de Uso	Ver trazas.
Actores	Administrador, Usuario.
Propósito	Acceder a las trazas de los usuarios en el sistema para conocer su desempeño en el mismo.
Resumen	
Se inicia cuando el actor accede a las trazas de los usuarios para monitorear el recorrido, desempeño y resultados de los usuarios en el sistema.	
El caso de uso concluye cuando los datos de las trazas son mostrados al actor.	
Referencias	R 11.
Precondiciones	Deben existir trazas de usuarios en el sistema.
Postcondiciones	Se muestran al administrador las trazas del sistema.
Prototipo	
Ver anexo II.33	

**Tabla 2.12** Descripción del caso de uso de sistema Ver trazas.

Caso de Uso	Gestionar Webs.
-------------	-----------------

Actores	Administrador
Propósito	Insertar, eliminar o modificar los sitios web que los usuarios pueden consultar sobre la Programación Lineal.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar sitios web en el sistema. Se muestra el formulario donde este inserta los sitios web.</li> <li>2. Cuando el administrador desea modificar sitios web. Se muestran los sitios web que contiene el sistema, selecciona el que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar sitios web. Se muestran los sitios web que contiene el sistema, selecciona el que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 12.1, R 12.2, R 12.3.
Precondiciones	Los sitios web que se desean modificar o eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.23, II.24, II.25	

**Tabla 2.13** Descripción del caso de uso de sistema Gestionar Webs.

Caso de Uso	Ver Webs.
Actores	Administrador, Usuario.
Propósito	Acceder a los sitios web que los usuarios pueden consultar sobre la Programación Lineal.
Resumen	
<p>Se inicia cuando el actor desea acceder a los sitios web.</p> <p>El caso de uso termina cuando los sitios web son mostrados.</p>	
Referencias	R 13.

Precondiciones	Los sitios web deben estar previamente insertados en el sistema.
Postcondiciones	Se muestra al cliente los sitios web.
Prototipo	
Ver anexo II.37	

**Tabla 2.14** Descripción del caso de uso de sistema Ver Webs.

Caso de Uso	Gestionar ejercicios de selección.
Actores	Administrador
Propósito	Insertar, eliminar o modificar un ejercicio de selección del sistema.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar un ejercicio de selección. Se muestra el formulario donde este inserta el ejercicio.</li> <li>2. Cuando el administrador desea modificar un ejercicio de selección. Se muestran los ejercicios de selección que contiene el sistema, selecciona el que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar un ejercicio de selección. Se muestran los un ejercicios de selección que contiene el sistema, selecciona el que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 14.1, R 14.2, R 14.3
Precondiciones	Los ejercicios de selección que se desean modificar o eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.13, II.14, II.15	

**Tabla 2.15** Descripción del caso de uso de sistema Gestionar ejercicios de selección.

Caso de Uso	Gestionar ejercicios de enlazar.
-------------	----------------------------------

Actores	Administrador
Propósito	Insertar, eliminar o modificar un ejercicio de enlazar del sistema.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> <li>1. Cuando el administrador desea insertar un ejercicio de enlazar. Se muestra el formulario donde este inserta el ejercicio.</li> <li>2. Cuando el administrador desea modificar un ejercicio de enlazar. Se muestran los ejercicios de enlazar que contiene el sistema, selecciona el que desea modificar y realiza los cambios.</li> <li>3. Cuando el administrador desea eliminar un ejercicio de enlazar. Se muestran los un ejercicios de enlazar que contiene el sistema, selecciona el que desea eliminar y se elimina del sistema.</li> </ol> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
Referencias	R 15.1, R 15.2, R 15.3
Precondiciones	Los ejercicios de enlazar que se desean modificar o eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.1, II.2, II.3	

**Tabla 2.16** Descripción del caso de uso de sistema Gestionar ejercicios de enlazar.

Caso de Uso	Gestionar ejercicios de verdaderos y falsos.
Actores	Administrador
Propósito	Insertar, eliminar o modificar un ejercicio de verdaderos y falsos del sistema.
Resumen	
Referencias	R 16.1, R 16.2, R 16.3
Precondiciones	Los ejercicios de verdaderos y falsos que se desean modificar o

	eliminar deben encontrarse en la base de datos.
Postcondiciones	Se actualizan los cambios en la base de datos.
Prototipo	
Ver anexo II.4, II.5, II.6	

**Tabla 2.17** Descripción del caso de uso de sistema Gestionar ejercicios de verdaderos y falsos.

Caso de Uso	Resolver autoexamen.
Actores	Administrador, Usuario.
Propósito	Que el actor pueda evaluar los conocimientos adquiridos en el tutorial.
Resumen	
Se inicia cuando el actor desea resolver el autoexamen.	
<p>El caso de uso termina cuando los ejercicios son completados, el autoexamen se evalúa y se le brinda la nota al actor, y en caso de no haber obtenido la nota máxima, los errores que este cometió.</p>	
Referencias	R 17
Precondiciones	Debe existir la cantidad necesaria de preguntas de cada tipo para generar el autoexamen.
Postcondiciones	Se evalúa el autoexamen y se guardan las trazas del actor.
Prototipo	
Ver anexo II.35	

**Tabla 2.18** Descripción del caso de uso de sistema Resolver autoexamen.

Caso de Uso	Ver Reporte Teoría
Actores	Administrador, Usuario.
Propósito	Conocer cuáles son las teorías que contiene la base de datos.
Resumen	

Se inicia cuando el actor desea conocer cuáles son las fuentes teóricas que contiene el sistema con sus respectivos autores. El caso de uso termina cuando los nombres de las teorías y sus autores son mostrados.	
Referencias	R 19
Precondiciones	Deben existir teorías insertadas en la base de datos.
Postcondiciones	Se muestra al actor el reporte.
Prototipo	
Ver anexo II.26	

**Tabla 2.19** Descripción del caso de uso de sistema Ver Reporte Teoría.

Caso de Uso	Ver Reporte Usuarios del Sistema.
Actores	Administrador, Usuario.
Propósito	Conocer cuáles son los usuarios que se encuentran registrados en el sistema.
Resumen	
Se inicia cuando el actor desea conocer cuáles son los usuarios que se encuentran registrados en el sistema.  El caso de uso termina cuando los datos de los usuarios del sistema se muestran al actor.	
Referencias	R 20
Precondiciones	Deben existir usuarios en la base de datos.
Postcondiciones	Se muestra al actor el reporte.
Prototipo	
Ver anexo II.27	

**Tabla 2.20** Descripción del caso de uso de sistema Ver Reporte Usuarios del Sistema.

Caso de Uso	Ver Reporte Usuarios Evaluados por Período.
Actores	Administrador, Usuario.

Propósito	Conocer los datos de los usuarios que se evaluaron en un período y la fecha de su evaluación.
Resumen	
Se inicia cuando el actor desea conocer que usuarios se evaluaron en cualquier período de tiempo.  El caso de uso termina cuando los datos de los usuarios son mostrados al actor.	
Referencias	R 21
Precondiciones	-
Postcondiciones	Se muestra al actor el reporte.
Prototipo	
Ver anexo II.28	

**Tabla 2.21** Descripción del caso de uso de sistema Ver Reporte Usuarios Evaluados por Período.

Caso de Uso	Ver Reporte Usuarios por Nota.
Actores	Administrador, Usuario.
Propósito	Conocer los datos de los usuarios que obtuvieron una nota determinada.
Resumen	
Se inicia cuando el actor desea conocer que usuarios obtuvieron una nota determinada.  El caso de uso termina cuando los datos de los usuarios son mostrados al actor.	
Referencias	R 22
Precondiciones	Deben existir usuarios evaluados en la base de datos.
Postcondiciones	Se muestra al actor el reporte.
Prototipo	
Ver anexo II.29	

**Tabla 2.22** Descripción del caso de uso de sistema Ver Reporte Usuarios por Nota.

Caso de Uso	Ver Reporte Usuarios que más se Evalúan.
-------------	--

Actores	Administrador, Usuario.
Propósito	Conocer los datos de los usuarios que mayor cantidad de evaluaciones tengan.
Resumen	
<p>Se inicia cuando el actor desea conocer que usuarios tienen mayor cantidad de evaluaciones.</p> <p>El caso de uso termina cuando los datos de los usuarios son mostrados al actor.</p>	
Referencias	R 23
Precondiciones	Deben existir usuarios evaluados en la base de datos.
Postcondiciones	Se muestra al actor el reporte.
Prototipo	
Ver anexo II.30	

**Tabla 2.23** Descripción del caso de uso de sistema Ver Reporte Usuarios que más se Evalúan.



## **2.8 Conclusiones**

En el presente capítulo queda definido el modelo de objetos del dominio del tutorial. Se definieron y representaron los principales conceptos asociados al dominio del problema. Tras lograr la modelación del dominio de los procesos actuales se ha logrado una mejor comprensión de los problemas que se tienen que resolver, y así derivar los requerimientos para la aplicación.

## **Capítulo 3: Construcción de la Solución Propuesta**

### **3.1 Introducción**

El propósito del diseño es decidir cómo se llevará a cabo el sistema, jugando su papel en la parte del proceso de desarrollo de software.

En este capítulo se plasman los resultados de la etapa de diseño del sistema, utilizando UML para su modelado. Se describe la puesta en práctica de la construcción de la solución propuesta. Se presenta el diagrama de clases del diseño y el diagrama de componentes. Se plantea también los estándares de diseño y de programación que se tuvieron en cuenta es decir las tecnologías usadas para la realización de este Tutorial.

### **3.2 Diagrama de clases del diseño**

En estos diagramas es donde se pueden ver las relaciones entre todas las clases del sistema, además de los atributos y operaciones de cada una de las mismas.

#### **Diagrama de clases del diseño:**

CU Gestionar Ejercicios de Enlazar (ver Anexo I. 1)

CU Gestionar Ejercicios de Verdaderos y Falsos (ver Anexo I. 2)

CU Gestionar SS (ver Anexo I. 3)

CU Gestionar Ejercicio Resuelto (ver Anexo I. 4)

CU Gestionar Ejercicio Propuesto (ver Anexo I. 5)

CU Gestionar Teoría (ver Anexo I. 6)

CU Gestionar Usuarios (ver Anexo I. 7)

CU Gestionar Web (ver Anexo I. 8)

CU Ver Teoría (ver Anexo I. 9)

CU Reporte Usuarios del Tutorial (ver Anexo I. 10)

CU Reporte Usuarios Evaluados por Período (ver Anexo I. 11)

CU Reporte Usuarios por Nota (ver Anexo I. 12)

CU Reporte Usuarios que más se Evalúan (ver Anexo I. 13)

CU Eliminar Traza (ver Anexo I. 1)

CU Reporte Teoría (ver Anexo I. 10)

CU Resolver Autoexamen (ver Anexo I. 15)

CU Ver Direcciones de Internet (ver Anexo I. 17)

CU Ver Problemas Propuestos (ver Anexo I. 18)

CU Ver Problemas Resueltos (ver Anexo I. 19)

CU Ver Trazas (ver Anexo I. 20)

CU Ver Usuarios Conectados (ver Anexo I. 21)

### 3.3 Diseño de la base de datos

#### 3.3.1 Diagrama de clases persistentes

En el diagrama de clases persistentes sólo aparecen las clases persistentes. Estas son las clases capaces de mantener su valor en el espacio y en el tiempo.

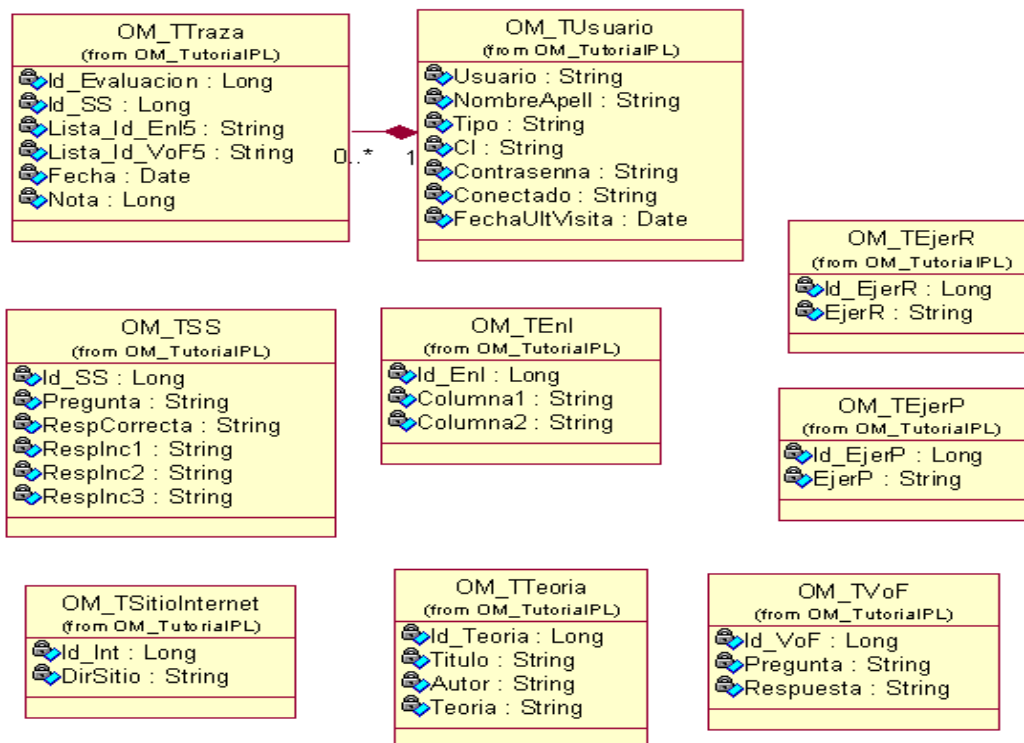


Figura 3.1 Diagrama de Clases Persistentes.

### 3.3.2 Modelos de datos

El modelo de datos describe la representación lógica y física de datos persistentes en el sistema, es generado a partir del diagrama de clases persistentes.

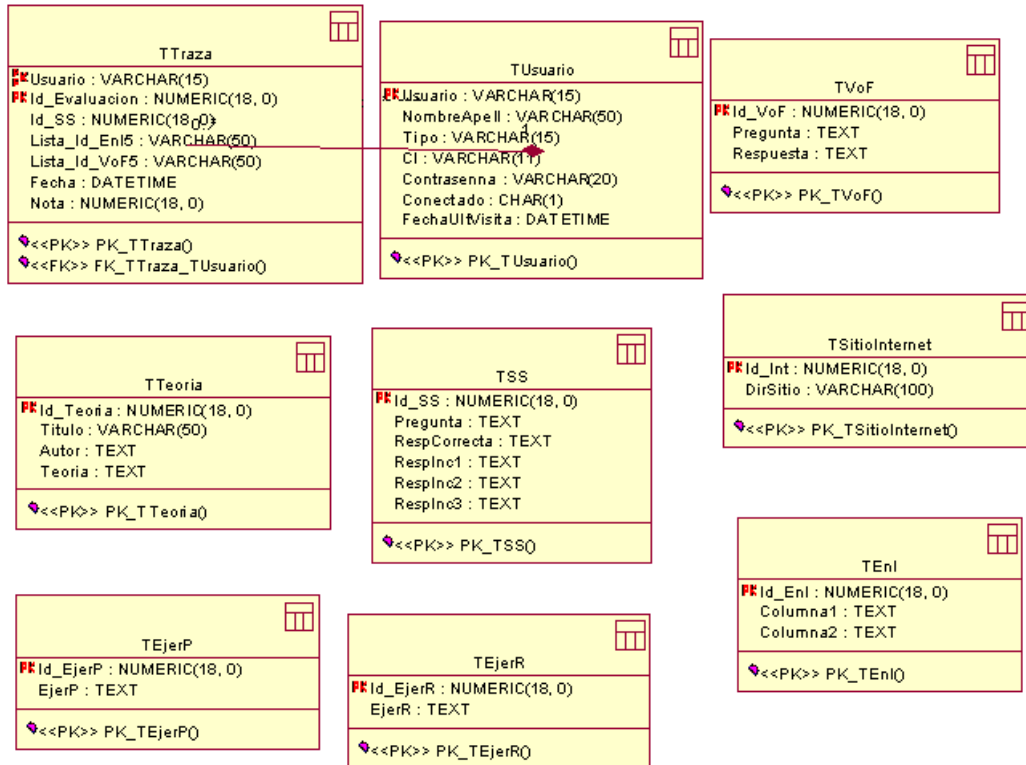


Figura 3.2 Diagrama del Modelo de Datos

### 3.4 Principios del diseño

#### 3.4.1 Estándares en la interfaz de la aplicación

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

La interfaz es en realidad un modelo mental permanente, es decir una representación cognitiva o conceptualización que el usuario hace del sistema. A fin de que este modelo

se mantenga a lo largo del programa ha de tener una consistencia, es decir mantener su coherencia de principio a fin.

Por ello se han de mantener las reglas, los criterios en la operatividad, la imagen parcial o total, etc.; pues una incoherencia de diseño puede aportar pérdidas de eficacia del propio contenido que se quiera transmitir.

La interfaz diseñada para ambos módulos del sistema está estrechamente vinculada y presenta las siguientes características:

- En todo el sistema se contrastan colores con verde claro, blanco.
- El tipo de letra utilizada es Arial de estilo regular y tamaño 8 en negrita.
- Información legible.
- No presenta una alta carga visual.
- Facilidad de aprendizaje, navegabilidad y uso.
- Representación permanente de un contexto de acción, es decir, la estructura y el acceso a los servicios es mantenida para todas las páginas del sistema.
- La entrada de información por parte de los usuarios se realiza a través de los componentes del formulario.
- El objeto de interés siempre es fácil de identificar.
- Las interacciones se basan en selecciones de tipo menú y en acciones físicas sobre elementos de código visual botones, imágenes y mensajes.
- Las operaciones que se realizan al acceder a la información almacenada en la base de datos y ficheros son rápidas e incrementales con efectos inmediatos.
- Los reportes emitidos por el sistema son estructurados en tablas.
- Presenta versiones imprimibles en blanco y negro para los reportes, gráficas y listas.

### **3.4.2 Formatos de reportes**

Los reportes se obtendrán en tablas en formato Web, utilizando una letra oscura, legible y el fondo con colores claros para no recargar y hacer engorrosa su impresión.

La aplicación tiene dentro de sus principales funcionalidades, además de mantener actualizada la teoría, los ejercicios resueltos, así como los ejercicios propuestos del

tutorial, realizar un autoexamen que le permitirá a los usuarios medir los conocimientos adquiridos en el mismo.

Los reportes se obtendrán para mostrar el estado de los usuarios y de la teoría del tutorial. Le permitirá al administrador seguir el desempeño de los usuarios en los autoexámenes realizados y la nota alcanzada en los mismos, también puede ver los usuarios que se evaluaron en un período de tiempo además de los que más se evalúan. La teoría también puede ser controlada por el administrador.

### **3.4.3 Concepción general de la ayuda**

Dentro del mundo de las aplicaciones en general, la ayuda constituye una parte importante del sistema. Las tendencias actuales apuntan a que estas no deben ser muy detallistas o extensas, sino simplemente explicaciones sencillas y aclaraciones del producto y de las operaciones que puede realizar el usuario sobre el mismo.

El sistema cuenta con una ayuda general la cual se encuentra disponible al usuario en cada momento, permitiéndole navegar por sus tópicos y consultar cualquier duda sobre el sistema, lo cual permite que el usuario pueda hacer un uso eficiente de la aplicación. La ayuda tendrá además como funciona cada parte del sistema, explicará el funcionamiento de cada ventana, con sus opciones y operaciones.

### **3.4.4 Tratamiento de excepciones**

Las situaciones que pueden provocar fallos en la ejecución normal de un programa se denominan excepciones. El sistema propuesto presenta una interfaz diseñada, implementada y dirigida a evitar tales situaciones y errores con el fin de garantizar la integridad y confiabilidad de la información que se gestiona. El sistema tiene la obligación de detectar problemas en el proceso de autenticación por parte de algún usuario. Las posibilidades de introducir información errónea por parte del usuario serán mínimas, pues aunque en muchas ocasiones el usuario teclee datos y en otras seleccione elementos de la pantalla, se mantendrá un nivel de validación de la información y en caso de errores se le comunicará el error cometido a través de mensajes de error. Los mensajes de error que emitirá el sistema se mostrarán en un lenguaje de fácil comprensión para los usuarios.

### **3.4.5 Seguridad**

El diseño del sistema tiene provisto dentro de las políticas y reglas que rigen su funcionamiento, la seguridad y protección de la información. El sistema exige una autenticación por parte de los usuarios que ingresan al sistema, con el objetivo de controlar los niveles de acceso a la información. Los usuarios sin privilegios administrativos que accedan al sistema tendrán acceso a la información, herramientas, ayuda y demás opciones del tutorial; pero no pueden modificar ni eliminar la información que el mismo contiene; sólo los administradores del sistema pueden modificar el contenido del sistema, crear cuentas de administración o eliminar cuentas de usuarios. Se puede notar además, que la consistencia de los datos es otro aspecto que se toma en cuenta, y para ello el sistema cuenta con formularios validados con funciones del lenguaje de programación utilizado que garantizan que la información que se registre en la base de datos y en los ficheros sea totalmente consistente e íntegra.

### **3.4.6 Estándares de codificación**

Definir los estándares al escribir el código para el correcto desarrollo de las aplicaciones es de suma importancia, ya que facilita el mantenimiento del software, garantiza la obtención de un código claro y comprensible, y ayuda a reducir los errores que se puedan presentar durante el desarrollo del mismo.

Existen varios aspectos que pueden hacer un código más legible; algunos de estos son el empleo de nombres descriptivos y el uso de una indentación coherente y de comentarios informativos. Se describen a continuación algunas convenciones tomadas con relación a estos aspectos para el código escrito en Object Pascal en el sistema.

Se establece que todas las líneas dentro de un método estarán indentadas con respecto a la instrucción que encabeza a éste, lo mismo ocurre con todas las líneas que conformen el cuerpo de un ciclo estructural condicional. Los inicios (begin) y cierre (end) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen y se evitan si hay sólo una instrucción. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos.

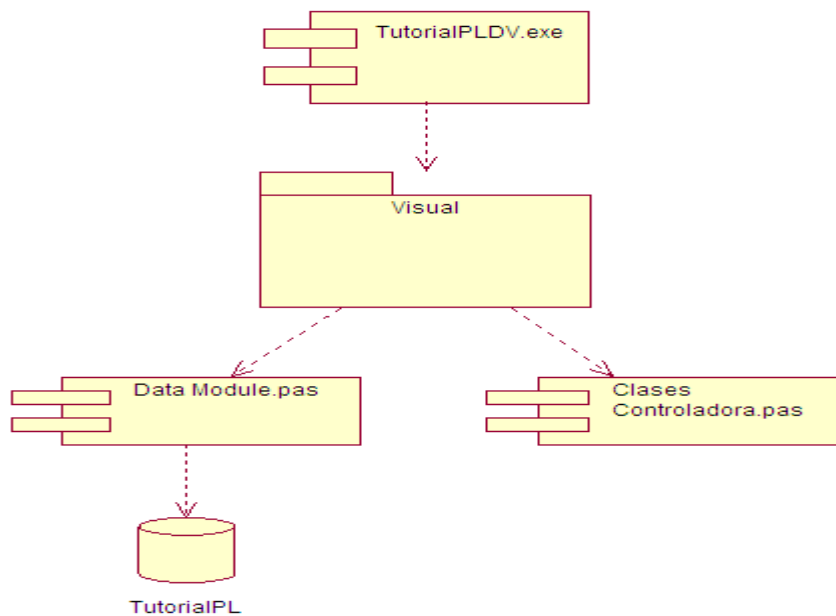


Los nombres de las variables, los controles, los procedimientos y funciones fueron adoptados lo más explicativos posibles, siempre respondiendo a su propósito. Para cada tipo de control se tuvo en cuenta el uso de los prefijos más utilizados, una muestra de esto se muestra en la siguiente tabla:

Un buen comentario añade información al código de una manera clara y ayuda a entender el objetivo del mismo. Se tomó como regla, comentar los procedimientos y funciones que resulten complejos al principio de los mismos para explicar cómo se deben usar sin necesidad de leer el código. Se comentan además algunos algoritmos que pudieran resultar de difícil comprensión.

### 3.5. Diagrama de Componentes

Un diagrama de componentes muestra un conjunto de componentes y sus relaciones. Gráficamente representan una colección de nodos o componentes y arcos, los primeros representan componentes, interfaces y los segundos relaciones de dependencia, generalización / especialización, asociación, agregación / composición y realización.



**Figura 3.3** Diagrama de componentes.

### 3.6 Modelo de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo.

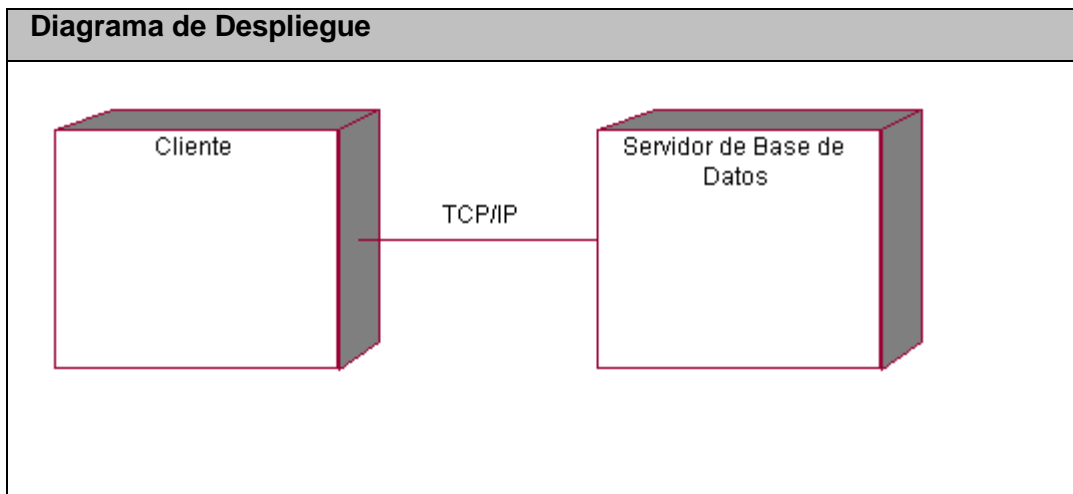


Figura. 3.4 Diagrama de despliegue.

### **3.7 Conclusiones**

En el presente capítulo se mostraron los resultados de la etapa de diseño del sistema. Se desarrollaron los diagramas de clases, el diseño de la base de datos y el modelo de implementación.

Se describieron la concepción del tratamiento de errores y de la ayuda, los principios de codificación, de diseño y el formato de los reportes.

## Capítulo 4: Estudio de la Factibilidad

### 4.1 Introducción

El estudio de factibilidad de un proyecto constituye una parte importante antes de la ejecución del mismo. Permite saber si lo que se está realizando tiene un costo favorable en dependencia del esfuerzo y recursos necesarios a utilizar.

Ofrece además un estimado del tiempo de desarrollo del producto así como de la cantidad de personas necesarias para realizarlo.

### 4.2 Planificación

#### *Entradas Externas*

Nombre de la entrada externa	Cant. de ficheros	Cant. de elem. de datos	Clasificación
Crear Usuario	1	5	Simple
Modificar Usuario	1	5	Simple
Eliminar Usuario	1	1	Simple
Cambiar Contraseña	1	1	Simple
Insertar Teoría	1	2	Simple
Modificar Teoría	1	2	Simple
Eliminar Teoría	1	1	Simple
Insertar Ejercicio Resuelto	1	1	Simple
Modificar Ejercicio Resuelto	1	2	Simple
Eliminar Ejercicio Resuelto	1	1	Simple
Insertar Ejercicio Propuesto	1	1	Simple
Modificar Ejercicio Propuesto	1	1	Simple
Eliminar Ejercicio Propuesto	1	1	Simple
Insertar Sitio de Internet	1	1	Simple
Modificar Sitio de Internet	1	1	Simple
Eliminar Sitio de Internet	1	1	Simple
Insertar Ejercicio de Verdadero o Falso	1	2	Simple

Modificar Ejercicio de Verdadero o Falso	1	2	Simple
Eliminar Ejercicio de Verdadero o Falso	1	2	Simple
Insertar Ejercicio de Selección Simple	1	5	Simple
Modificar Ejercicio de Selección Simple	1	5	Simple
Eliminar Ejercicio de Selección Simple	1	5	Simple
Insertar Ejercicio de Enlace	1	2	Simple
Modificar Ejercicio de Enlace	1	2	Simple
Eliminar Ejercicio de Enlace	1	1	Simple
Eliminar Trazas	1	1	Simple
<b>Total</b>	<b>Simple:26</b>	<b>Media: 0</b>	<b>Compleja: 0</b>

### **Salidas externas**

<b>Nombre de la salida externa</b>	<b>Cant. de ficheros</b>	<b>Cant. de elem. de datos</b>	<b>Clasificación</b>
Ver Teoría	1	2	Simple
Ver Ejercicios Resueltos	1	1	Simple
Ver Ejercicios Propuestos	1	1	Simple
Ver Sitios de Internet	1	2	Simple
Ver Usuarios Conectados	1	5	Simple
Reporte de Usuarios del Tutorial	1	5	Simple
Reporte de la Teoría	1	3	Simple
Reporte de los Usuarios que más se evalúan	2	5	Simple
<b>Total</b>	<b>Simple:8</b>	<b>Media: 0</b>	<b>Compleja: 0</b>

### **Peticiones**

<b>Nombre de la petición</b>	<b>Cant. de ficheros</b>	<b>Cant. de elem. de datos</b>	<b>Clasificación</b>
Reporte de Usuarios por Nota	2	7	Simple
Reporte de usuarios evaluados por período	2	5	Simple
<b>Total</b>	<b>Simple:2</b>	<b>Media: 0</b>	<b>Compleja: 0</b>

**Ficheros internos**

Nombre del fichero interno	Cant. de records	Cant. de elem. de datos	Clasificación
TTeoría	1	3	Simple
TUsuario	1	7	Simple
TEjercicioResuelto	1	2	Simple
TEjercicioPropuesto	1	2	Simple
TSitiosInternet	1	2	Simple
TTraza	1	7	Simple
TVoF	1	3	Simple
TEnl	1	3	Simple
TSS	1	6	Simple
<b>Total</b>	<b>Simple:9</b>	<b>Media: 0</b>	<b>Compleja: 0</b>

Elementos	S	X Peso	M	X Peso	C	X Peso	PF. Subtotal
Ficheros lógicos internos	9	(*7)	0	(*10)	0	(*15)	63
Ficheros de interfaces externas	0	(*5)	0	(*7)	0	(*10)	0
Entradas externas	26	(*3)	0	(*4)	0	(*6)	78
Salidas externas	8	(*4)	0	(*5)	0	(7)	32
Peticiones	2	(*3)	0	(*4)	0	(6)	6
<b>Total</b>							<b>179</b>

**Tabla Puntos de Función sin ajustar.**

**Cálculo de las instrucciones fuentes:**

El cálculo de las instrucciones fuentes, según COCOMO II, se basa en la cantidad de instrucciones por punto de función que genera el lenguaje de programación empleado.

Características	Valor		
Puntos de función desajustados	178		
Lenguaje	Object Pascal	HTML	SQL
% de utilización en la aplicación	75% (≈134)	5% (≈9)	20% (≈36)
Instrucciones fuentes por puntos de función	29	43	39
Instrucciones fuentes	3886	387	1404
Total Instrucciones fuentes (LOC)	5677		

**Tabla Instrucciones fuentes.**

### 4.3 Costos

#### Multiplicadores de esfuerzo:

Multiplicador	Descripción	Valor
RCPX	La complejidad del producto es media.	1.00
RUSE	Se implementa código reutilizable para su aprovechamiento en el proyecto.	1.07
PDIF	Considerando que la plataforma es estable y el tiempo de almacenamiento es menor que un 50% se estableció un nivel bajo para este multiplicador.	0.87
PERS	La capacidad de los especialistas es alta.	0.83
PREX	La experiencia en este tipo de aplicación es alta.	0.87
FCIL	Se utilizan herramientas e instrumentos de programación modernos.	1.00
SCED	Se estima que el producto se hará en el tiempo establecido.	1.00
		<b>0.67</b>

**Tabla 4.9 Definición de los Multiplicadores de Esfuerzo (MEj).**

7

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 0.67$$

i=1

**Factores de Escala:**

<b>Factor</b>	<b>Descripción</b>	<b>Valor</b>
PREC	Aspectos novedosos.	2.48
FLEX	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa.	3.04
TEAM	Interacciones principalmente cooperativas.	1.10
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan. Poca incertidumbre, riesgos no críticos	1.41
PMAT	Relación con el proceso de madurez del software. Nivel 3.	3.12
5 $\sum_{i=1} SFi$		11.15

**Tabla 4.8 Definición de los valores de los Factores de Escala (SFi).**

$$SF = \sum_{i=1}^5 SFi = PREC + FLEX + RESL + TEAM + PMAT = 11.15$$

**Valores calibrados:**

$$A = 2.94; B = 0.91; C = 3.67; D = 0.28$$

$$E = B + 0.01 * \sum SFi = 0.91 + 0.01 * 11.15 = 1.02$$

$$F = D + 0.2 * (E - B) = 0.28 + 0.2 * (1.02 - 0.91) = 0.302$$

**Cálculo del esfuerzo (PM):**

$$PM = A * (MSLOC)^E * \Pi Emi = 2.94 * (5,677)^{1.02} * 0.67 = 11.58 \approx 12 \text{ Hombres/Mes.}$$

**Cálculo del tiempo de desarrollo:**



$$TDEV = C * PM^F = 3.67 * (12)^{0.302} = 7,77 \approx 8 \text{ meses (Estimado)}$$

**Cálculo de la cantidad de hombres:**

$$CH = PM / TDEV = 12/8 \approx 2 \text{ hombres}$$

Como el equipo de trabajo está formado realmente por 1 persona, se vuelve a calcular el tiempo de desarrollo para la cantidad real de hombres.

$$CH^* = 1 \text{ hombres.}$$

$$TEDV = PM/CH^* = 12/1 = 12 \text{ meses.}$$

**Cálculo del costo:**

Asumiendo como salario promedio mensual (SP) \$225.00

$$CHM = CH * SP = 1 * \$225.00 = \$225.00$$

$$\text{Costo} = CHM * PM = \$225 * 12 = \$ 2700,00$$

- Cálculos

Cálculo de:	Valor
Esfuerzo (PM: Hombres - mes)	12 Hombres/Mes
Tiempo de Desarrollo(meses)	12 Meses
Cantidad de Hombres	1
Costo	\$ 2700,00
Salario medio	\$ 225.00

**Tabla 5.8 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.**

#### 4.4 Beneficios intangibles

Los beneficios que se obtendrán con el desarrollo del sistema propuesto son intangibles, debido a que la aplicación diseñada para la educación, permitirá a los usuarios aumentar sus conocimientos sobre la Programación Lineal más específicamente la Sistema de Restricciones.

Se dispondrá de teorías, ejercicios resueltos, ejercicios propuestos, direcciones de internet, así como una evaluación permitirá a los usuarios ejercitarse.

#### **4.5 Análisis de costos y beneficios**

El desarrollo de todo producto informático va asociado a un costo, el justificarlo depende de los beneficios tangibles e intangibles que trae consigo.

La utilización de este nuevo tutorial para la enseñanza aprendizaje de la sistema de restricciones en el tema Programación Lineal, brinda un apoyo para los estudiantes de Contabilidad y Finanzas de la UNISS, y como recurso importante, debe alcanzar una calidad óptima de almacenamiento y velocidad.

Los usuarios que interactuarán con el tutorial tendrán la oportunidad de aumentar sus conocimientos sobre la sistema de restricciones en el tema Programación Lineal.

Analizando el costo del proyecto que es de \$ 2700,00 contra los numerosos beneficios que reportará, detallados con anterioridad; y la necesidad de desarrollar una herramienta para la enseñanza aprendizaje de la sistema de restricciones en el tema Programación Lineal, solicitado por la UNISS para elevar el conocimiento de los estudiantes de contabilidad y finanzas.

## **4.6 Conclusiones**

En este capítulo se ha realizado el estudio de factibilidad del sistema propuesto, el cual arrojó los siguientes resultados: el sistema tiene un esfuerzo de 12 hombre/mes, para un tiempo de desarrollo de 12 meses, para una persona desarrollándolo, el costo que implica es de \$2700.00, calculado para un salario mensual de \$225.

Se concluye entonces que el sistema es totalmente factible y puede procederse a su implantación.

## Conclusiones

- Los antecedentes teóricos metodológicos identificados exponen la utilidad de la aplicación de los tutoriales como medio y herramienta de trabajo para estimular el proceso de enseñanza aprendizaje de los estudiantes. Por otro lado permitió determinar que se emplearía la metodología RUP y el lenguaje UML para realizar el diseño de la propuesta, y las herramientas Borland Delphi 7 y SQL Server 2000 para la implementación.
- Se diseñó una propuesta de solución, partiendo de la descripción de los procesos del dominio y de la modelación de los casos de uso correspondientes, así como los requisitos funcionales y no funcionales definidos por el usuario.
- Se realizó un análisis de los costos y beneficios del tutorial en el cual se demostró la factibilidad del mismo. Los resultados de este análisis demuestran que se puede proceder a la implementación del tutorial por los beneficios que este aportaría al proceso de enseñanza aprendizaje del Sistema de Restricciones de la Programación Lineal.
- Se implementó un tutorial que siguió los principios del diseño, los estándares de codificación correctamente y se ajustó a los requerimientos funcionales del sistema.

## Recomendaciones

- Continuar profundizando en el tema y valorar su utilización en otras carreras como Ingeniería Informática, en la que se imparte la Programación Lineal.
- Extender el tutorial a otras universidades del país.

## **Bibliografía**

Alvarez de Zayas, C. M. (1999). *La escuela en la vida*. La Habana: Pueblo y Educación.

Baelo Álvarez, R., & Cantón Mayo, I. (2009). Las tecnologías de la información y la comunicación en la educación superior. *Revista Iberoamericana de Educación* .

Blanco, L. (1999). Hiperuniversidad: nueva alternativa educativa. *GIGA: La revista cubana de Computación* , 48-51.

Bravo, D. R. (2004). *Software para la elaboración de Mapas Conceptuales*. ISPJAE, Ciudad de la Habana.

Castellanos, D., & otros. (2002). *Aprender y enseñar en la escuela: una concepción desarrolladora*. Ciudad de la Habana.

Castellanos, D., & otros. (2001). *El proceso de enseñanza-aprendizaje desarrollador en la Secundaria Básica*. Centro de Estudios Educativos I.S.P. "Enrique José Varona", La Habana.

Colom, A., Salinas, J., & Sureda, J. (1988). *Tecnología y medios educativos*. Madrid: Cincel-kapelusz.

Danilov, M. A. (2003). *El proceso de enseñanza en la escuela*. España.

Delgado, M. (2002). *Tesis presentada en opción al Título de Máster en Informática Educativa*. Ciudad de la Habana.

Díaz Antón, M. G., Pérez, M. A., Grimmán, A. C., & Mendoza, L. E. *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática*.

Forniero, R., Arencibia, V., & Hernández, R. (2002). *Las tecnologías de la información y la Comunicación en la formación inicial y continua de los profesionales de la Educación Retos*. Ciudad de la Habana.

García, D., & López, E. (s.f.). Sistema Tutor para la enseñanza del Algebra Lineal. *Revista de Ingeniería Industrial* .

Haag, S., Cummings, M., & McCubbrey, D. (2004). *Management information systems for the information age*. New York: McGraw-Hill.

Jacobson, I. (2000). *El Proceso Unificado de Desarrollo de software*.

Jacobson, I. (2004). *El Proceso Unificado de Desarrollo de Software* (Vol. I y II). La Habana: Félix Varela.

Labarrere, A. F. (1988). *Bases psicopedagógicas de la enseñanza de la solución de problemas matemáticos en la escuela cubana*. La Habana: Pueblo y Educación.

Majimutov, M. I. (1983). *La enseñanza problémica*. La Habana: Pueblo y Educación.

Majó, J., & Marqués, P. (2002). *La revolución educativa en la era de internet*. Barcelona: Cisspraxis.

*Manual de SQL*. (s.f.). Obtenido de <http://walter.freeservers.com>

Martínez, F. S. (1996). *La enseñanza ante los nuevos canales de información*. Madrid.

Matos, R. M. *Introducción al Trabajo con Base de Datos*.

Microsoft Corporation. (1993-2005). Microsoft ® Encarta ® 2006.

OCDE(Organisation for Economic Co-operation and Development. (2002). *Reviewing the ICT sector definition: Issues for discussion*. Recuperado el 8 de Mayo de 2006, de <http://www.oecd.org/dataoecd/3/8/20627293.pdf>

Pacios Fernández, L. L., Arenas Gutiérrez, R., & Pérez Llanes, G. (2010). *Sistemas Hipermedia Adaptativos: una aproximación al tema*. Obtenido de

[http://www.google.com.cu/#hl=es&q=hipermedia.pdf&meta=&aq=&aqi=&aql=&oq=hipermedia.pdf&gs\\_rfai=&fp=bf8281a9b87c25dc](http://www.google.com.cu/#hl=es&q=hipermedia.pdf&meta=&aq=&aqi=&aql=&oq=hipermedia.pdf&gs_rfai=&fp=bf8281a9b87c25dc)

Pérez Arranz, F. (2001). *El uso cotidiano de los libros electrónicos*. Obtenido de [http://www.google.com.cu/#hl=es&q=65a1.pdf&meta=&aq=&aqi=&aql=&oq=65a1.pdf&gs\\_rfai=&fp=bf8281a9b87c25dc](http://www.google.com.cu/#hl=es&q=65a1.pdf&meta=&aq=&aqi=&aql=&oq=65a1.pdf&gs_rfai=&fp=bf8281a9b87c25dc)

Prieto, D. (1995). *Mediación pedagógica y Nuevas tecnologías*. Santafé de Bogotá.

REDVET. (2009). *Revista electrónica de Veterinaria* , 10.

Rodríguez, R. (1991). *Búsqueda y estudio de los sistemas tutoriales para la importación de la computación*.

Romero J, G., & Martínez R, R. E. *Plataforma para la Generación de Sistemas Tutoriales Inteligentes en la Capacitación de Personal Técnico de Centrales Eléctricas*.

Romero, P. G. (s.f.). *Análisis y Diseño Orientado a Objetos*.

Salazar, C., & Antonio, L. (s.f.). *Prolegómenos Sobre el Lenguaje de Modelado Unificado (UML)*. Obtenido de [http://www.google.com.cu/#hl=es&q=Lenguaje+de+Modelado+Unificado+\(UML\)&meta=&aq=&aqi=g1&aql=&oq=Lenguaje+de+Modelado+Unificado+\(UML\)&gs\\_rfai=&fp=bf8281a9b87c25dc](http://www.google.com.cu/#hl=es&q=Lenguaje+de+Modelado+Unificado+(UML)&meta=&aq=&aqi=g1&aql=&oq=Lenguaje+de+Modelado+Unificado+(UML)&gs_rfai=&fp=bf8281a9b87c25dc)

Schank, R. (1995). *Engines for Education*.

Soto, L. (4 de Marzo de 2006). *Especificación de requerimientos*. Obtenido de <http://mitecnologico.com/Main/EspecificacionesDeRequerimientos>

SQL. (s.f.). Obtenido de [www.arsys.es/soporte/programacion](http://www.arsys.es/soporte/programacion)

Tejedor, F. J., & García Valcárcel, A. *Perspectivas de las nuevas tecnologías en la educación*. Madrid: Narcea.



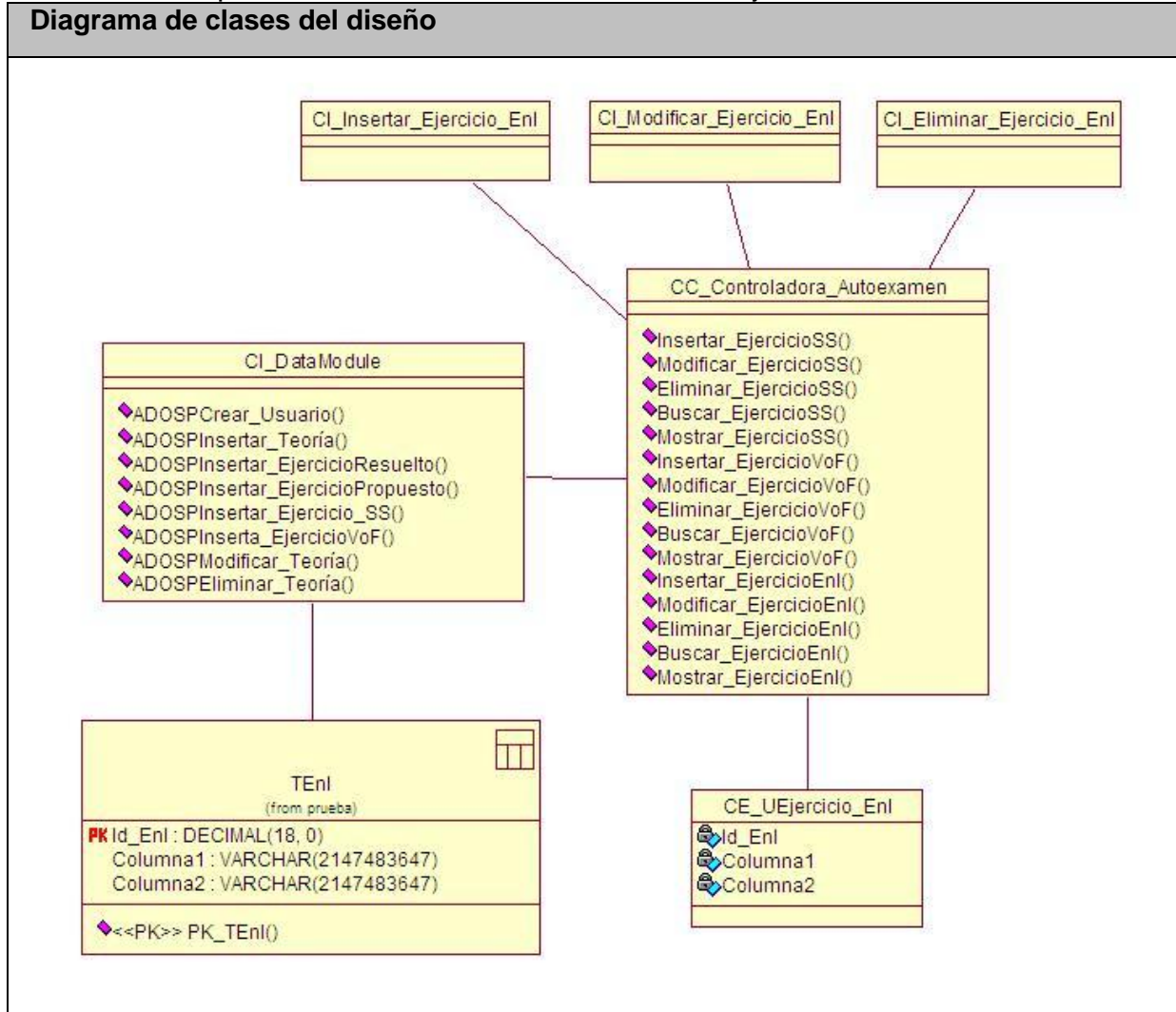
Twigg, C. (1994). *The changing Definition of learning*. Educom Review.

UNESCO. (2003). *Proyecto general de educación para América Latina y el Caribe*. La Habana.

## Anexos

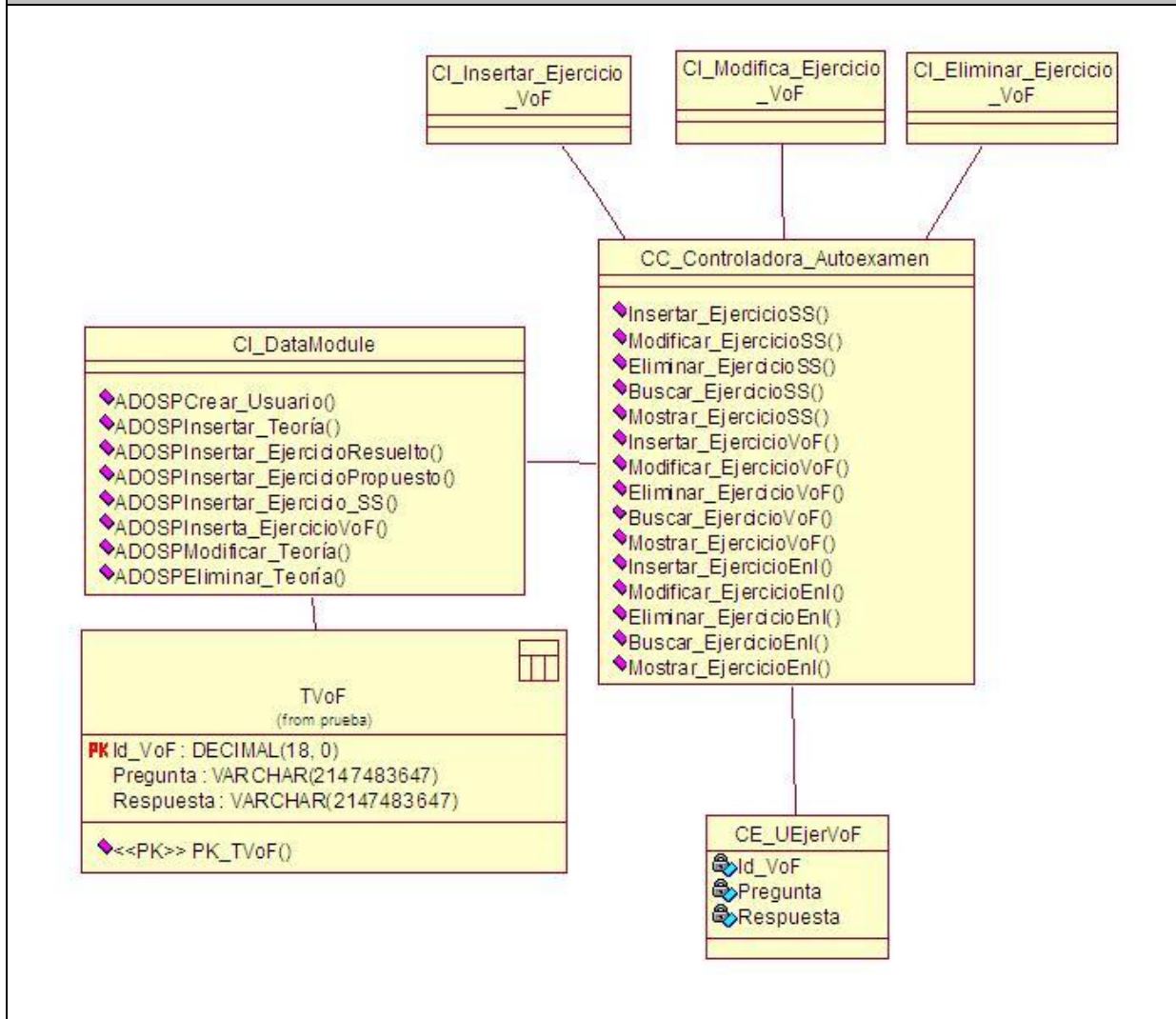
### Anexo I.1 Paquete Gestión. Caso de uso Gestionar Ejercicio de Enlace.

#### Diagrama de clases del diseño

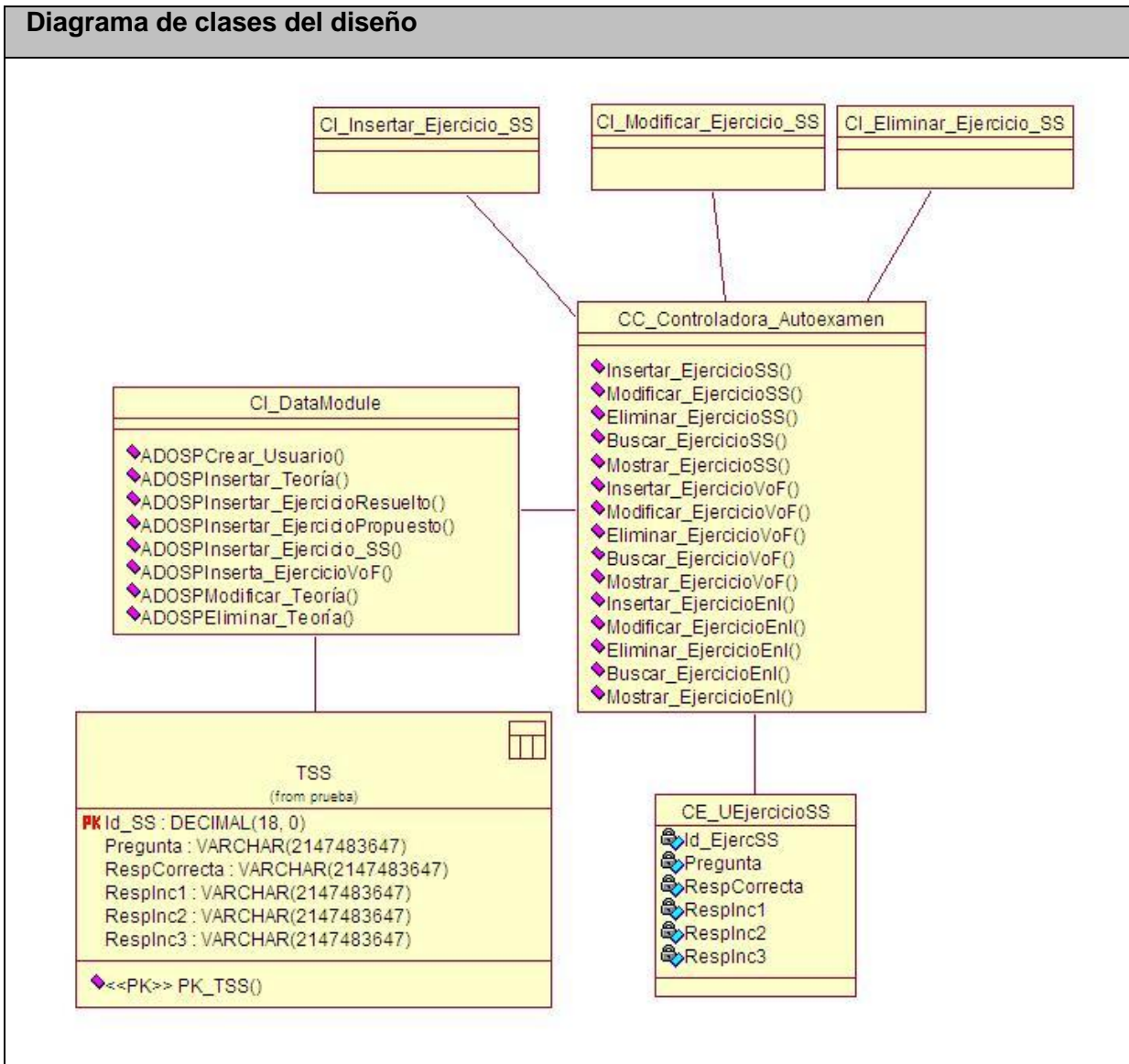


Anexo I.2 Paquete Gestión. Caso de uso Gestionar Ejercicio de Verdadero o Falso.

Diagrama de clases del diseño

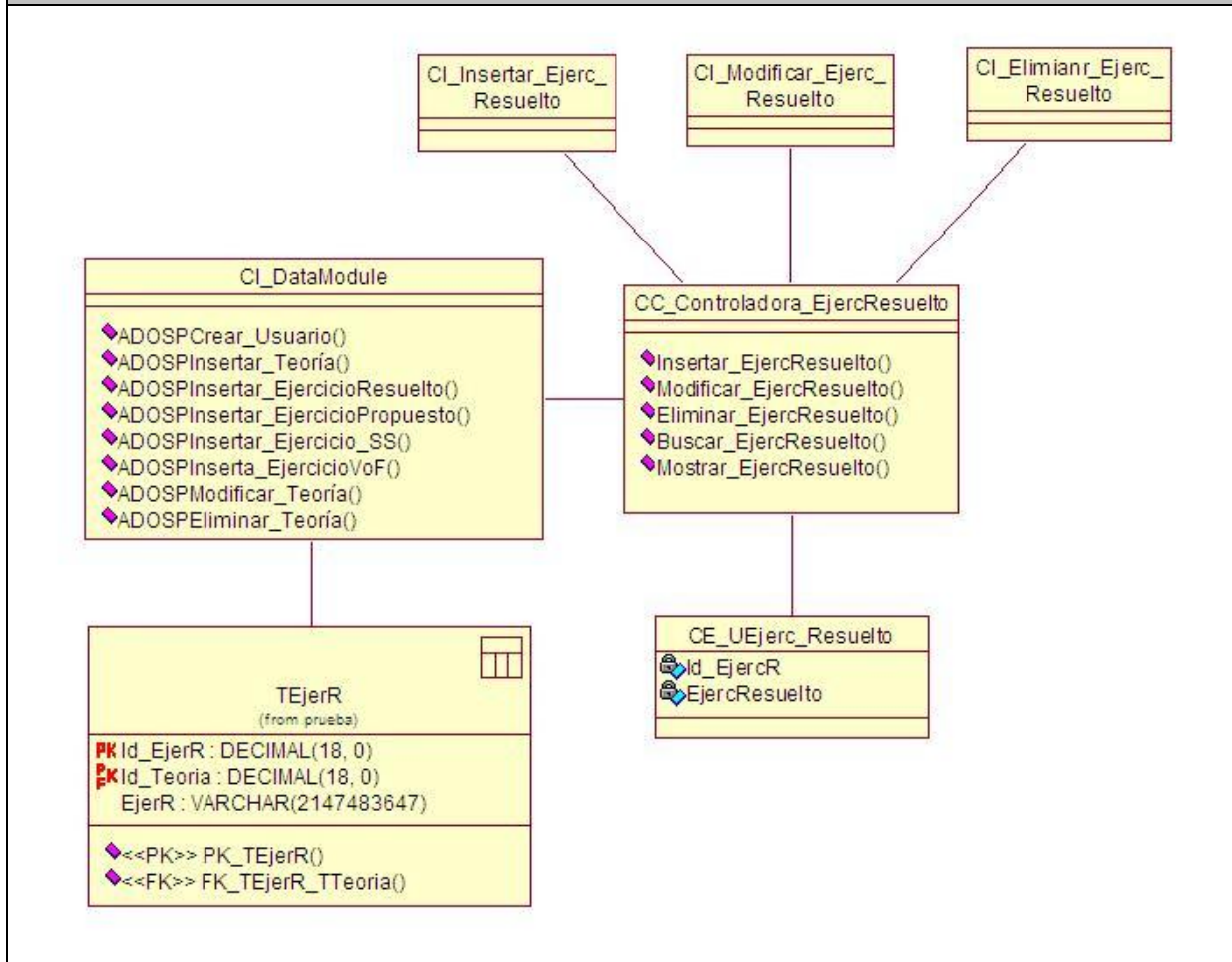


### Anexo I.3 Paquete Gestión. Caso de uso Gestionar Ejercicio de Selección Simple



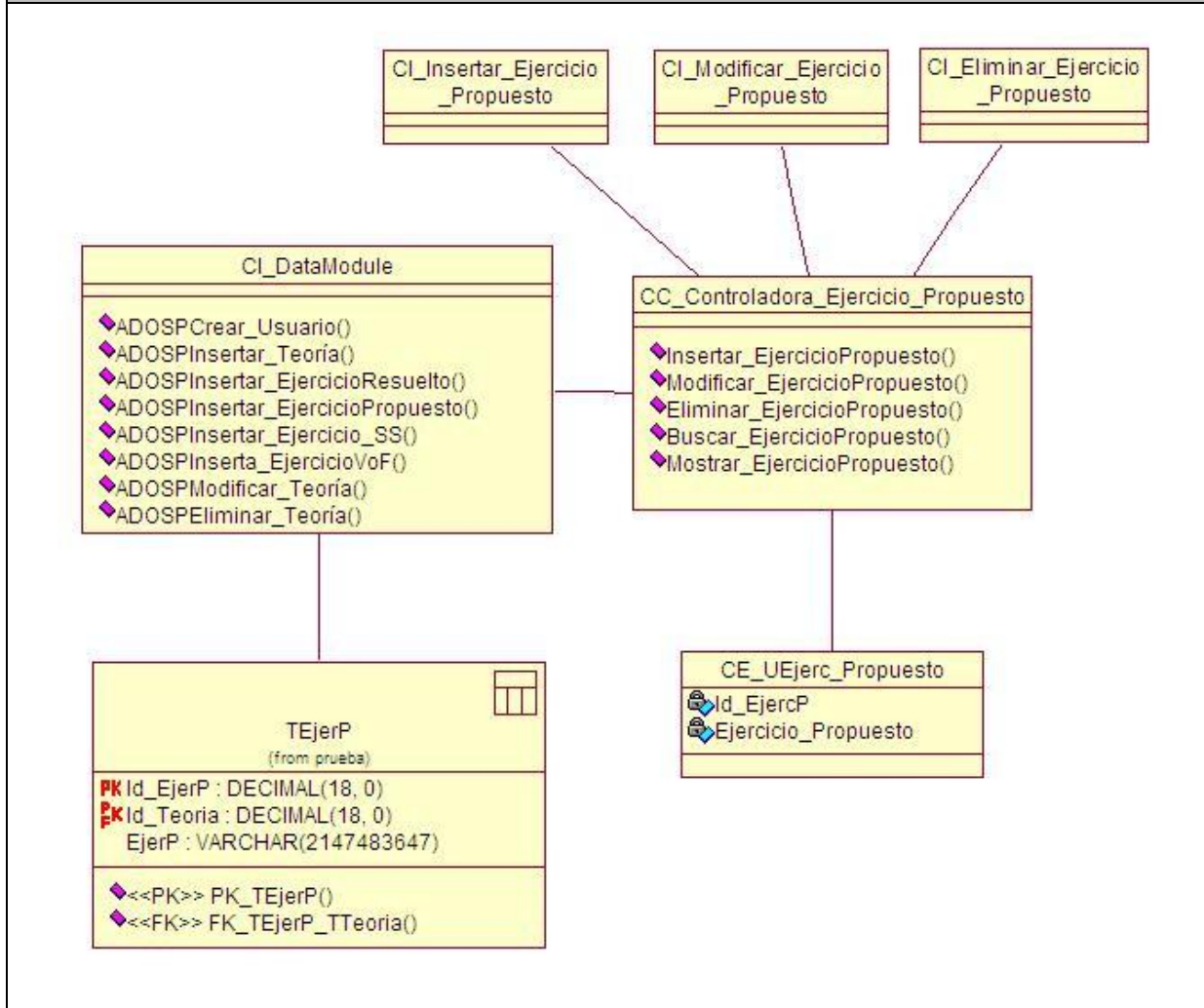
## Anexo I.4 Paquete Gestión. Caso de uso Gestionar Ejercicio Resuelto.

### Diagrama de clases del diseño



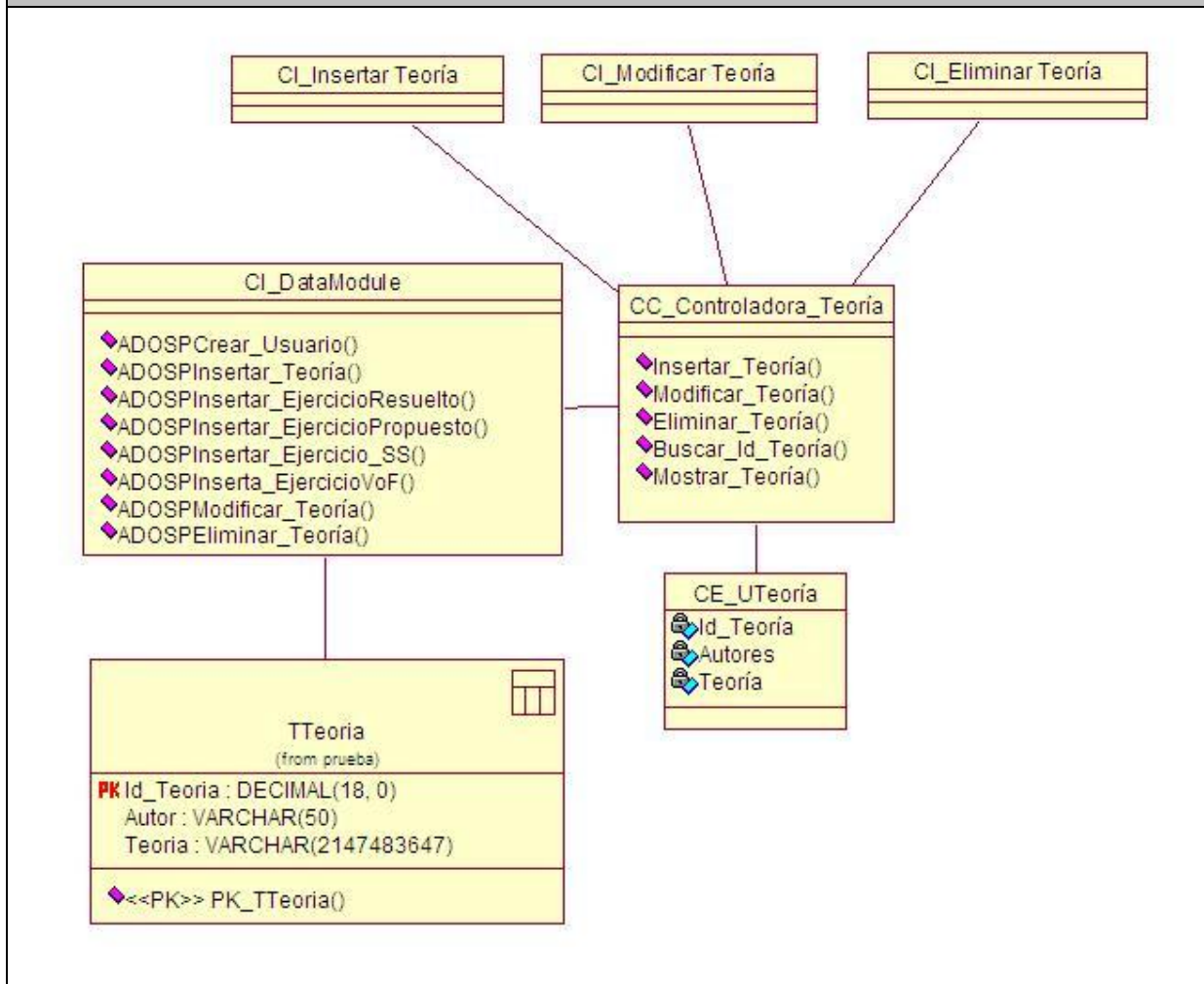
Anexo I.5 Paquete Gestión. Caso de uso Gestionar Ejercicio Propuesto.

Diagrama de clases del diseño



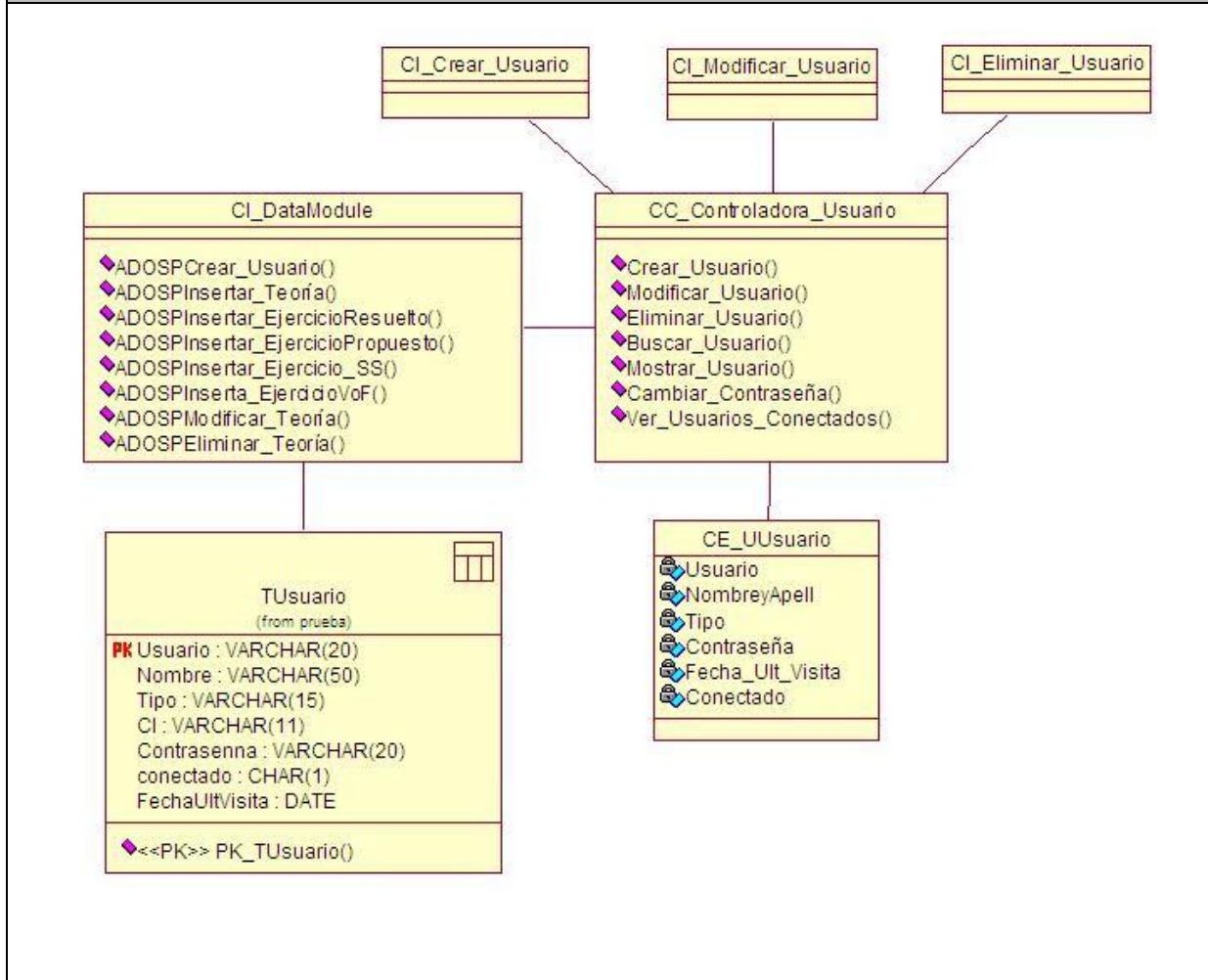
Anexo I.6 Paquete Gestión. Caso de uso Gestionar Teoría.

Diagrama de clases del diseño



Anexo I.7 Paquete Gestión. Caso de uso Gestionar Usuarios.

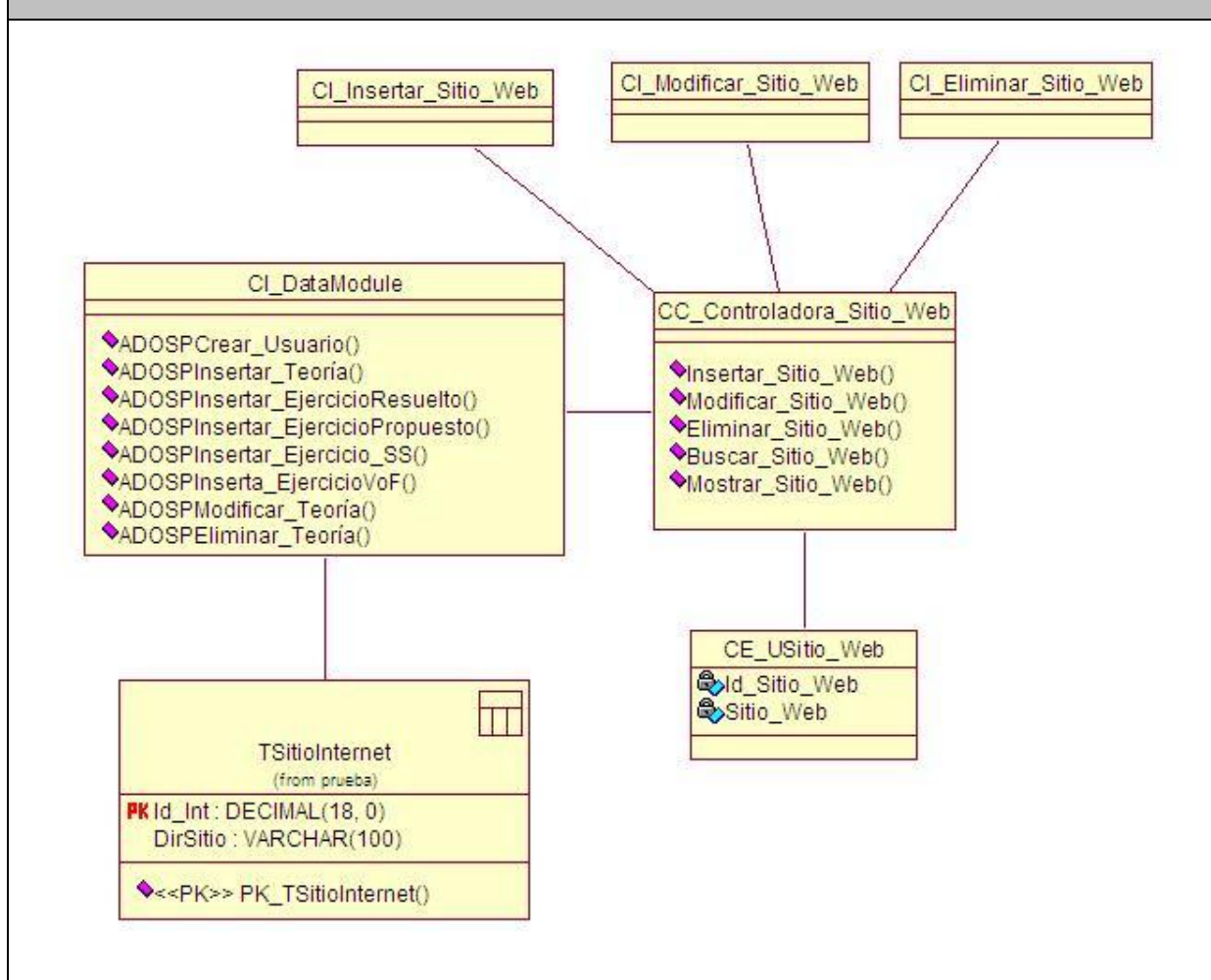
Diagrama de clases del diseño





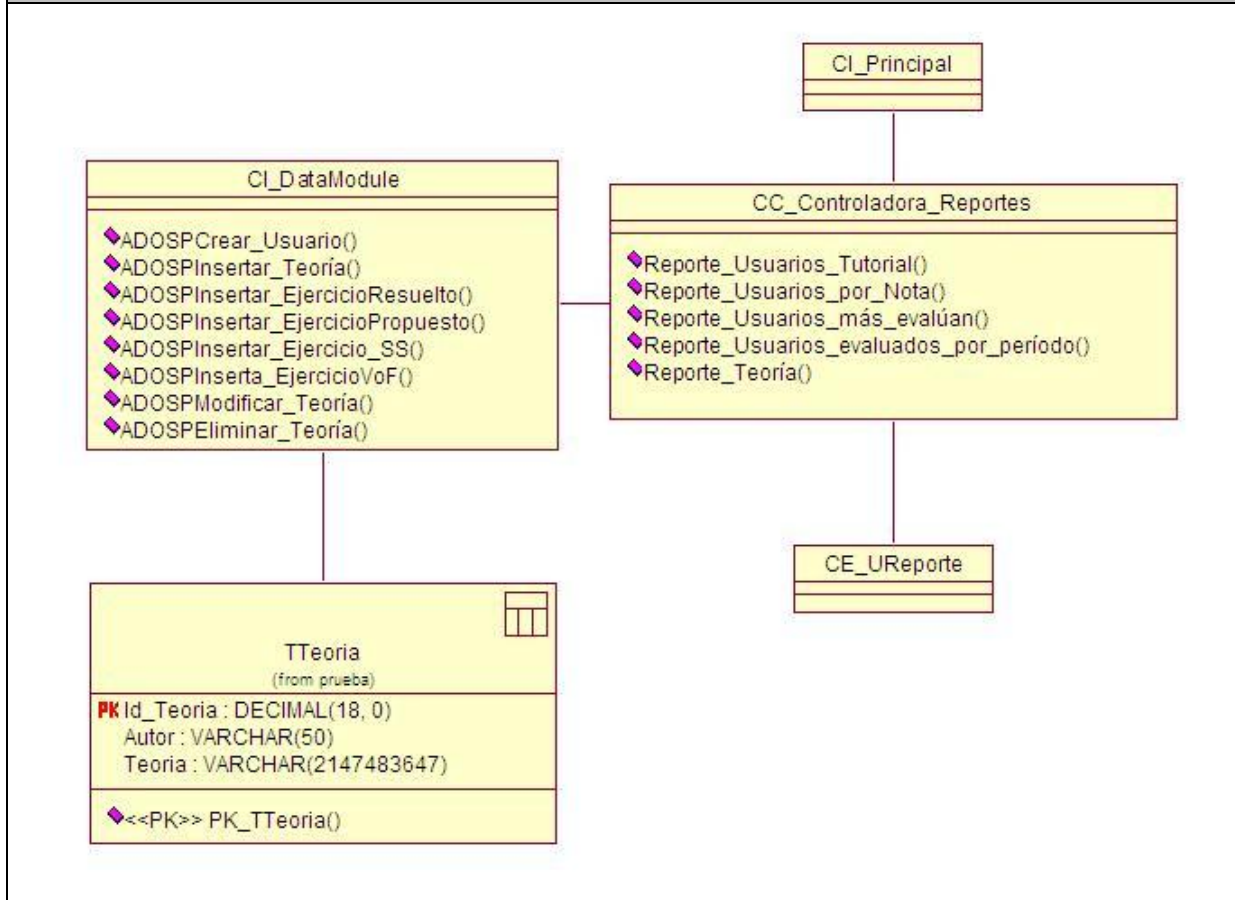
Anexo I.8 Paquete Gestión. Caso de uso Gestionar Web.

Diagrama de clases del diseño



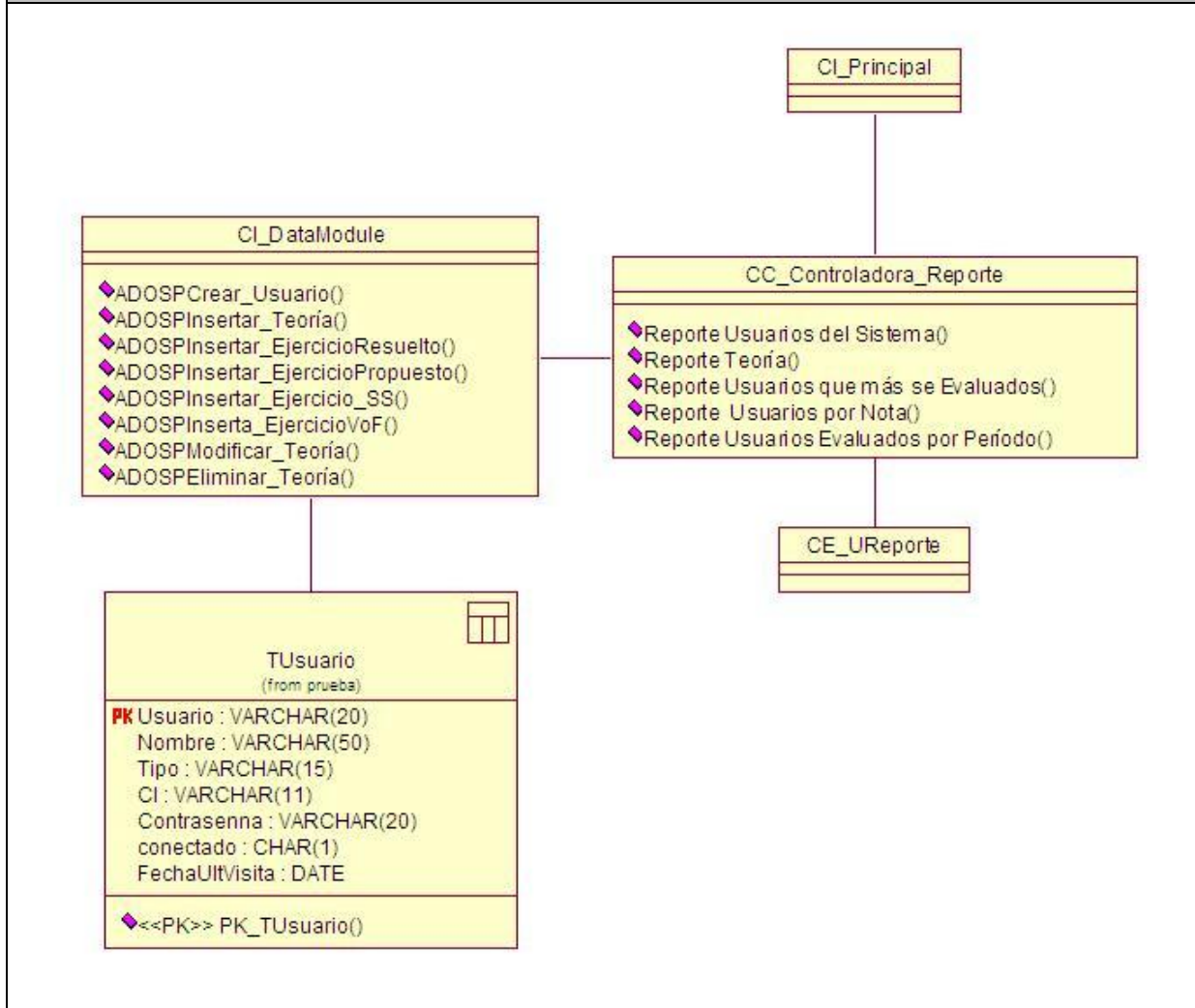
Anexo I.9 Paquete Reporte. Caso de uso Ver Teoría.

Diagrama de clases del diseño



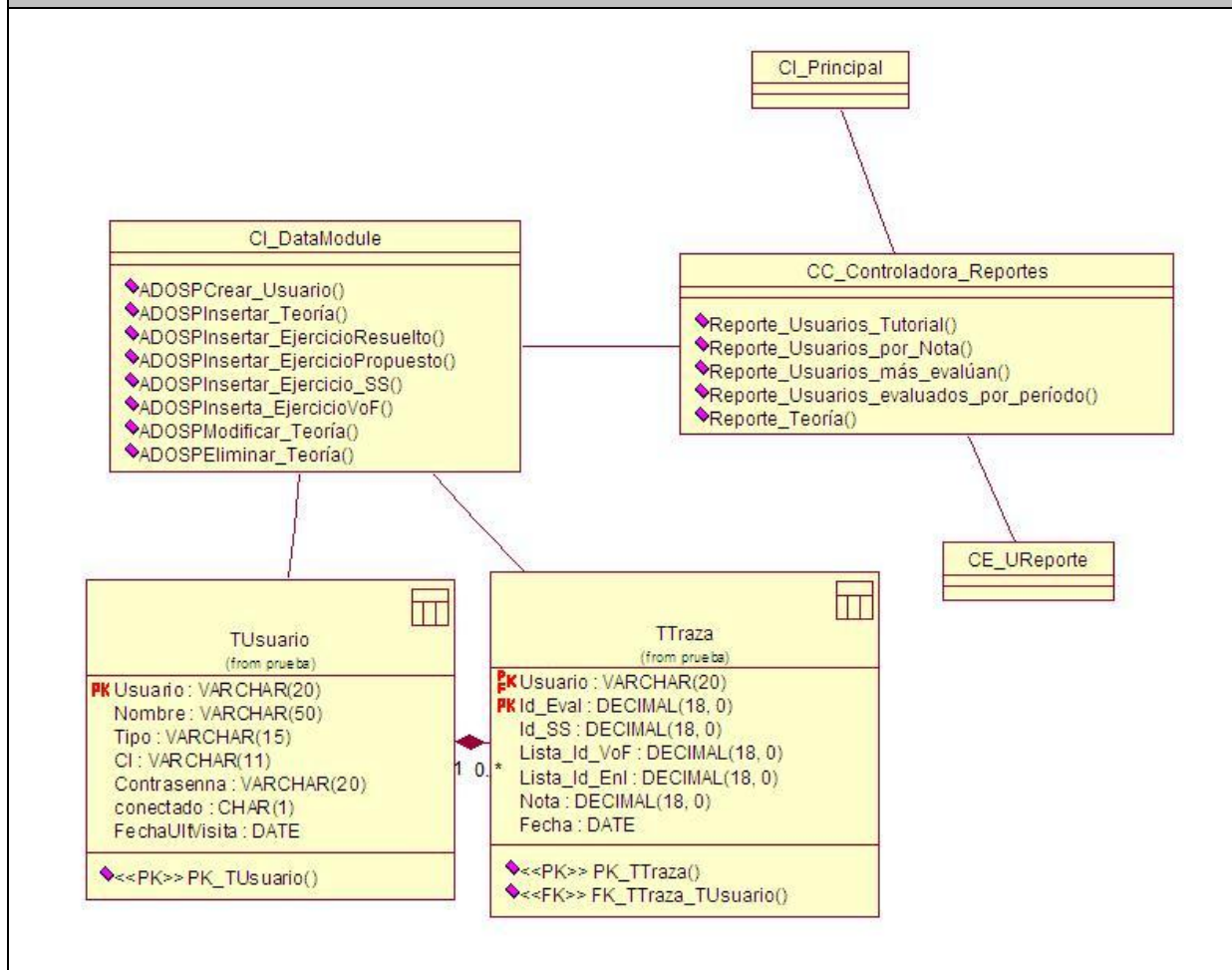
Anexo I.10 Paquete Reporte. Caso de uso Reporte Usuarios del Tutorial.

Diagrama de clases del diseño



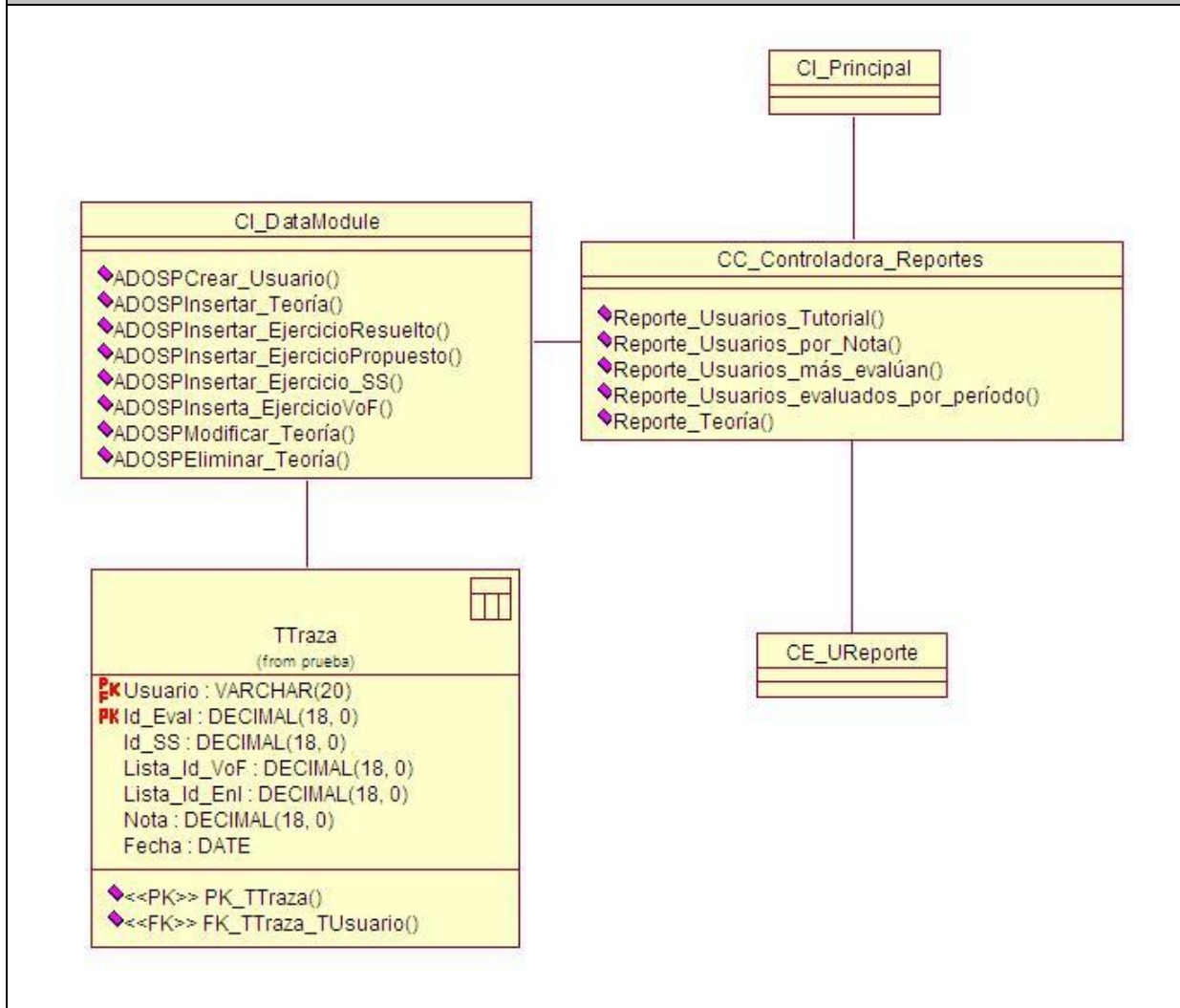
Anexo I.11 Paquete Reporte. Caso de uso Reporte Usuarios evaluados por Período.

Diagrama de clases del diseño



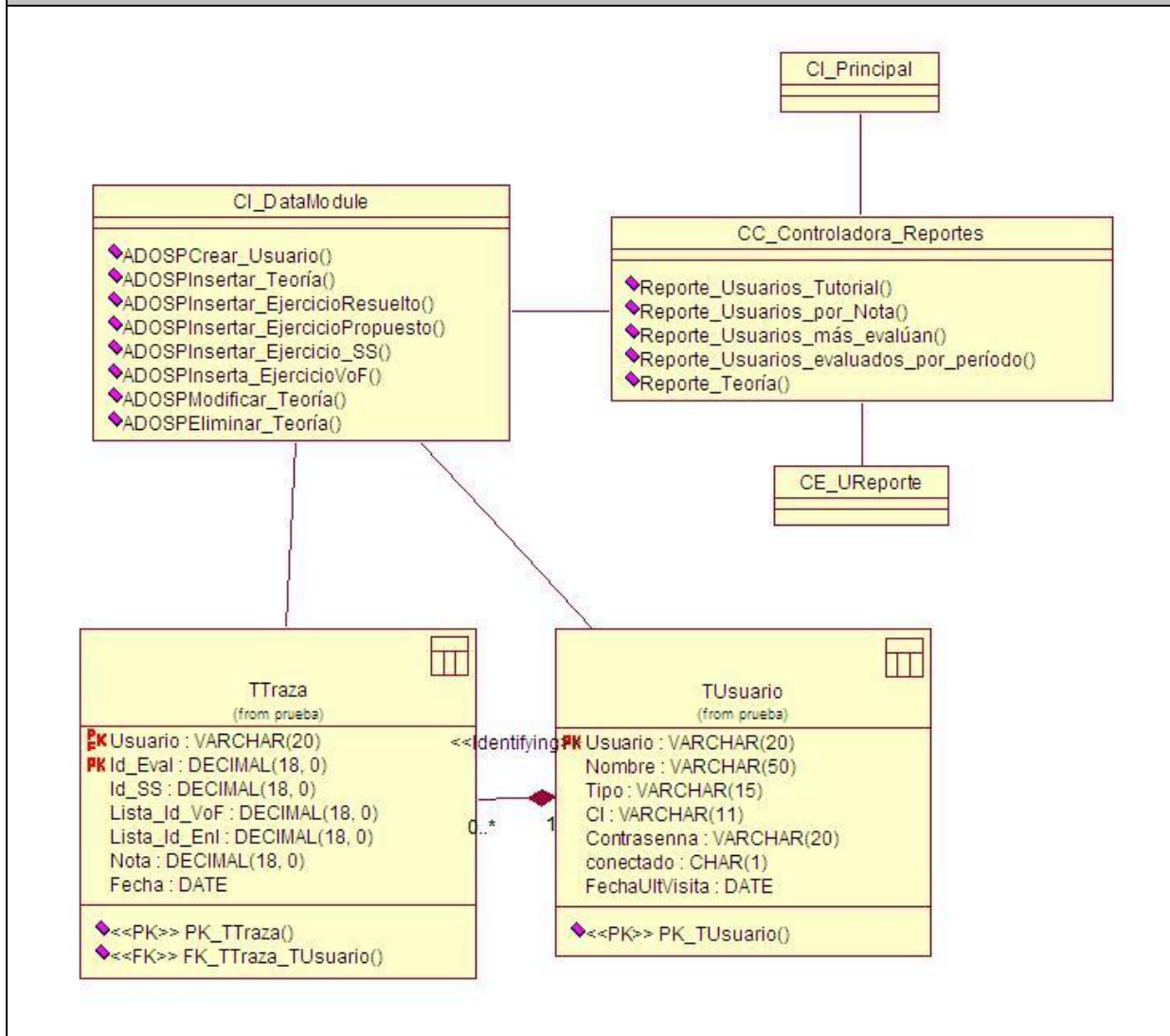
Anexo I.12 Paquete Reporte. Caso de uso Reporte Usuarios por Nota.

Diagrama de clases del diseño



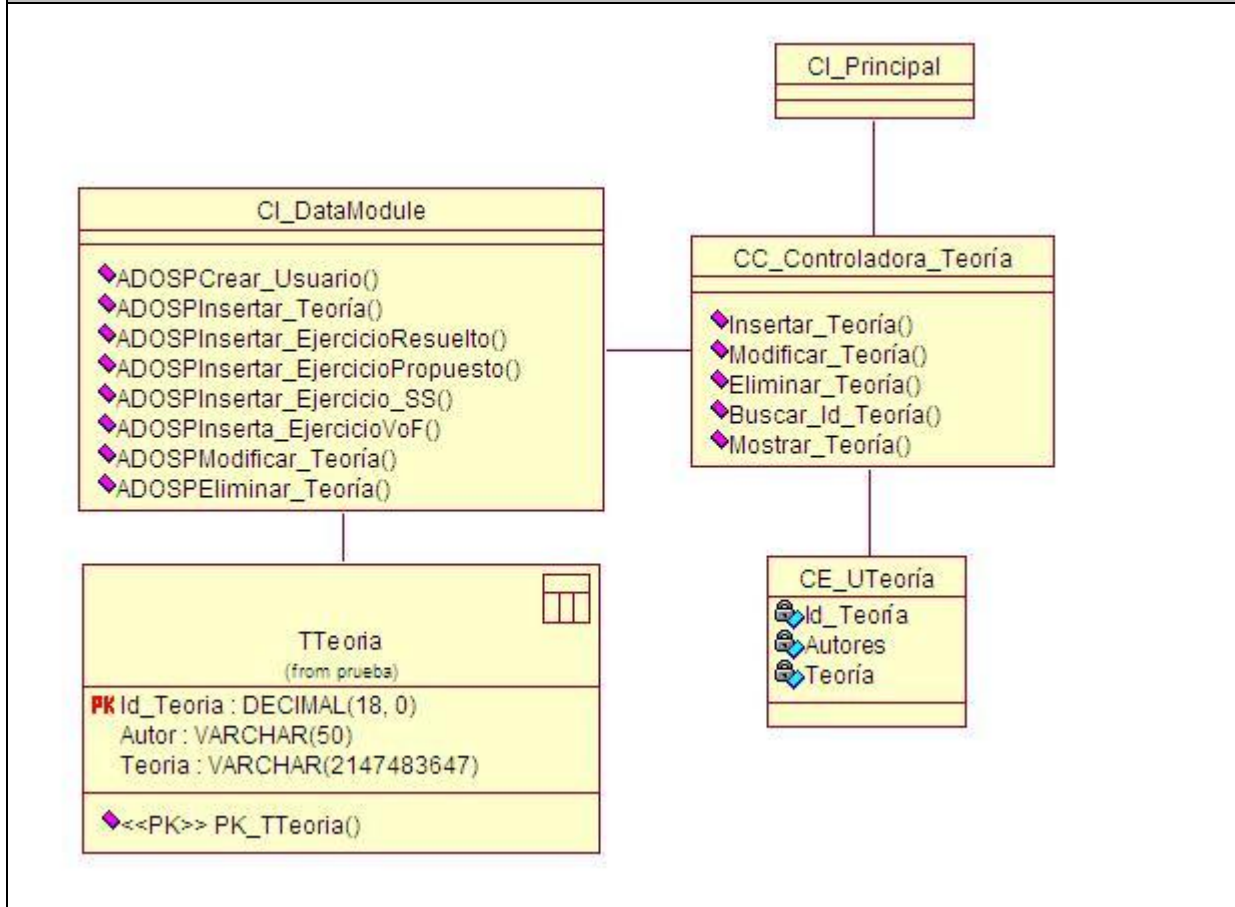
Anexo I.13 Paquete Reporte. Caso de uso Reporte Usuarios que más se evalúan.

Diagrama de clases del diseño



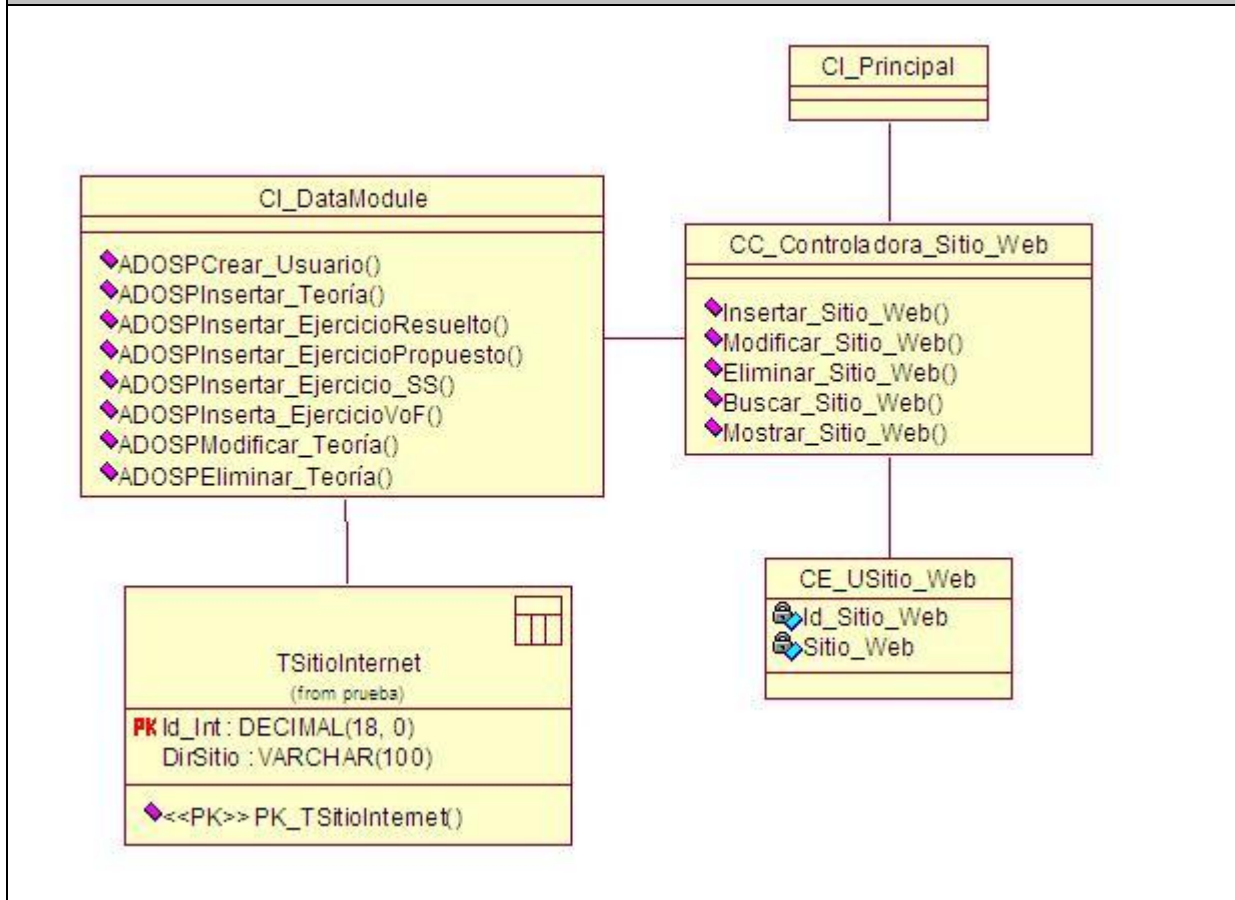
Anexo I.14 Paquete Reporte. Caso de uso Ver Teoría.

Diagrama de clases del diseño



Anexo I.15 Paquete Reporte. Caso de uso Ver direcciones de Internet.

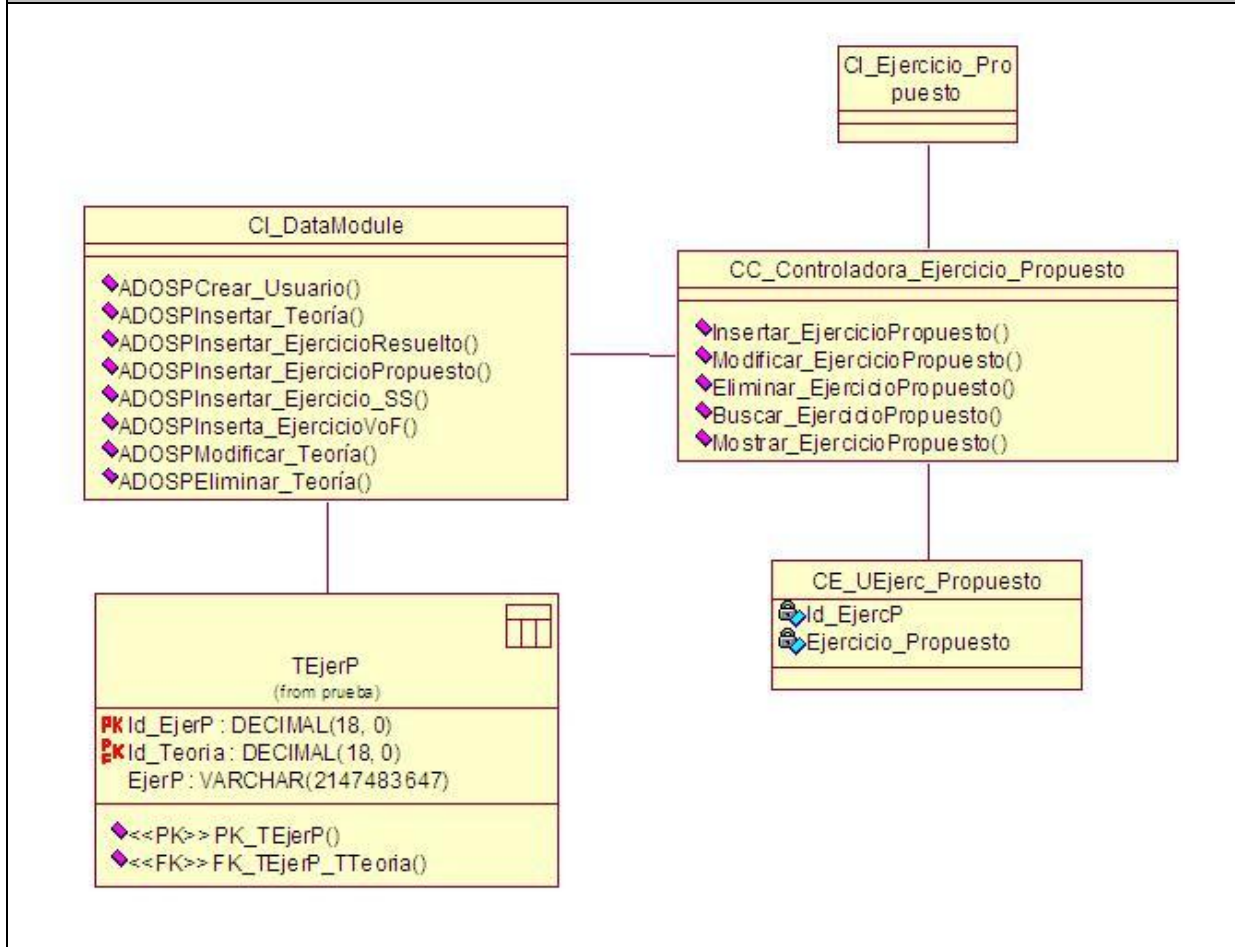
Diagrama de clases del diseño





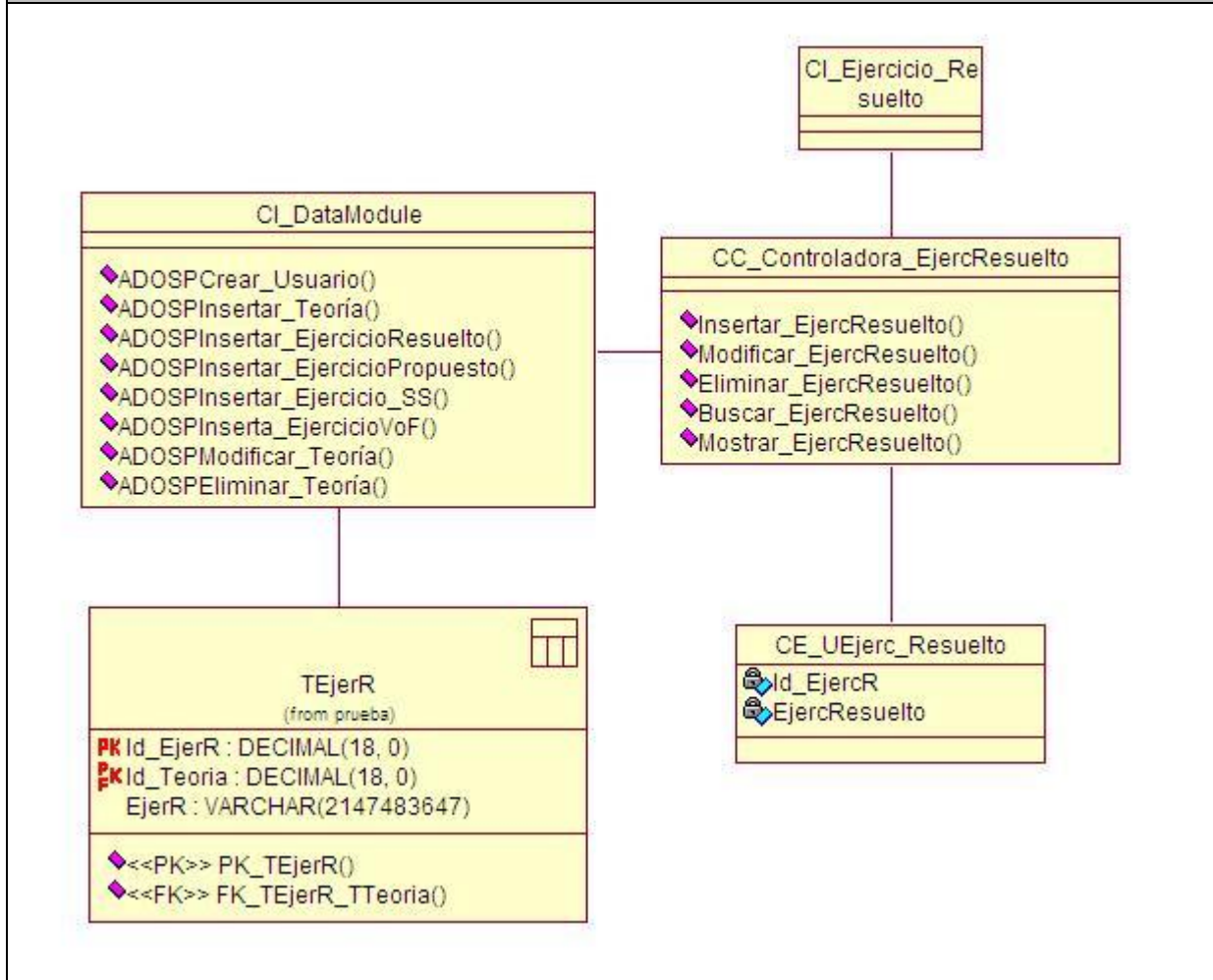
Anexo I.16 Paquete Reporte. Caso de uso Ver Problemas Propuestos.

Diagrama de clases del diseño



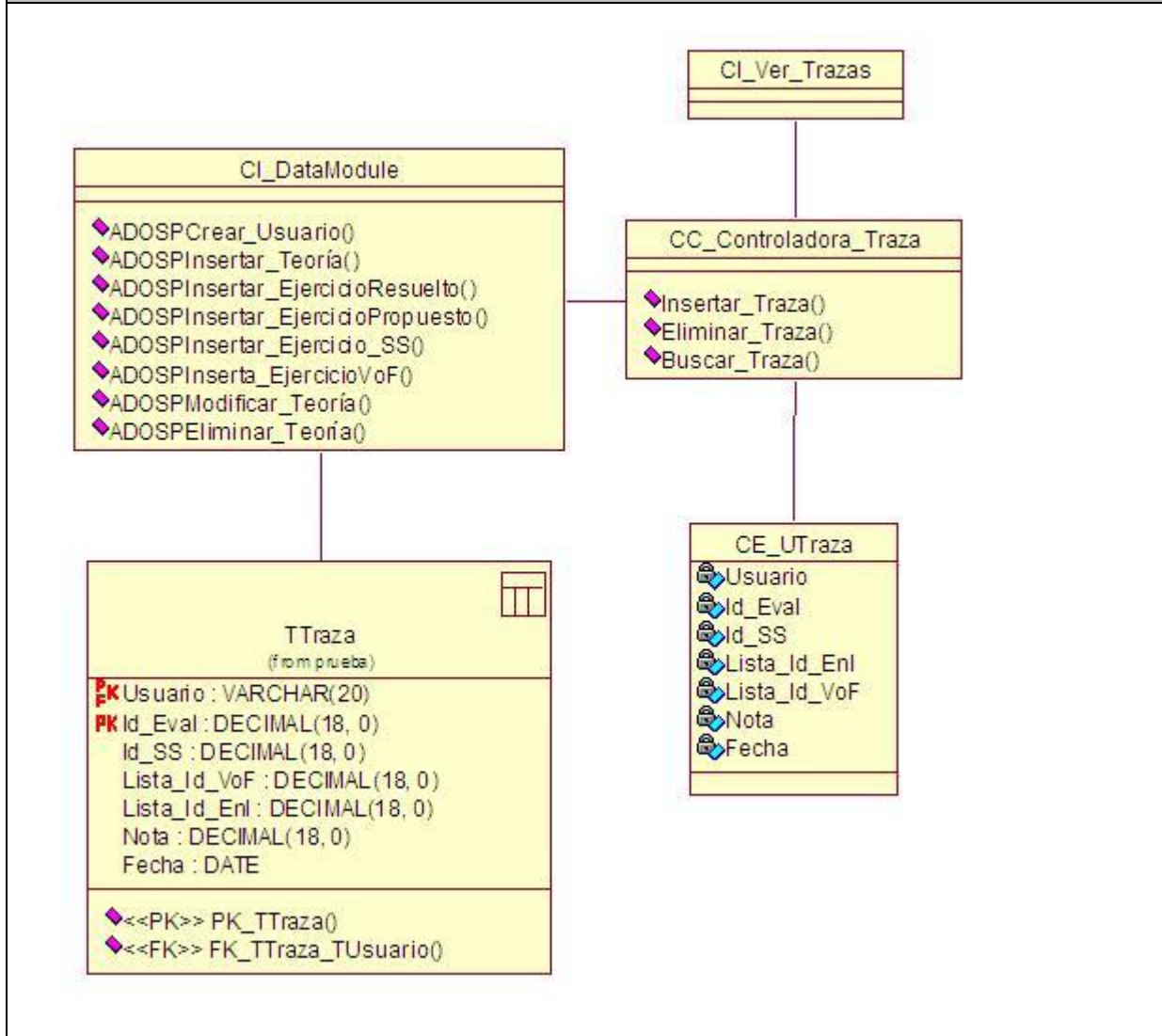
Anexo I.17 Paquete Reporte. Caso de uso Ver Problemas Resueltos.

Diagrama de clases del diseño



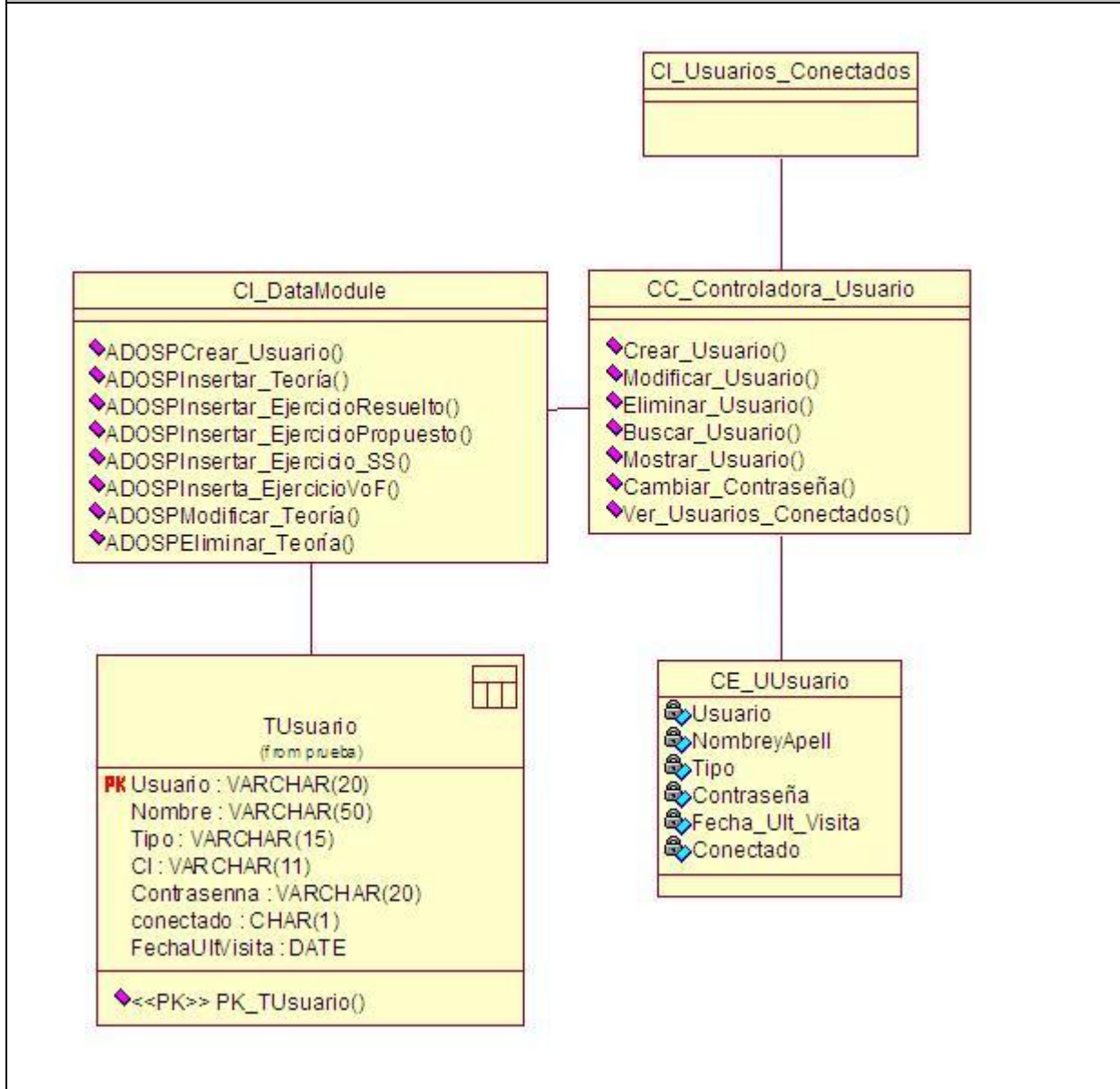
Anexo I.18 Paquete Reporte. Caso de uso Ver Trazas.

Diagrama de clases del diseño



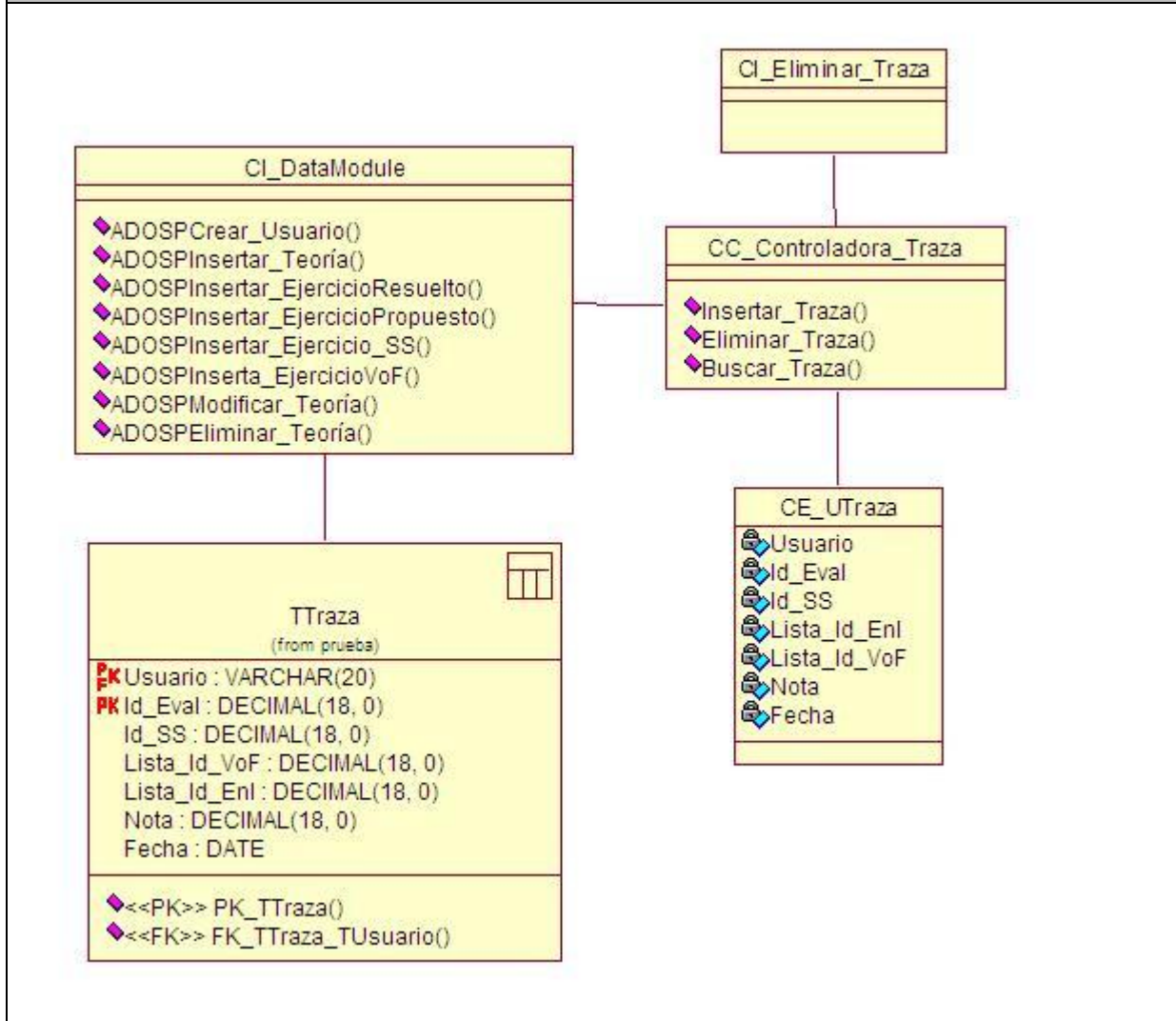
Anexo I.19 Paquete Reporte. Caso de uso Ver Usuarios Conectados.

Diagrama de clases del diseño



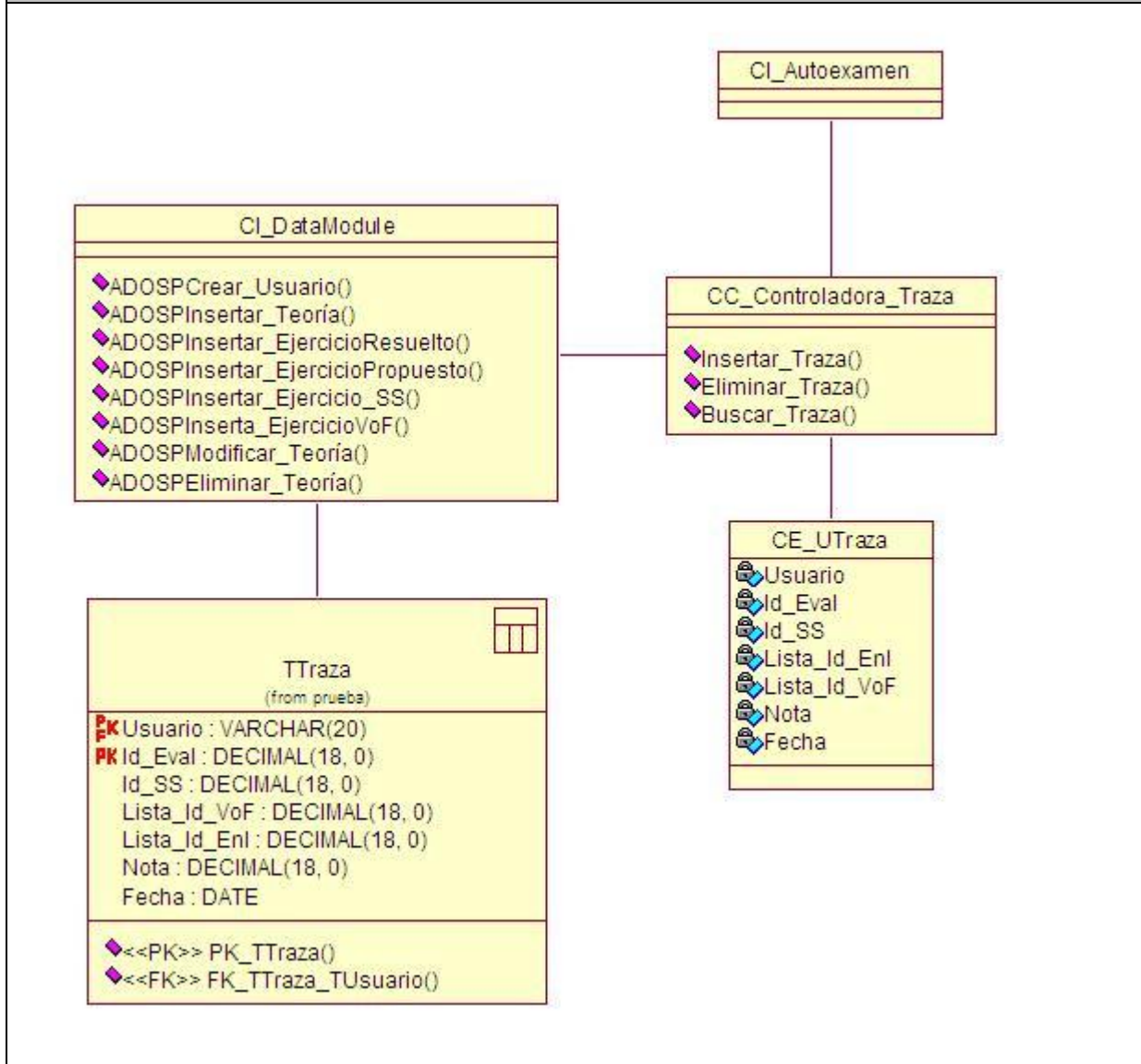
Anexo I.20 Paquete Autoexamen. Caso de uso Eliminar Trazas.

Diagrama de clases del diseño



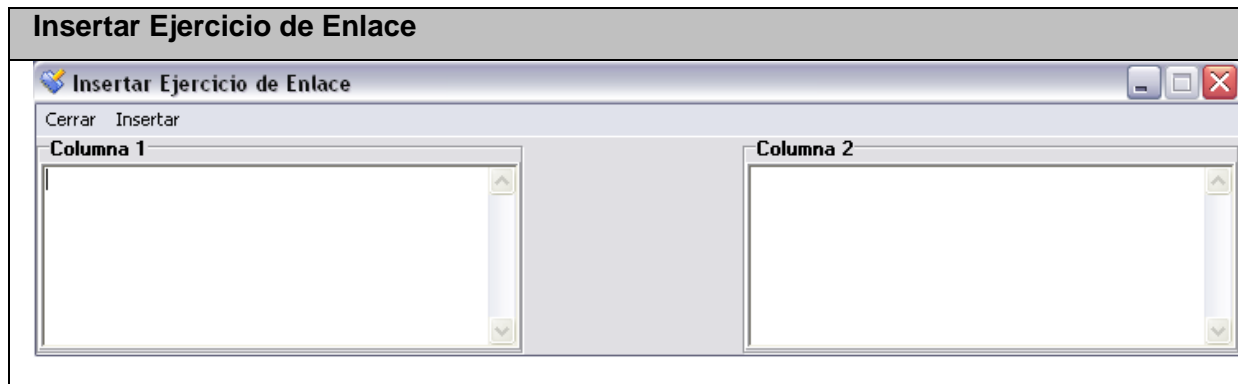
Anexo I.21 Paquete Autoexamen. Caso de uso Resolver Autoexamen.

Diagrama de clases del diseño

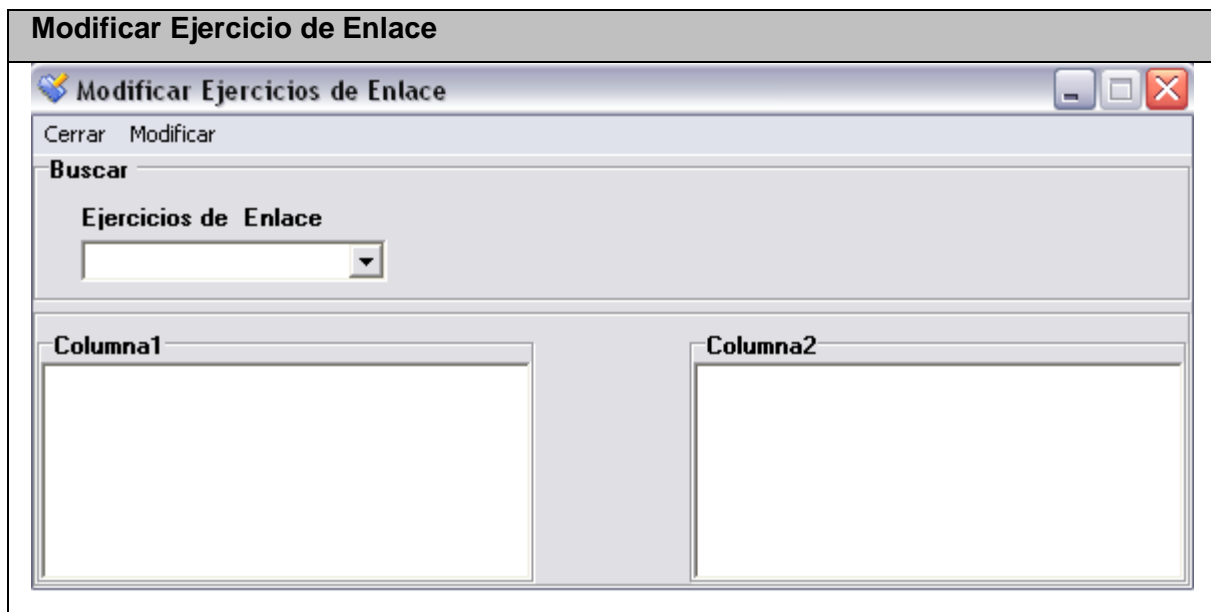


## Anexo II

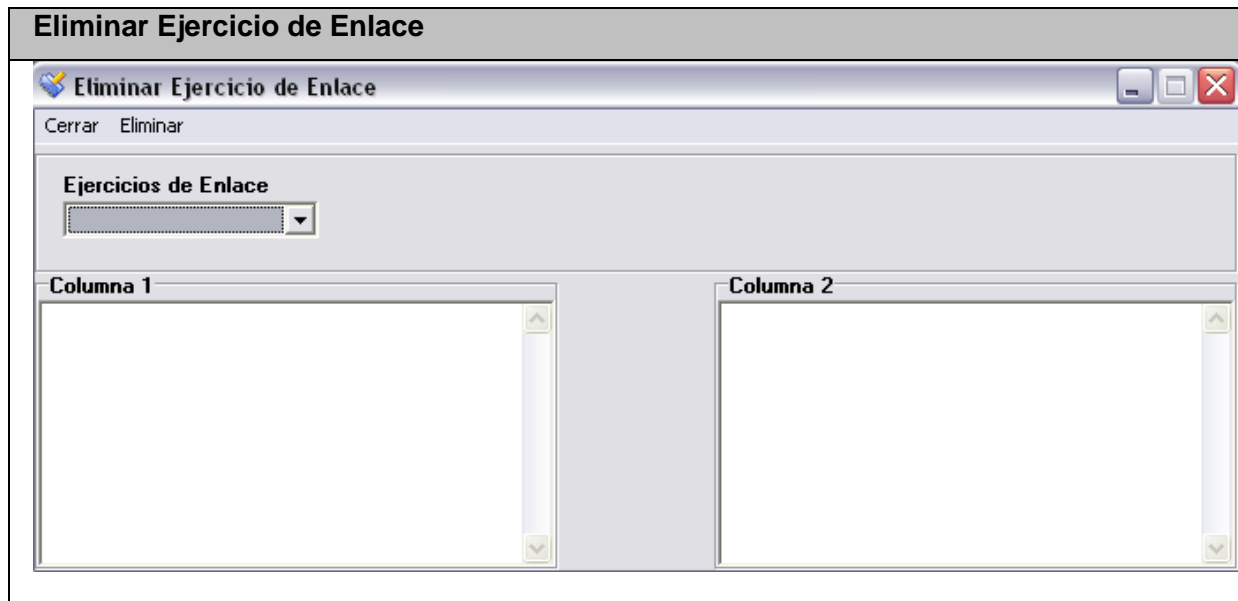
### Anexo II.1 Interfaz del Caso de Uso Gestionar Ejercicio de Enlace.



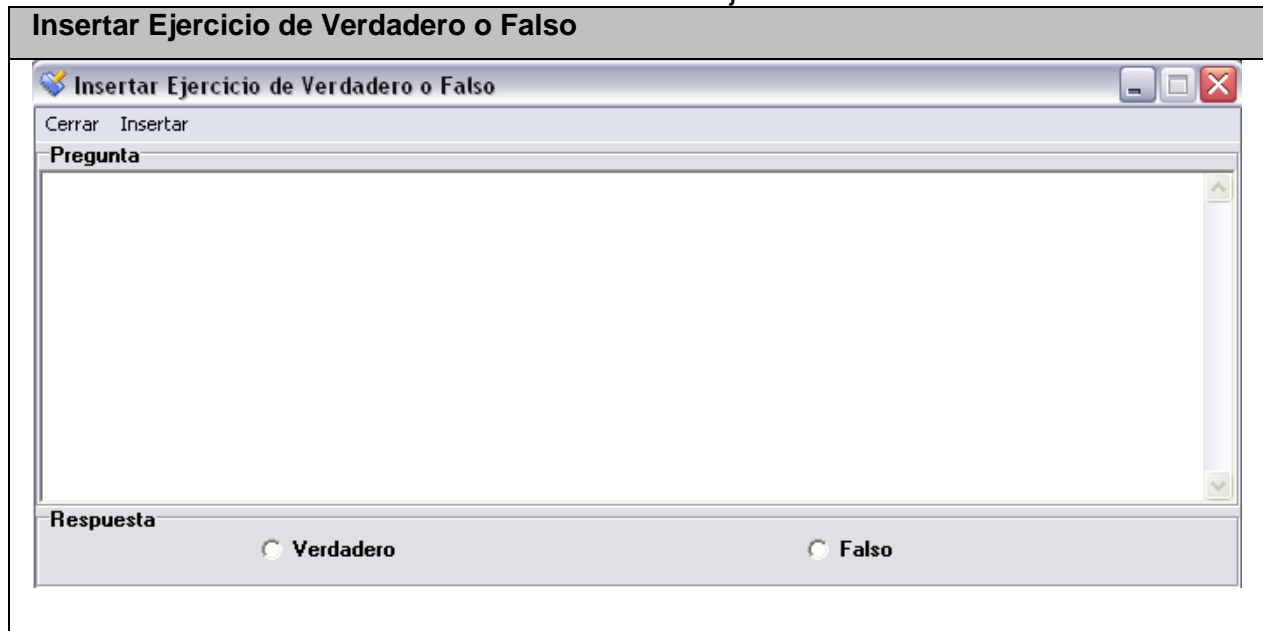
### Anexo II.2 Interfaz del Caso de Uso Gestionar Ejercicio de Enlace.



Anexo II.3 Interfaz del Caso de Uso Gestionar Ejercicio de Enlace.



Anexo II.4 Interfaz del Caso de Uso Gestionar Ejercicio de Verdadero o Falso.





Anexo II.5 Interfaz del Caso de Uso Gestionar Ejercicio de Verdadero o Falso.

**Modificar Ejercicio de Verdadero o Falso**

Cerrar Modificar

**Buscar**

Ejercicios de verdadero o falso

Pregunta

Respuesta

Verdadero  Falso

Anexo II.6 Interfaz del Caso de Uso Gestionar Ejercicio de Verdadero o Falso.

**Eliminar Ejercicio de Verdadero o Falso**

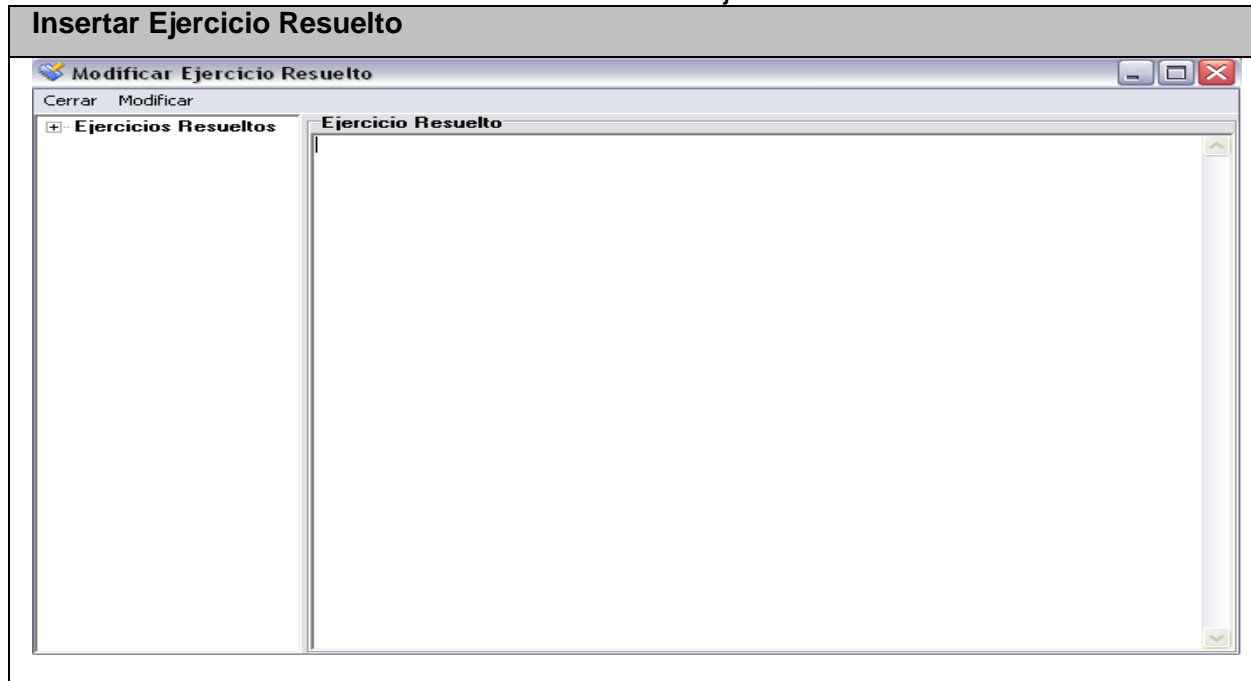
Cerrar Eliminar

Ejercicios de Verdadero o Falso

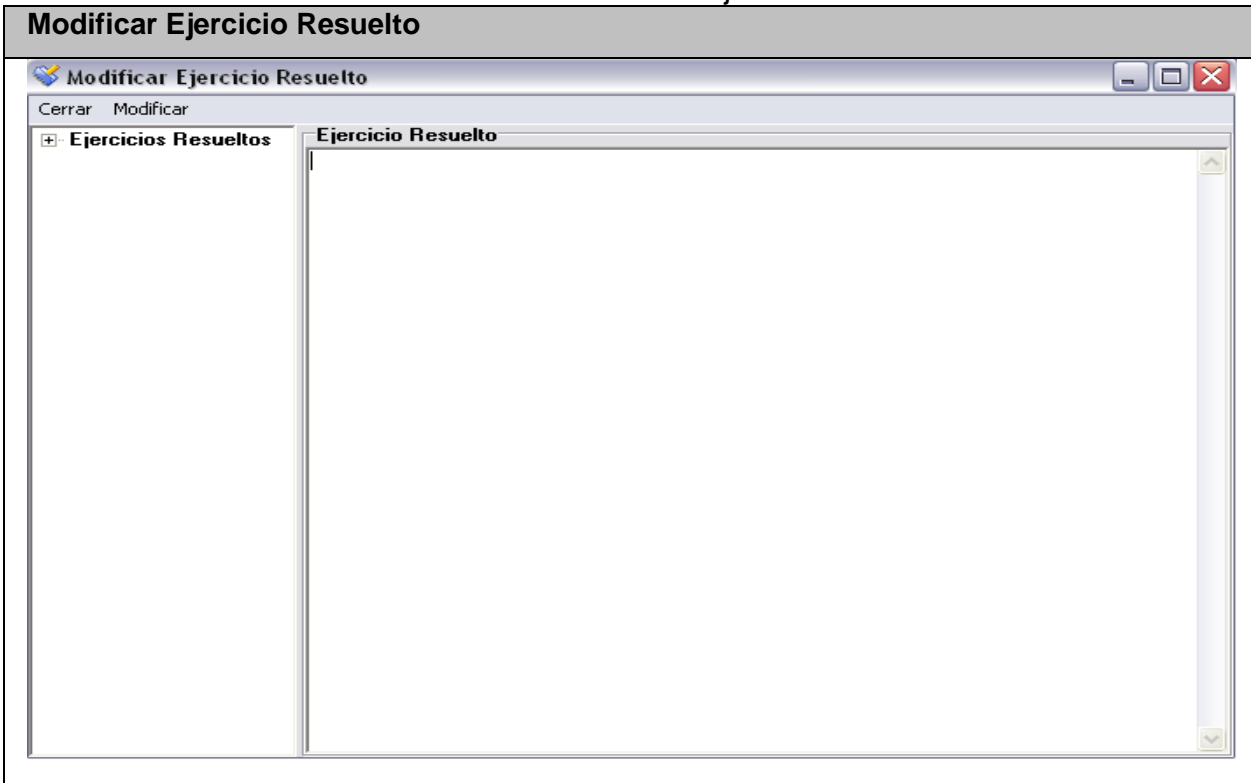
Pregunta

Respuesta

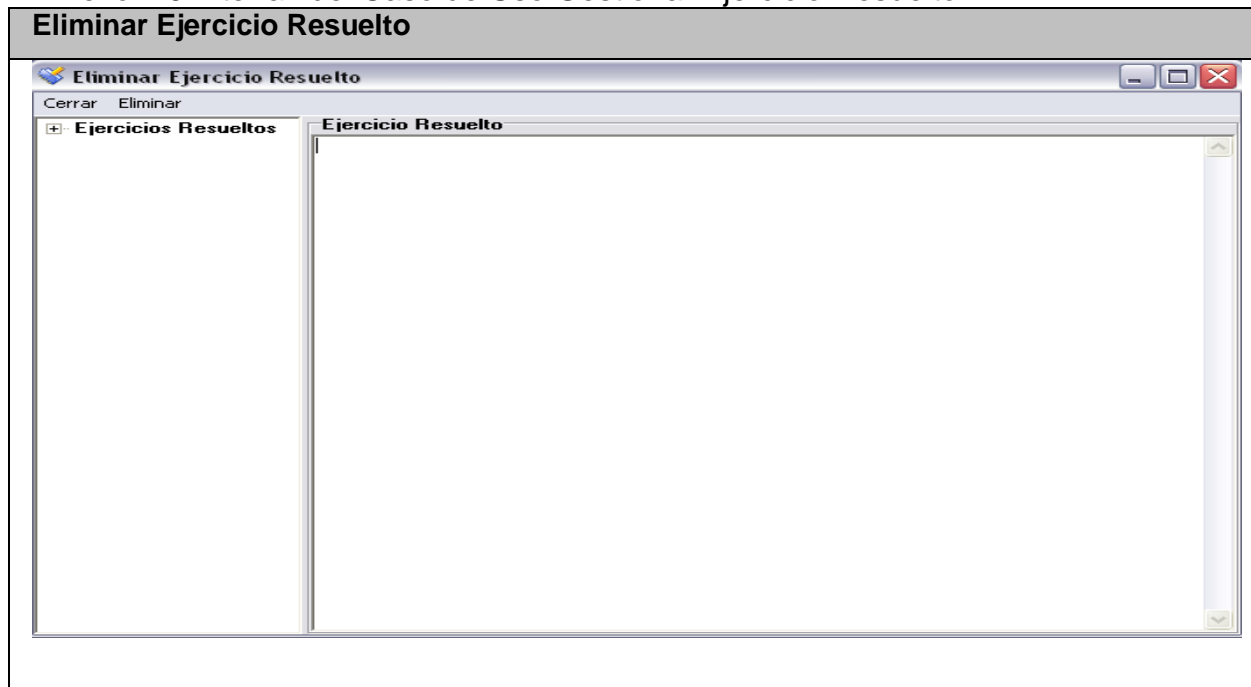
Anexo II.7 Interfaz del Caso de Uso Gestionar Ejercicio Resuelto.



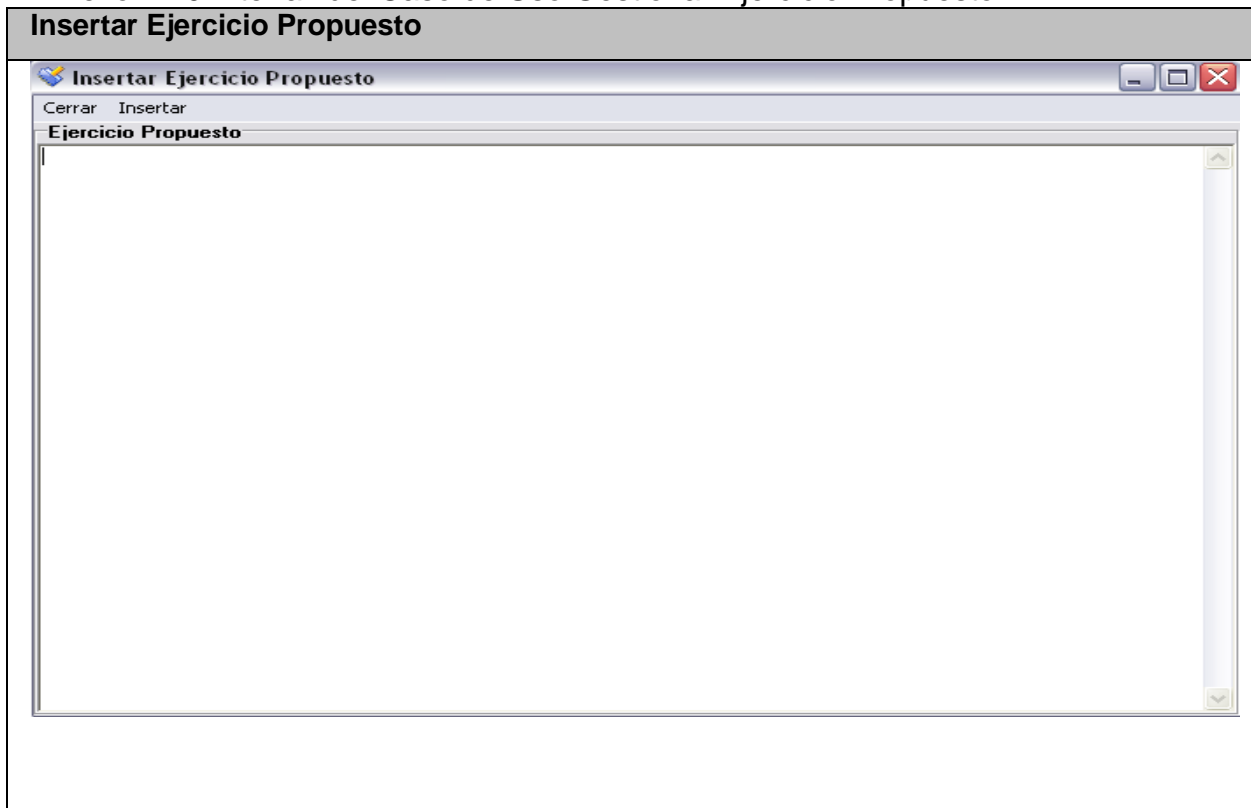
Anexo II.8 Interfaz del Caso de Uso Gestionar Ejercicio Resuelto.



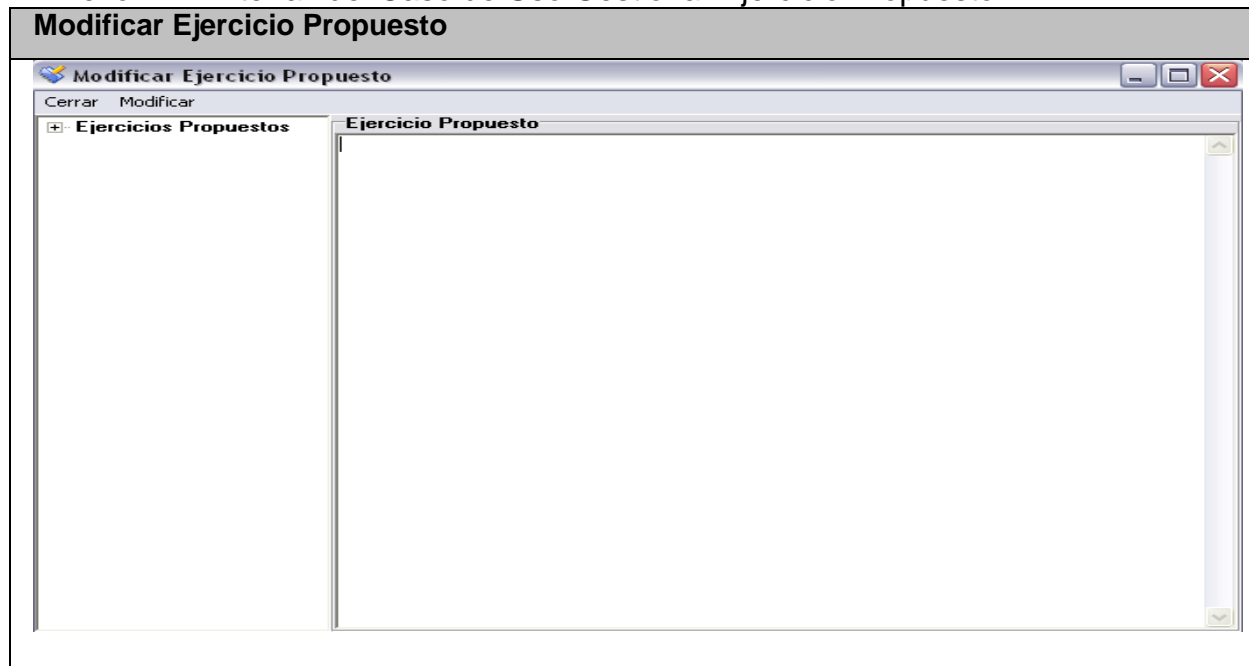
Anexo II.9 Interfaz del Caso de Uso Gestionar Ejercicio Resuelto.



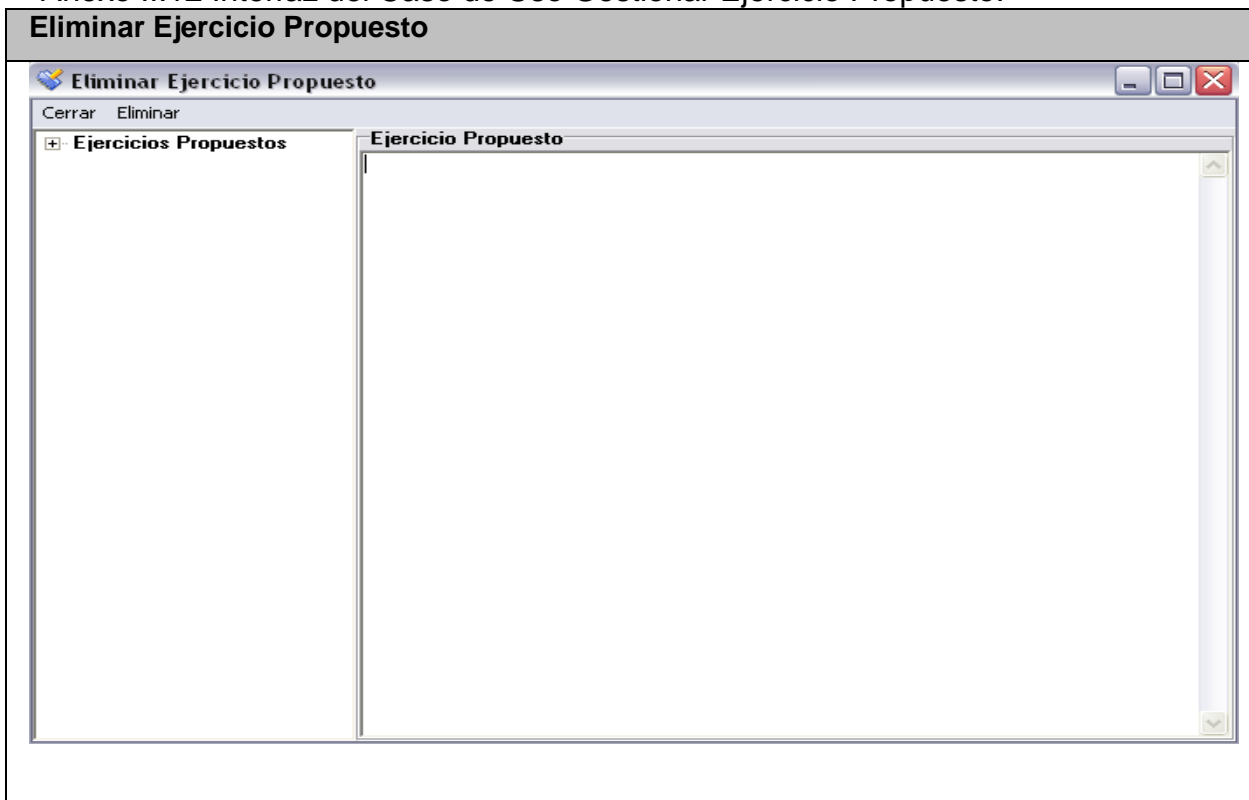
Anexo II.10 Interfaz del Caso de Uso Gestionar Ejercicio Propuesto.



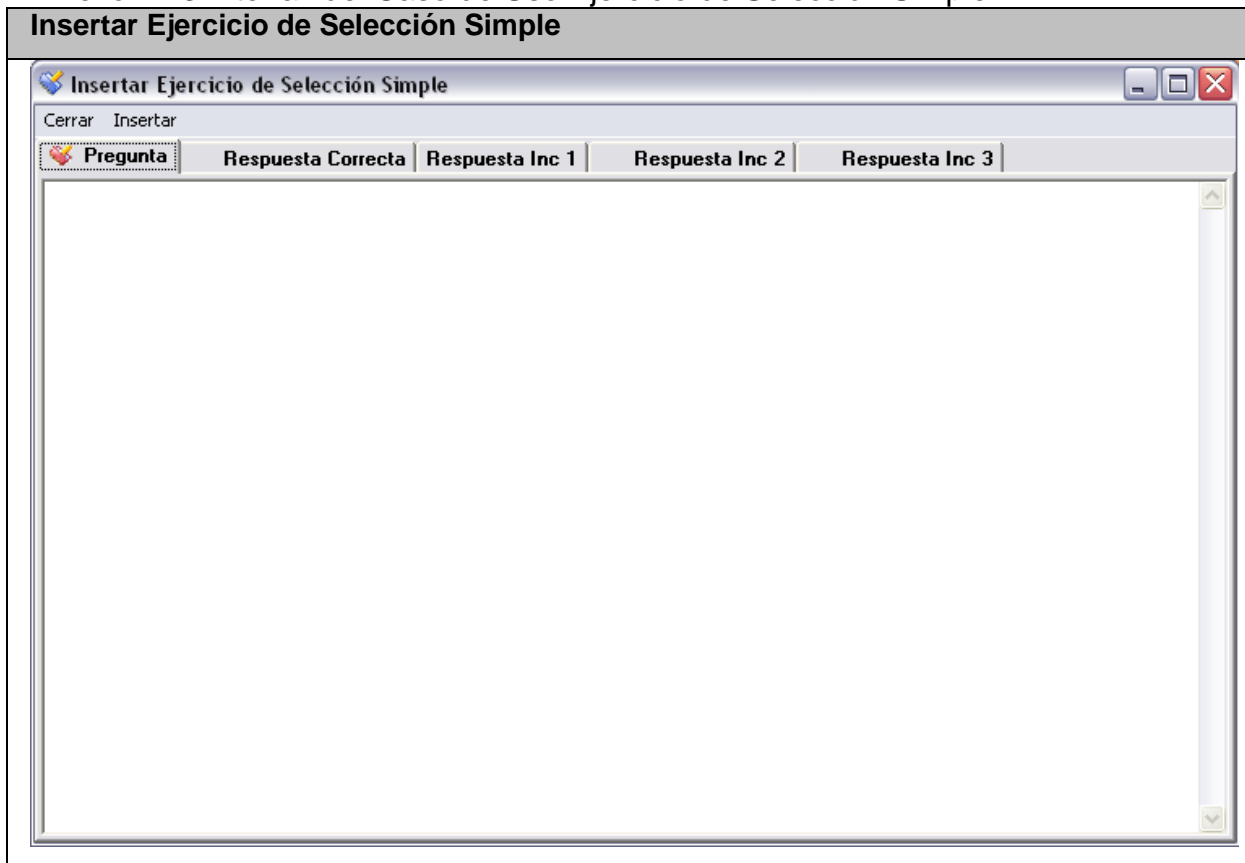
Anexo II.11 Interfaz del Caso de Uso Gestionar Ejercicio Propuesto.



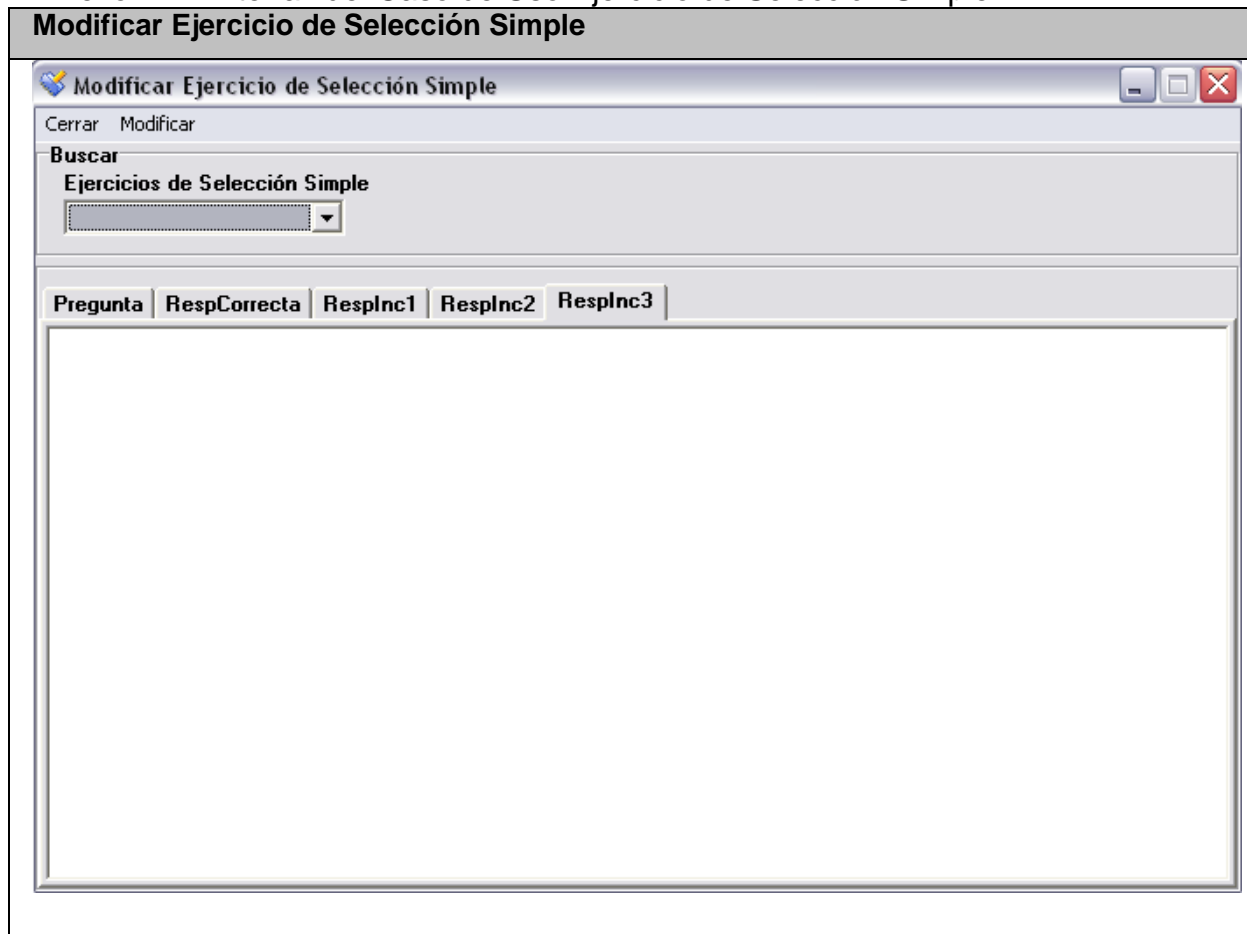
Anexo II.12 Interfaz del Caso de Uso Gestionar Ejercicio Propuesto.



Anexo II.13 Interfaz del Caso de Uso Ejercicio de Selección Simple.



Anexo II.14 Interfaz del Caso de Uso Ejercicio de Selección Simple.



Anexo II.15 Interfaz del Caso de Uso Ejercicio de Selección Simple.

**Eliminar Ejercicio de Selección Simple**

Eliminar Ejercicio de SS

Cerrar Eliminar

**Buscar**

Id Ejercicio de SS

Pregunta Respuesta Correcta Respuesta Incorrecta1 Respuesta Incorrecta 2 Respuesta Incorrecta 3

Pregunta	Respuesta Correcta	Respuesta Incorrecta1	Respuesta Incorrecta 2	Respuesta Incorrecta 3
----------	--------------------	-----------------------	------------------------	------------------------

Anexo II.16 Interfaz del Caso de Uso Gestionar Teoría.

**Insertar Teoría**

Cerrar Insertar

**Título del Libro**

**Autor/Autores**

**Teoría**



Anexo II.17 Interfaz del Caso de Uso Gestionar Teoría.

**Modificar Teoría**

Cerrar Modificar

**Título**

**Autor/Autores**

**Teoría**

Anexo II.18 Interfaz del Caso de Uso Gestionar Teoría.

**Eliminar Teoría**

Cerrar Eliminar

**Título**

**Autor/Autores**

**Teoría**

Anexo II.19 Interfaz del Caso de Uso Gestionar Usuarios.

**Crear Usuario**

Crear Cuenta de Usuario

**Crear usuario**

**Usuario**

**Contraseña**

**Confirmar Contraseña**

**Nombre y Apellidos**

**C.I**

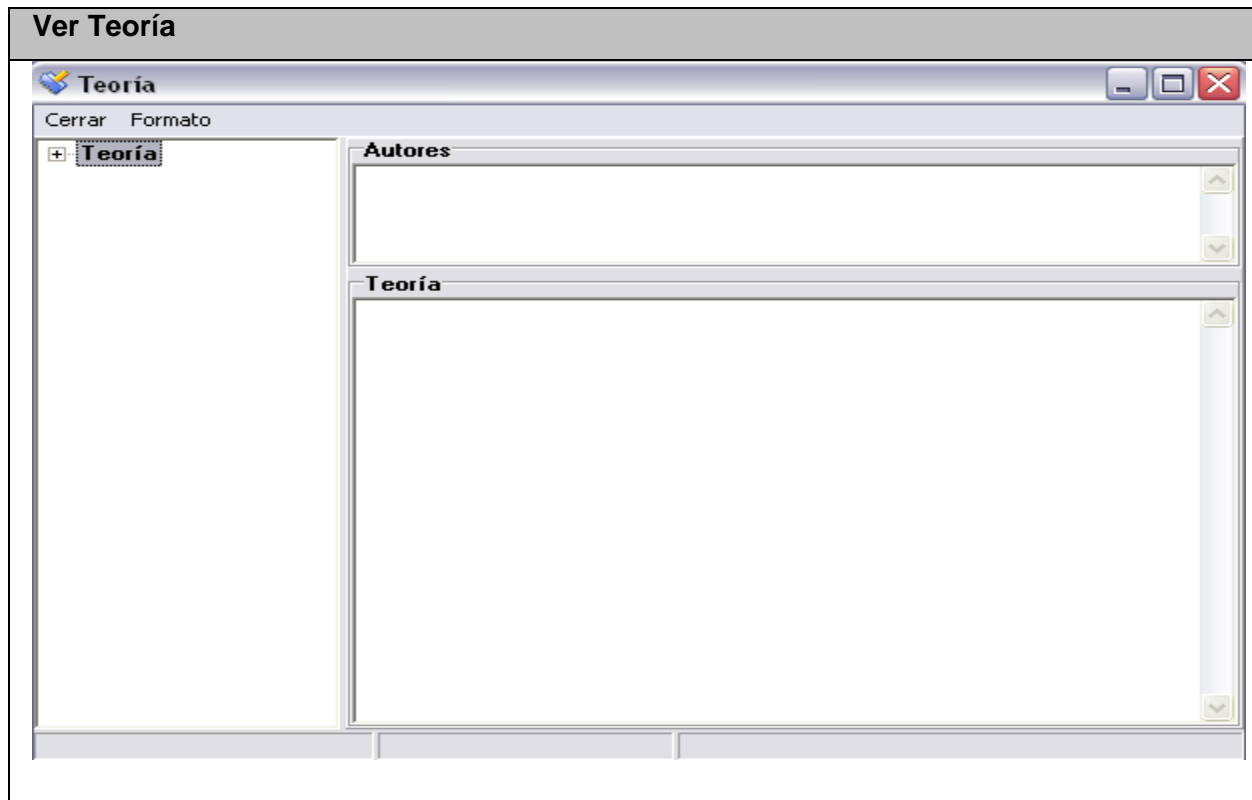
**Crear** **Cancelar**

Anexo II.20 Interfaz del Caso de Uso Gestionar Usuarios.

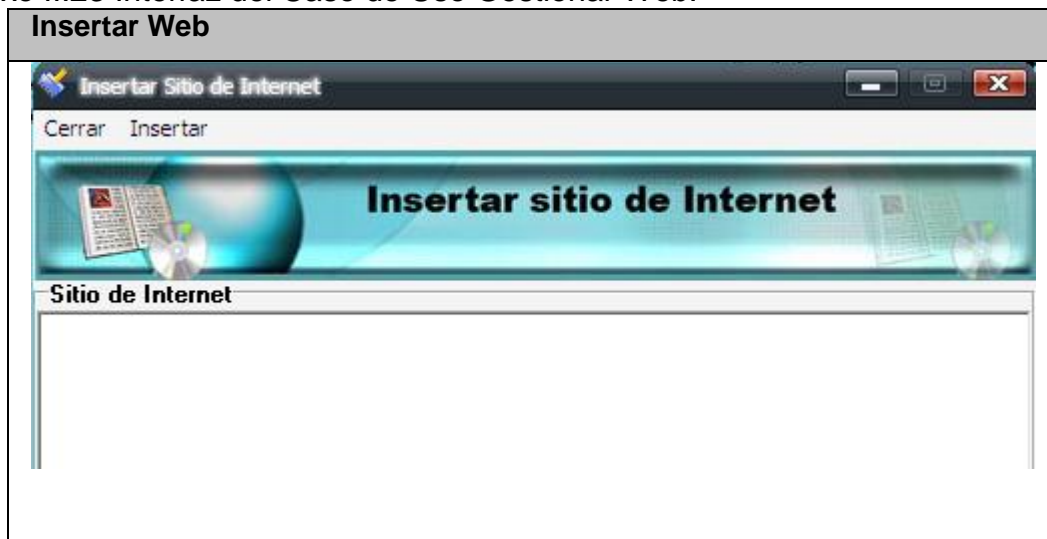
Anexo II.21 Interfaz del Caso de Uso Gestionar Usuarios.

Usuario	NombreApell	Tipo	CI	Contrasen
▶ adrian	Adrian Rodriguez Sanz	Usuario	8609201578	a
iosmel	iosmel manchez mmartin	Usuario	8608271584	io
yoa	Yoandry Borges Castillo	Administrador	8502281474	yoa

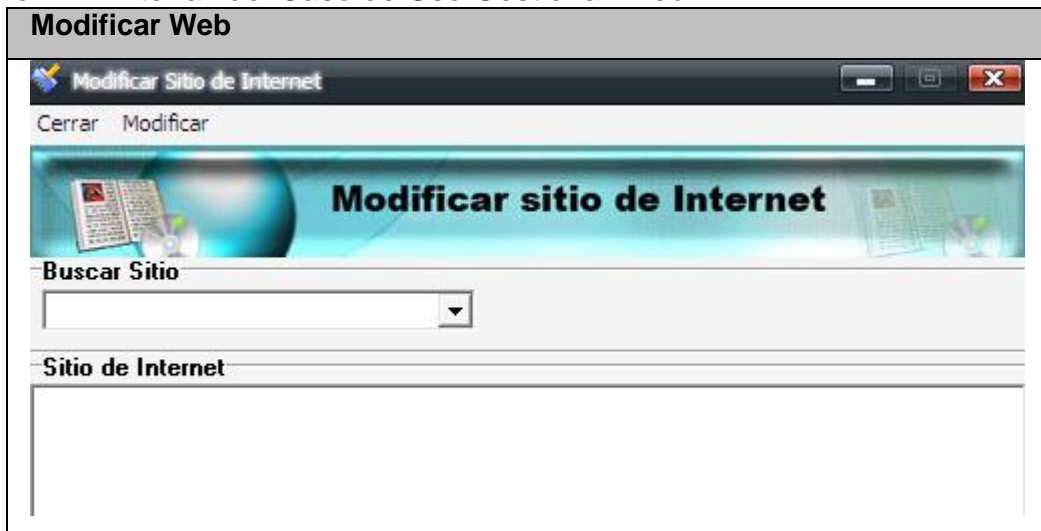
Anexo II.22 Interfaz del Caso de Uso Ver Teoría.



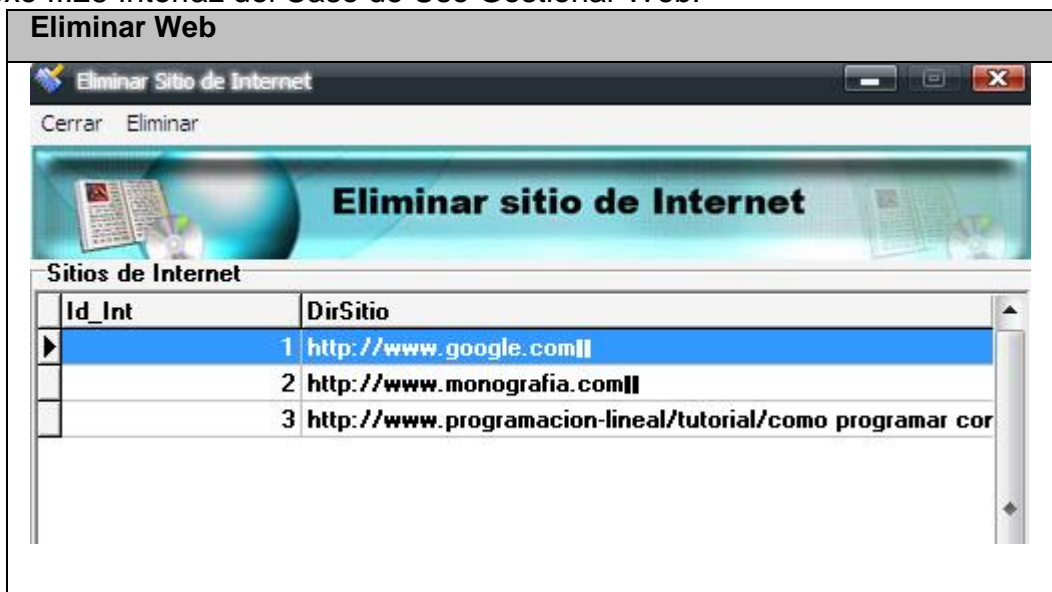
Anexo II.23 Interfaz del Caso de Uso Gestionar Web.



Anexo II.24 Interfaz del Caso de Uso Gestionar Web.



Anexo II.25 Interfaz del Caso de Uso Gestionar Web.



## Anexo II.26 Interfaz del Caso de Uso Reporte de Teoría.

**Reporte de la Teoría**

Reporte de la Teoría Existente

Id Teoría	Título	Autor/Autores
1	Programación Lineal	Colectivo de Autores
2	La Programación Lineal y sus Beneficios	Martha del Río José Miguel Hernández Brito

Universidad de Sancti Spiritus (UNISS)

## Anexo II.27 Interfaz del Caso de Uso Reporte Usuarios del Tutorial.

### Reporte Usuarios del Tutorial

Usuario	Nombre y Apellidos	Tipo	Carnet de Identidad	Fecha Ultima Visita
adrian	Adrian Rodriguez Sanz	Usuario	86092015780	20/05/2010
Iosmel	iosmel manchez mmartin	Usuario	86082715841	05/05/2010
yoa	Yoandry Borges Castillo	Administrador	85022814741	07/06/2010

Universidad de Sancti Spiritus(UNISS)

Anexo II.28 Interfaz del Caso de Uso Reporte Usuarios evaluados por Período.

**Reporte Usuarios evaluados por Período**

Reporte de Usuarios Evaluados por Períodos

Usuario	Nombre y Apellidos	Tipo	Carnet de Identidad	Fecha
adrian	Adrian Rodriguez Sanz	Usuario	86092015780	20/05/2010
yoa	Yoandry Borges Castillo	Administrador	85022814741	21/05/2010

Universidad de Sancti Spiritus (UNISS)



## Anexo II.29 Interfaz del Caso de Uso Reporte Usuarios por Nota.

### Reporte Usuarios por Nota

Usuario	Nombre y Apellidos	Tipo	Carnet de Identidad	Cantidad de Evaluaciones
adrian	Adrian Rodriguez Sanz	Usuario	86092015780	1
yoa	Yoandry Borges Castillo	Administrador	85022814741	1

Universidad de Sancti Spiritus (UNISS)

## Anexo II.30 Interfaz del Caso de Uso Reporte Usuarios que más se evalúan.

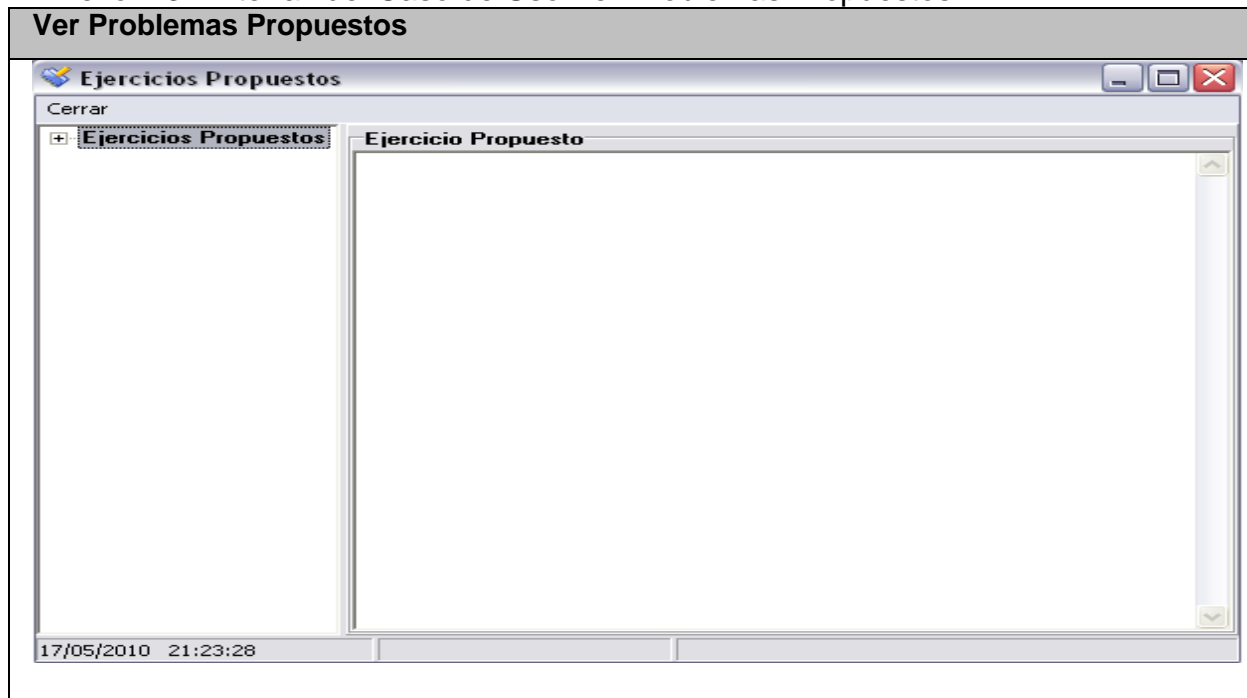
### Reporte Usuarios que más se evalúan

Reporte de los Usuarios que más se Evalúan

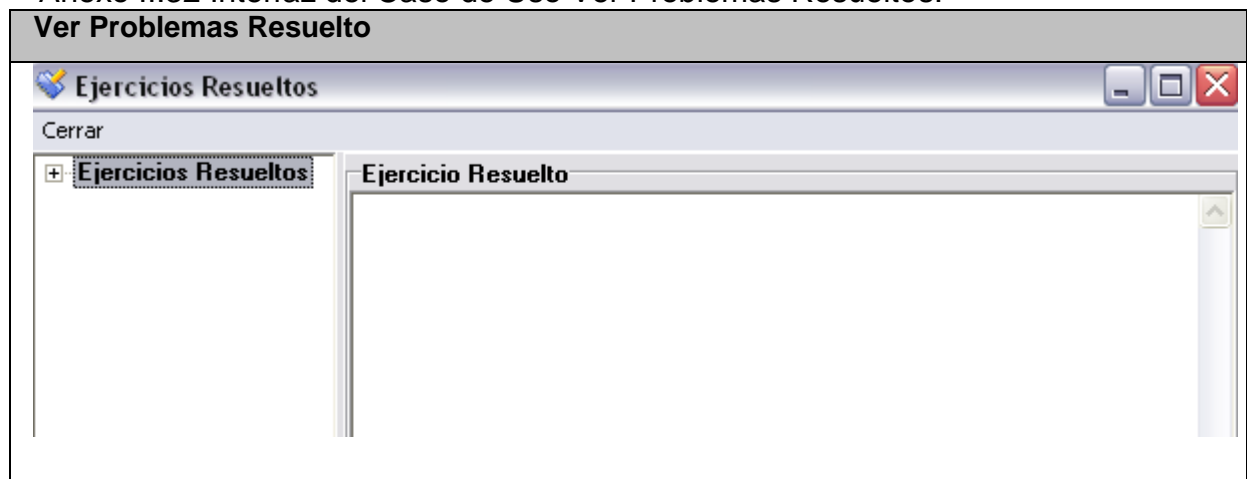
Usuario	Nombre y Apellidos	Tipo	Carnet de Identidad	Cantidad de Evaluaciones
adrian	Adrian Rodriguez Sanz	Usuario	86092015780	1
yoa	Yoandry Borges Castillo	Administrador	85022814741	1

Universidad de Sancti Spiritus (UNISS)

Anexo II.31 Interfaz del Caso de Uso Ver Problemas Propuestos.



Anexo II.32 Interfaz del Caso de Uso Ver Problemas Resueltos.



Anexo II. 33 Interfaz del Caso de Uso Ver Trazas.

**Ver Trazas**

Historial del Usuario

Cerrar Eliminar

**Historial de Usuarios**

Usuario	Id_Evaluacion	Id_SS	Lista_Id_Enl5	Lista_Id_Vo
▶ yoa		1	1 1,2,3,4,5	1,2,3,4,5
yoa		2	2 2,3,4,5,6	2,3,4,5,6
yoa		3	1 5,6,4,8,7	2,6,4,3,1
yoa		4	1 4,5,6,7,8	2,3,6,9,8
yoa		5	1 10,4,2,6,9	7,3,5,8,6
yoa		6	1 8,7,3,5,9	7,6,8,2,9
yoa		7	1 8,7,3,5,9	7,6,8,2,9
yoa		8	2 7,2,5,9,10	4,8,7,3,5
yoa		9	1 1,7,5,9,2	6,3,2,1,9

Anexo II.34 Interfaz del Caso de Uso Ver Usuarios Conectados.

**Ver Usuarios Conectados**

Usuarios Conectados

Cerrar Buscar

Usuario	NombreApell	Tipo	CI	Contrasenna
▶ yoa	Yoandry Borges Castillo	Administrador	8502281474	yoa

Anexo II.35 Interfaz del Caso de Uso Resolver Autoexamen.

**Resolver Autoexamen**

Autoexamen

Cerrar Dar Nota

Pregunta 1 | Pregunta 2 | Pregunta 3

10 es lo mismo que 5+5

prueba verdadera

rojo y azul da lila

33 + 33 es 777

el sol es rojo

The image shows a software window titled 'Resolver Autoexamen'. At the top, there is a menu bar with 'Cerrar' and 'Dar Nota'. Below the menu bar, there are three tabs: 'Pregunta 1', 'Pregunta 2', and 'Pregunta 3'. The main area contains five question entries, each with a text box on the left and a dropdown menu on the right. The questions are: '10 es lo mismo que 5+5', 'prueba verdadera', 'rojo y azul da lila', '33 + 33 es 777', and 'el sol es rojo'. The first dropdown menu is highlighted in blue.

Anexo II.36 Interfaz del Caso de Uso Eliminar Trazas.

Eliminar Trazas

Historial del Usuario

Cerrar Eliminar

### Historial de Usuarios

Usuario	Id_Evaluacion	Id_SS	Lista_Id_EnI5	Lista_Id_Vo
yoa	1	1	1,2,3,4,5	1,2,3,4,5
yoa	2	2	2,3,4,5,6	2,3,4,5,6
yoa	3	1	5,6,4,8,7	2,6,4,3,1
yoa	4	1	4,5,6,7,8	2,3,6,9,8
yoa	5	1	10,4,2,6,9	7,3,5,8,6
yoa	6	1	8,7,3,5,9	7,6,8,2,9
yoa	7	1	8,7,3,5,9	7,6,8,2,9
yoa	8	2	7,2,5,9,10	4,8,7,3,5
yoa	9	1	1,7,5,9,2	6,3,2,1,9

Anexo II.37 Interfaz del Caso de Uso Ver Web.

Ver Web

