

UNIVERSIDAD “ JOSÉ MARTÍ PÉREZ ” DE SANCTI SPÍRITUS



FACULTAD DE INGENIERÍA

INGENIERÍA INFORMÁTICA

TUTORIAL PARA CONTRIBUIR A LA ENSEÑANZA Y APRENDIZAJE DE LA
MODELACIÓN MATEMÁTICA EN EL TEMA PROGRAMACIÓN LINEAL.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA EN INFORMÁTICA

AUTOR

CARLOS JAVIER VÁZQUEZ HERNÁNDEZ

TUTOR

MSC. AURELIO LORENZO HERNÁNDEZ REYES

Sancti Spíritus

Julio de 2010

Pensamiento.

La ciencia más útil es aquella cuyo fruto es el más comunicable.

Leonardo da Vinci

Dedicatoria.

*A mis padres y toda mi familia por
el apoyo y la confianza en estos 5 años,
a mi mujer y mi hijo.*

Carlos

Agradecimientos.

- ❖ A mi mamá por su apoyo y haber confiado en mí en los momentos difíciles en toda la carrera.
- ❖ A mi papá por la educación y servirme de guía tanto en mi vida personal como durante la trayectoria de la carrera.
- ❖ A Edgar mi amigo, mi hermano.
- ❖ A cuatro abuelos que han sido como mis padres para mí a Rebeca, Carlos, Nilma, y Juan que ya no está pero siempre estará en mi corazón.
- ❖ A Yailen mi mujer por su esfuerzo y compartir conmigo las noches de desvelos y ser parte de mí en todo este tiempo.
- ❖ A Wilhem, Alayn, por ser más que amigos.
- ❖ A mi tutor por su paciencia y dedicación.
- ❖ A todos aquellos profesores que de una forma u otra desde que comencé mis estudios hicieron de mí lo que hoy soy.
- ❖ A todos mis tíos, mi hermana, primos.
- ❖ A todos mis compañeros de aula, Yoandry, Adrian, Dayron, raciel, Rafael, Manuel, losmel, Mario, Yanet, Claudia, Yise, Maylin, a todos.
- ❖ A los trabajadores de la empresa eléctrica en especial a Maria Ela, Yusmaikel, Norisbel, Jorgito.
- ❖ A todos los profesores que me han apoyado y ayudado en estos cinco años de carrera.
- ❖ A Yandira por todo su apoyo.

A todos gracias.

Resumen.

Muchas personas clasifican el desarrollo de la programación lineal entre los avances científicos más importantes de mediados del siglo XX. En la actualidad es una herramienta que ha ahorrado grandes sumas de dinero a muchas compañías, incluyendo la economía de países de todo el mundo; su aplicación llega a varios sectores de la sociedad. La programación lineal es una asignatura que se imparte en la carrera de contabilidad y finanzas, y aborda temáticas que resultan complejas. Los medios de enseñanzas son un factor fundamental en cualquier institución educacional, constituyendo una herramienta poderosa para el proceso de enseñanza y aprendizaje. En el presente trabajo se presenta un tutorial que está relacionado con el aprendizaje de tres aspectos fundamentales de la programación lineal: la definición de variables, construcción de la función objetivo y construcción del sistema de restricciones, para lo cual brinda una explicación teórica, ejercicios resueltos, propuestos, autoexámenes, así como libros y sitios Web relacionados con este tema, donde los estudiantes podrán encontrar elementos básicos para su autoaprendizaje. Para el desarrollo del tutorial se tuvo en cuenta el proceso de desarrollo RUP, haciendo uso del lenguaje UML. Se utilizó el object pascal como lenguaje de programación y el Borland Delphi 2006 como herramienta para su implementación, así como SQL Server 2000 como gestor de base de datos. El presente software pretende incentivar y hacer novedosa y atractiva la adquisición de conocimientos necesarios para enfrentar y dar solución a los diferentes problemas de programación lineal.

Abstract.

Lineal programming development is regarded by many among the most important scientific advances of middle twenty century. It has become a prominent tool that has saved a lot of money in many enterprises and has helped to improved the economy of lots of contries worldwide as well. On the other hand it`s application is available for many society sectors. Lineal programming is one of the subjects delivered in accountancy and finances career dealing with many complex topics. Education means are considered a cornerstone in every educacional institution, being a powerfull tool for the teaching and learning process. This work presents a tutorial related to the learning of three basic aspects of lineal programming: definition of variables, construction of objective function and construction of the restrictions system; offering a theoretic explanation, resolved and proposed exersices, autoexaminations as well as books and web sites related to the theme, where the students will be able to find out important information for their autotraining. The process of development RUP was taken into account for the creation of the tutorial, using the UML language. On the other hand object Pascal was used as programming language being Borland Delphi 2006 used as its implementation tool as well as SQL Server 2000 as database manager. The current software pretends to stimulate and make attractive the acquisition of required knowledges to face and solve the different problems of lineal programming.

Índice

Introducción.....	1
Capítulo 1. Fundamentación del tema.....	6
1.1 Introducción.....	6
1.2 Las Tecnologías de la Información y las Comunicaciones (TIC).....	6
1.2.1 Las TIC empleadas a la Educación.....	6
1.2.2 Medios de enseñanza.....	7
1.2.3 La computadora como medio de enseñanza.....	8
1.2.4 Breve historia de la Programación Lineal.....	9
1.2.5 ¿Qué es la Programación Lineal?.....	10
1.2.6 Software Educativo.....	12
1.2.7 Tutoriales.....	13
1.3 El proceso de enseñanza y aprendizaje de la programación lineal.....	13
1.3.1 El proceso de enseñanza de la programación lineal en la UNISS.....	14
1.3.2 Causas que dan origen a la situación problemática y las consecuencias.....	14
1.4 Análisis comparativo de otras soluciones existentes con la propuesta.....	17
1.5 Fundamentación de la metodología utilizada.....	18
1.5.1 Metodología RUP.....	18
1.5.2 Notación UML.....	20
1.6 Tecnologías y herramientas que serán utilizados.....	23
1.6.1 Arquitectura de desarrollo de N Capas.....	24
1.6.2 Microsoft SQL Server 2000.....	26
1.6.3 Borland Delphi 2006.....	29
1.7 Conclusiones.....	30
Capítulo 2. Descripción de la solución propuesta.....	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
2.2.3 Reglas a considerar.....	32
2.3 Requisitos funcionales.....	33
2.4 Requisitos no funcionales.....	36
2.5 Descripción del sistema propuesto.....	39
2.6 Modelo de casos de uso del sistema.....	42
2.6.1 Paquete Administración.....	43
2.6.2 Paquete Gestión.....	43
2.6.3 Paquete Información.....	43
2.6.4 Paquete Evaluación.....	44
2.7 Descripción de los casos de uso del sistema.....	44

2.8 Conclusiones.....	56
Capítulo 3. Construcción de la solución propuesta.....	57
3.1 Introducción.....	57
3.2 Diagrama de clases.	57
3.3 Diseño de la base de datos.	57
3.3.1 Diagrama de clases persistentes.	57
3.3.2 Modelos de datos.	58
3.4 Interfaz de usuario.....	59
3.5 Concepción general de la ayuda.	60
3.6 Tratamiento de errores.	60
3.7 Estándares de codificación.	60
3.8 Diagrama de componentes.....	61
3.9 Modelo de despliegue.	62
3.10 Conclusiones.....	62
Capítulo 4. Estudio de la factibilidad	63
4.1 Introducción.....	63
4.2 Planificación.....	63
4.3 Costos.	67
4.4 Beneficios tangibles e intangibles.	71
4.5 Análisis de costos y beneficios.	71
4.6 Conclusiones.....	72
Conclusiones.....	73
Recomendaciones.....	74
Bibliografía.....	75
Anexos	81

Índice de Tablas.

TABLE 1 ANÁLISIS DE LOS TRABAJOS DE DIPLOMA EN LA CARRERA DE CONTABILIDAD Y FINANZAS	15
TABLE 2 ACTORES DEL SISTEMA	41
TABLE 3 ENTRADAS EXTERNAS.....	63
TABLE 4 SALIDAS EXTERNAS.....	64
TABLE 5 FICHEROS INTERNOS.....	65
TABLE 6 PUNTOS DE FUNCIÓN SIN AJUSTAR.....	66
TABLE 7 INSTRUCCIONES FUENTES	67
TABLE 8 MULTIPLICADORES DE ESFUERZO	67
TABLE 9 DEFINICIÓN DE LOS VALORES DE LOS FACTORES DE ESCALA (SFI).....	68
TABLE 10 CÁLCULO DEL ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.....	70

Índice de Figuras.

FIGURA 1 CICLOS DE DESARROLLO.....	19
FIGURA 2 MODELO DE DISEÑO EN TRES CAPAS	25
FIGURA 3 MODELO DE OBJETO.....	32
FIGURA 4 DIAGRAMA DE PAQUETE.....	42
FIGURA 5 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE ADMINISTRACIÓN	43
FIGURA 6 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE GESTIÓN	43
FIGURA 7DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE INFORMACIÓN.....	44
FIGURA 8 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE EVALUACIÓN	44
FIGURA 9 DIAGRAMA DE CLASES PERSISTENTES.	58
FIGURA 10 MODELOS DE DATOS.....	59
FIGURA 11 DIAGRAMA DE COMPONENTES.	61
FIGURA 12 DIAGRAMA DE DESPLIEGUE.....	62

Introducción.

Con el pasar de los años las tecnologías de la investigación y las comunicaciones (TIC), se han ido desarrollando y perfeccionando. Cuba a pesar de encontrarse potencialmente bloqueado por el gobierno de los Estados Unidos, hace todo lo que está a su alcance para estar a la altura de los países desarrollados en materia de las (TIC), llevando a cabo la informatización de todos los sectores del país. El Ministerio de la Educación, quien es el encargado de la enseñanza y formación de valores en los jóvenes también está presente en este proceso por todas las ventajas que trae consigo el mismo.

Las TIC se presentan cada vez más como una necesidad en el contexto de sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de una educación de alto nivel constantemente actualizada se convierten en una exigencia permanente.(Rosario, 2005)

En la Educación, donde muchas asignaturas abordan temas relativamente complejos, se considera de vital importancia la búsqueda de soluciones para lograr que el estudiantado adquiera los conocimientos esperados.

La aplicación de los medios de enseñanza permite, entre otras cosas: mejorar la calidad de las clases y adquirir los conocimientos de una manera más fácil. Se demostró en muchas ocasiones por la ciencia pedagógica, que la computación favorece la asimilación de los contenidos. (García Suares & López Delgado, 2007)

La creación de software educativo se considera un poderoso medio de enseñanza teniendo en cuenta que dentro del aprendizaje humano la mayor interrelación con el mundo exterior se da a través del órgano visual, es decir, del mecanismo sensorial de la vista. Por eso el empleo de los medios de enseñanza y en especial los medios visuales, facilita el óptimo aprovechamiento de nuestros mecanismos sensoriales. Los resultados obtenidos (que solamente deben utilizarse como datos probables aproximados) de los experimentos realizados en diversas partes del mundo, coinciden en destacar que el conocimiento del mundo exterior, es logrado por el hombre en una proporción aproximada:

Mediante la vista 83%

Mediante el oído 11%

Mediante el olfato 3.5%

Mediante el tacto el 1.5%

Mediante el gusto el 1%

Es por ello que las aplicaciones que se realizan en centros educacionales requieren de un estudio minucioso para que desde el punto de vista pedagógico cumpla con lo esperado. Por este motivo no sólo se pretende que el producto final tenga la calidad esperada como aplicación; sino que tenga aún más como medio de enseñanza. (Hombre y naturaleza)

La Universidad José Martí Pérez de la provincia de Sancti-Spíritus (UNISS); cuenta con una facultad de contabilidad y finanzas, en la cual la temática conocida como Programación Lineal juega un papel muy importante para los estudiantes de tercer año, por ser una materia a vencer para poder pasar de semestre y una herramienta muy poderosa para lograr alcanzar mejores resultados en su vida profesional. Son conocidos los problemas que existen para impartir esta temática; así como los que afrontan los estudiantes para asimilar sus contenidos. En la actualidad las clases se imparten por los métodos tradicionales, con medios de enseñanzas tales como: una pizarra, una tiza y alguna que otras láminas. Los estudiantes no muestran mucho interés por la materia y en algunos de los casos la consideran difícil. Los resultados aproximados en la provincia de Sancti-Spíritus, por lo general; es que solo 1 estudiante de cada 100 logra poner en práctica los conocimientos de programación lineal en su vida profesional y dada la necesidad que tiene el país de tomar decisiones correctas a la hora de desarrollar cualquier proyecto y siendo conocido el papel tan importante que juega la programación lineal en la toma de decisiones, se convierte en un deber para La UNISS; lograr que los estudiantes de contabilidad y finanzas sean futuros profesionales capacitados para mejorar los resultados económicos de la institución donde laboren. La bibliografía existente en estos momentos no deja de ser buena, pero le falta enriquecerse muchísimo, por lo general el estudiantado para ampliar sus conocimientos una vez concluidas las clases sólo cuenta con el libro de texto y algunos que otros artículos publicados en las diferentes bibliotecas de la provincia; pero el hecho de tener que trasladarse a diferentes lugares en búsqueda de información hace que muchos pierdan el interés y sólo se apoyen en el libro de texto básico; es por ello que otros de los problemas que existe es; que no se cuenta con un sitio donde el estudiante pueda encontrar todo lo que busca y desea conocer. Todo lo expresado anteriormente conforma la **situación problemática** de la presente investigación.

Problema a resolver: ¿Cómo contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS?

Para ello tendremos como **Objeto de estudio:** El proceso de enseñanza y aprendizaje de la programación lineal.

El **campo de acción** es: El proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS.

Para dar solución al problema planteado se ha propuesto como **objetivo general:** diseñar e implementar un tutorial que contribuya al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS.

Para cumplir el objetivo general se han trazado las siguientes **preguntas científicas:**

- ¿Cuáles son los fundamentos teóricos y metodológicos que sustentan el uso de un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS?
- ¿Cómo diseñar un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS?
- ¿Cómo realizar un análisis de los costos y beneficios del tutorial?
- ¿Cómo implementar un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS?

Para dar solución a las preguntas científicas se trazaron las siguientes **tareas de investigación:**

- Determinación de los fundamentos teóricos y metodológicos que sustentan el uso de un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática

del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS.

- Diseño de un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS.
- Estudio de los costos y beneficios del tutorial.
- Implementación de un tutorial para contribuir al proceso de enseñanza y aprendizaje de la formulación matemática del tema programación lineal en los estudiantes de tercer año de contabilidad y finanzas en la UNISS.

El presente trabajo se estructura de la siguiente forma: introducción, cuatro capítulos, conclusiones, recomendaciones y bibliografía.

Capítulo 1. Fundamentación del tema:

En el capítulo se abordan los principales conceptos asociados al dominio del problema, los objetivos estratégicos de la organización, flujo actual de los procesos involucrados en el campo de acción, un análisis comparativo de otras soluciones existentes con la propuesta, así como una descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta. Fundamentación de las metodologías utilizadas, del lenguaje y gestores de bases de datos.

Capítulo 2. Descripción de la solución propuesta:

Este capítulo abordará los principales conceptos asociados al dominio del problema; así como las reglas a considerar, requisitos funcionales y no funcionales, la concepción general del sistema, modelos de casos de uso del sistema con la descripción de cada uno de los casos de uso.

Capítulo 3. Construcción de la solución propuesta:

En este capítulo se muestra la construcción de la solución propuesta, mediante los diferentes diagramas tales como: Diagrama de Clases, Diagrama de Clases Persistentes, Modelo de Datos. También tratará sobre la interfaz de usuario, concepción general de la ayuda, tratamiento de errores y estándares de codificación. Se modelará además el Modelo de Despliegue y el Diagrama de Componentes.

Capítulo 4. Estudio de factibilidad:

Está dedicado al estudio de factibilidad, es decir, la estimación de personal, tiempo, costo, beneficios y análisis de costo-beneficios del proyecto en cuestión.

Capítulo 1. Fundamentación del tema.

1.1 Introducción.

En el presente capítulo se abordarán los principales conceptos asociados al dominio del problema. También se incluyen los objetivos estratégicos de la organización, flujo actual de los procesos involucrados en el campo de acción, un análisis de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. También se hará un análisis comparativo de otras soluciones existentes con la propuesta, así como una descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta. Fundamentación de las metodologías utilizadas, del lenguaje y gestores de bases de datos.

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

Se denominan Tecnologías de la Información y las Comunicaciones al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TIC incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual. (Rosario, 2005)

Expresado de otra manera las Tecnologías de la Información y la Comunicación (TIC) constituyen una serie de técnicas de aplicaciones y equipos, redes de computación, bases de datos y otros medios puestos al servicio de los sistemas de información basados en computadoras. Asociado a las nuevas tecnologías, aparecen nuevos códigos y lenguajes, que permiten nuevas realidades expresivas, como es el caso de las aplicaciones o programas como también suelen ser llamados. El usuario extrae información, participa activamente en la construcción de su conocimiento sobre un dominio concreto.

1.2.1 Las TIC empleadas a la Educación.

Las TIC constituyen una herramienta cada vez más poderosa e indispensable en las instituciones educacionales, donde pueden emplearse como: fuente de información, guía para el proceso de aprendizaje de los estudiantes, controladora de los contenidos a enseñar, ejercitar habilidades, entrenar lo aprendido, motivar el estudio, entre otras muchas aplicaciones.

En la actualidad se han desarrollado muchas aplicaciones con diferentes objetivos y funcionalidades pedagógicas, es por ello que las tecnologías de la información y las comunicaciones han incrementado de modo considerable su presencia como medio de enseñanza a disposición de los docentes y educandos. (Jímenes Valle, 2008)

Algunos de los papeles que juegan las TIC en la educación son:

- Auxiliar a los estudiantes a escribir y calcular.
- Guiar a los estudiantes.
- Facilitar la adquisición de los recursos educativos desde ubicaciones remotas.
- Ayudar a los profesores en la evaluación del progreso del estudiante y la administración de la instrucción.
- Fomentar la colaboración entre estudiantes y profesores.

A finales del siglo pasado, la casi totalidad de los campos profesionales han visto incrementado su potencial de desarrollo con la incorporación de las nuevas tecnologías y ello ha motivado un cambio sustancial en el modo de ejercer las funciones específicas en cada uno de estos campos. Esta situación lleva como contrapartida la demanda implícita de nuevos roles para el desarrollo profesional. En el caso concreto de la educación no hay excepción, a lo largo de estos últimos años, se enfatiza mucho en relación con el cambio en el perfil del maestro como consecuencia de la integración de las nuevas tecnologías en el ámbito escolar. (Conociendo algo de software educativo, 2009)

Las TIC son empleadas en los diversos medios de transmisión de la información, insertándose así en diferentes esferas de la vida cotidiana, y entre ellas, en el proceso docente – educativo. En las condiciones de hoy en día es posible disponer de herramientas de apoyo al proceso educativo que se hallen más cercanas a la manera de percibir y entender el mundo, tanto por parte de los niños, como de los jóvenes y adultos, de una manera dinámica, llena de estímulos paralelos, capacitados para el cambio constante e integrado. (Gómez Martínez, 2009)

1.2.2 Medios de enseñanza.

Se considera medio de enseñanza a todos los componentes del proceso docente que actúan como soporte material de los métodos (instructivos o educativos), con el propósito de lograr los objetivos planteados. (Almeida Campos, 2009)

Los medios, materiales o recursos de enseñanza son uno de los componentes relevantes y presentes en cualquier proceso dirigido a provocar aprendizaje. Nuestra sociedad de finales del siglo XX está asistiendo al reto de tecnologizar, y por tanto, artificializar los procesos de comunicación entre los seres humanos. La TV., el casete, el ordenador, el fax, el vídeo, los libros, discos, diapositivas, etc., son parte de nuestra vida cotidiana. Con ellos nos informamos, nos ponemos en relación con el resto de los humanos, expresamos nuestras ideas y sentimientos,..., en definitiva gran parte de nuestra existencia es lo que con los medios hagamos. (Batista Pacheco, 2008)

La escuela, como institución integrada y parte constitutiva del entramado social, no es ajena al desarrollo tecnológico.

Aunque siempre ha estado rezagada en la introducción de los nuevos medios en relación a la evolución del resto de la sociedad, inevitablemente los procesos de enseñanza tienden cada vez a emplear distintos tipos de tecnologías. La propia sociedad así lo demanda. (Soler Castillo, 2008).

El concepto de medios de enseñanza y el papel que juega dentro del proceso docente educativo se ha definido por pedagogos de todas partes del mundo. En un principio fueron denominados “auxiliares” que empleaban los maestros para sus clases, lo cual era algo erróneo ya que son componentes del proceso sistémico del cual no se puede separar.

Se pudiera definir que los medios de enseñanza son diferentes imágenes y representaciones de objetos y fenómenos, que se crean para la docencia. También objetos naturales e industriales, que contienen información y se emplean como fuente del conocimiento.

1.2.3 La computadora como medio de enseñanza.

Desde que el hombre comenzó a hablar, la palabra fue el primer y más importante medio de comunicación; ya antes de esta renovación racional, el gesto, la acción, los sonidos aún pobremente articulados o los que producían con piedras y pieles estiradas, fueron los primeros

medios de que se valieron los hombres para comunicarse entre sí y hacer llegar a las nuevas generaciones los elementos necesarios para vivir y actuar sobre el mundo circundante. (Almeida Campos, 2009)

El desarrollo de la ciencia y la técnica trajo asociado el uso de la computadora como medio de enseñanza ya que la misma constituía una herramienta poderosa para la enseñanza y el aprendizaje dada las cualidades que poseía. Este medio permite que aquellas personas interesadas en aprender contaran con recursos como son imágenes, videos, libros, explicaciones habladas, así como programas encaminados a la educación. Es conocido y constituye un hecho para el mundo que una vez empleada la computadora como medio de enseñanza se lograron alcanzar mejores resultados en la educación.

El empleo de las tecnologías de la información y las comunicaciones, son una realidad palpable, donde su uso se vuelve cada vez más necesario. La computadora en la enseñanza puede ser utilizada en el proceso de enseñanza aprendizaje de formas tales como:

- Objeto de estudio.
- Medio de enseñanza.
- Herramienta de trabajo

"Hablar de computación, es hablar de un tema apasionante en todos los sentidos, nos hace soñar sobre el futuro, nos hace discutir sobre las tecnologías apropiadas y sus costos, las políticas para desarrollar una industria, institución y un país. Pero fundamentalmente hablar de computación o informática es hablar de la necesidad de recursos humanos capacitados, de los cambios en la forma de trabajar y los nuevos empleos, de las nuevas posibilidades de desarrollo individual y hasta de aprendizaje con la inserción de la computadora; hablar de computación es hablar de educación. (Toledo Osorio, 2009)

1.2.4 Breve historia de la Programación Lineal.

En los siglos XVII y XVIII, grandes matemáticos, como Newton, Leibnitz, Bernoulli y, sobre todo, Lagrange, se ocuparon de lograr máximos y mínimos condicionados de determinadas funciones. Luego, el matemático francés Jean Baptiste-Joseph Fourier (1768-1830) fue el

primero en intuir, no de forma precisa, los métodos de lo que en la actualidad conocemos con el nombre de programación lineal y la potencialidad que de ellos se deriva.

Si exceptuamos al matemático Gaspar Monge (1746-1818), quien en 1776 se interesó por problemas de este género, debemos remontarnos al año 1939 para encontrar nuevos estudios relacionados con los métodos de la actual programación lineal. En ese año, el matemático ruso Leonid Vitalevich Kantorovitch publica una extensa monografía titulada Métodos matemáticos de organización y planificación de la producción en la que por primera vez se hace corresponder a una extensa gama de problemas una teoría matemática precisa y bien definida, llamada hoy en día programación lineal.

En 1941-1942 se formula por primera vez el problema de transporte, estudiado independientemente por Koopmans y por Kantorovitch, razón por la cual se suele conocer con el nombre de problema de Koopmans-Kantorovitch.

En los 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.

1.2.5 ¿Qué es la Programación Lineal?

La programación lineal es un conjunto de técnicas racionales de análisis y de resolución de problemas que tiene por objeto ayudar a los responsables en las decisiones sobre asuntos en los que interviene un gran número de variables.

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”.

El procedimiento es iterativo, pues mejora los resultados de la función objetivo en cada etapa hasta alcanzar la solución buscada. Ésta se encuentra en un vértice del que no parta ninguna arista a lo largo de la cual la función objetivo aumente.

Aspectos fundamentales de la programación lineal.

Se denominan problemas de optimización aquellos tipos de problemas en los cuales se busca el máximo o mínimo de una función con un número determinado de variables, estando los valores de las mismas sujetos a ciertas limitaciones. (Felipe P. , Aspectos fundamentales de la programación lineal, 1987)

✚ Construcción del modelo de programación lineal.

El procedimiento para la construcción del modelo puede resumirse de la forma siguiente:

Paso 1 - Definición de las variables de decisión.

Paso 2 - Construcción del sistema de restricciones.

Paso 3 - Construcción de la función objetivo. (Felipe P. , Construcción de la función objetivo, 1987)

✚ Definición de las variables de decisión.

La definición de las variables de decisión es el primer paso en la construcción del modelo de programación lineal. Cada variable de decisión se identifica con cada una de las actividades en que se descompone el problema que se estudia.

El nombre de variables de decisión que se da a las actividades responde al hecho de que son variables sobre las cuales se puede tomar una decisión directamente para resolver el problema que nos ocupa, en otras palabras, son los elementos a través de los cuales se logra el objetivo que se persigue con la solución del problema. Por ejemplo si el objetivo consiste en minimizar el costo total de producción, el mismo se puede lograr a través del establecimiento de un surtido de productos, el cual conformaría en este caso el conjunto de las variables de decisión. (Felipe P. , Definición de las Variables de decisión, 1987)

✚ Construcción del Sistema de Restricciones.

El sistema de ecuaciones y/o inecuaciones junto con la condición constituyen las limitaciones que forman el conjunto posible 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. (Felipe P. , Construcción del sistema de restricciones, 1987)

Construcción de la Función objetivo.

La función objetivo es una función lineal que incluye a todas las variables del problema (aunque haya algunas que posean coeficientes nulo) y expresa el propósito central que se persigue con la solución del problema. (Felipe P. , Construcción de la función objetivo, 1987)

1.2.6 Software Educativo.

Los softwares educativos son todos aquellos programas para ordenador, creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y aprendizaje. (Marquez, 2010)

No obstante según esta definición, más basada en un criterio de finalidad que de funcionalidad, se excluyen del software educativo todos los programas de uso general en el mundo empresarial que también se utilizan en los centros educativos con funciones didácticas o instrumentales como por ejemplo: procesadores de textos, gestores de bases de datos, hojas de cálculo, editores gráficos... Estos programas, aunque puedan desarrollar una función didáctica, no han estado elaborados específicamente con esta finalidad. (Marquez, 2010)

Expresado de otra manera podemos asumir que es un producto elaborado en computadora usado con fines pedagógicos, convirtiendo a la máquina en un medio con fines educativos. Los softwares educativos pueden ser aplicados a cualquier nivel de enseñanza, así como pueden ser diseñados para cualquier tipo de materia. La forma de elaboración de los mismos está dada por la necesidad que tenga el cliente, es por ello que puede representarse como multimedia, sitios web, así como aplicaciones de desktop.

Para el desarrollo del software educativo (SE), es necesario involucrar a maestros y profesores de cada educación con experiencia suficiente, en el diseño pedagógico y evaluación del proyecto ya implementado, para garantizar el cumplimiento de los requisitos, específicos educativos, planteados para cada software. (Verano Catoní, 2009)

Se denomina software educativo al destinado a la enseñanza y el autoaprendizaje, que además permita el desarrollo de ciertas habilidades cognitivas. Así como existen profundas diferencias entre las filosofías pedagógicas, también existe una amplia gama de enfoques para la creación de software educativo atendiendo a los diferentes tipos de interacción que deberían existir entre los actores del proceso de enseñanza y aprendizaje: educador, aprendiz, conocimiento, computadora. Como software educativo tenemos desde programas orientados al aprendizaje hasta sistemas operativos completos destinados a la educación, como por ejemplo las distribuciones GNU/Linux orientadas a la enseñanza. (Castillo Lorenzo, 2008)

1.2.7 Tutoriales.

Un tutorial es una lección educacional que conduce al usuario a través de las características y funciones más importantes de cosas como aplicaciones de software, dispositivos de hardware, procesos, diseños de sistema y lenguajes de programación. (Verano Catoní, 2009)

Un tutorial normalmente consiste en una serie de pasos que van aumentando el nivel de dificultad y entendimiento. Por este motivo, es mejor seguir los tutoriales en su secuencia lógica para que el usuario entienda todos los componentes. (Marquez, 2010)

El término se utiliza mucho en Internet, ya que hay muchos sitios Web que ofrecen tutoriales, desde cómo codificar en HTML a cómo hacer que una tarjeta gráfica funcione más rápido (overclocking). Aunque un tutorial también puede presentarse impreso en papel, el término se utiliza normalmente para referirse a los programas de aprendizaje online. (Marquez, 2010)

Algunos autores utilizan el término de forma más general abarcando en la categoría Tutorial a todo el software de propósito docente, no obstante aquí solo abarcaremos aquellos que se plantean concretamente enseñar. Existen diversos tipos de tutoriales: secuenciales, ramificados e inteligentes.

1.3 El proceso de enseñanza y aprendizaje de la programación lineal.

La programación lineal como cualquier otra materia cuenta con la posibilidad de ser impartida de disímiles maneras, ya sea por los métodos tradicionales; donde se imparte por mediación de un profesor el cual apoyado en conferencias, clases prácticas, ejemplos resueltos, ejercicios

propuestos, seminarios, entre otros, logra transmitir las ideas y conceptos que posibilita el buen entendimiento de la misma por parte de quien la recibe. Otra de las tantas formas en que se ve enmarcado el proceso de enseñanza y aprendizaje de la programación lineal es mediante los textos en los cuales está contenida esta materia, que en estos momentos lo podemos encontrar en forma digital o en formato duro; es decir, en papel, ya sean libros o investigaciones desarrolladas por diferentes autores. También en algunos casos se cuenta con la posibilidad del uso de la Internet.

1.3.1 El proceso de enseñanza de la programación lineal en la UNISS.

En la actualidad la UNISS consolida 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; así como también se optimizan las estructuras y métodos de dirección. Se incrementa el uso de la informatización y las redes aseguran el empleo de las TIC. Sin embargo, para impartir la programación lineal se siguen 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. Esta materia es impartida a estudiantes que cursan el tercer año de contabilidad y finanzas como un tema de la asignatura Investigación de Operaciones. Los estudiantes para poder profundizar en el tema deben hacer búsquedas en la biblioteca sobre cualquier libro o documento que le pueda ser de utilidad. Se cuenta además con un software que permite trabajar con problemas de esta índole, el mismo sólo comprende lo relacionado con la resolución de ejercicios, no explica teoría alguna y de sentarse un estudiante a trabajar sin tener los conocimientos teóricos necesarios le sería imposible dar solución a un ejercicio. El Instituto Superior Politécnico José Antonio Echeverría cuenta con una multimedia para apoyar el estudio de la programación lineal, cuyo software se encuentra en estos momentos en todo el país; pero el mismo no se ajusta a todas las necesidades de la UNISS, sin dejar de mencionar que en gran medida, apoya el proceso docente educativo de la misma.

1.3.2 Causas que dan origen a 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 que tiene. A 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 trabajaran la parte económica de cualquier entidad del país, se convierte en un hecho la responsabilidad que tiene la UNISS en formar profesionales capaces de ayudar a mejorar la economía y tomar las correctas decisiones antes las posibles dificultades. Conociendo todo lo que se puede lograr por mediación de la programación lineal como es el caso de maximizar ganancias, minimizar costos, maximizar producción, entre otras tantas aplicaciones para las cuales puede ser útil, se convierte de vital importancia que los estudiantes adquieran la mayor cantidad de conocimiento con respecto a esta temática. De un estudio realizado en la biblioteca perteneciente a la UNISS, se pudo comprobar que de los estudiantes graduados de contabilidad y finanzas en Sancti-Spíritus desde el año 1998 hasta el año 2009, sólo 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 años vemos que:

TABLE 1 ANÁLISIS DE LOS TRABAJOS DE DIPLOMA EN LA CARRERA DE CONTABILIDAD Y FINANZAS

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 conocimiento que existe actualmente con respecto a la materia abordada. En la actualidad no existe un software que de la posibilidad a los estudiantes de aprender a plantear la modelación matemática de un problema de programación lineal, en el cual estén inmersos todos aquellos contenidos que necesitan estudiar, y de una manera amena, que cause interés. Es conocida la existencia de una multimedia relacionada con el tema, pero la misma no se ajusta a todas las necesidades de la UNISS; 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 en donde radican las principales dificultades.

Con la implementación del tutorial se quiere automatizar una parte explicativa, una evaluativa, una investigativa, una de preparación y una administrativa.

➤ Explicativa

En la parte explicativa aparecen las teorías que comprenden la definición de variables, construcción del sistema de restricciones y la construcción de la función objetivo. Aquí los estudiantes encontrarán todas aquellas teorías existentes en el software de los diferentes investigadores. Concluyendo esta parte, los estudiantes deben ser capaces de dado un problema de programación lineal definir las variables, construir el sistema de restricciones y la función objetivo.

➤ 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.

➤ Investigativa

Se les mostrarán sitios Web y documentos digitales vinculados con el tema para adquirir una mejor preparación, así como videos conferencias y explicaciones habladas.

➤ Evaluativa

Los estudiantes podrán evaluarse por mediación de autoexámenes. Aquí no sólo se miden los conocimientos adquiridos, sino que además en caso de tener alguna pregunta mal se le mostrará en que se ha equivocado.

➤ Administrativa

En esta parte juega un papel muy importante el profesor o administrador del sistema pues es el encargado de insertar todo el contenido que aparecerá en el tutorial y de tener algún error puede cambiar el fin con que está hecho el mismo. El profesor cuenta además con la posibilidad de ver todo lo que hizo un estudiante desde su primera evaluación hasta la última.

1.4 Análisis comparativo de otras soluciones existentes con la propuesta.

Existen muchas formas para aprender 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 que se ajuste a todas las necesidades de la UNISS, sólo aparecen textos que explican las teorías, llamados también por el nombre de tutorial, pero sin lograr ningún intercambio con los que acudan al mismo en búsqueda de conocimientos. También es conocida la multimedia desarrollada en el Instituto Superior Politécnico José Antonio Echeverría pero la misma no permite que sus contenidos puedan ser actualizados, no se le pueden incorporar nuevos datos.

El presente tutorial de programación lineal es un software que explica, motiva, evalúa, entre otras funcionalidades que posee; por lo que se puede considerar superior a cualquier texto tutorial que pueda ser encontrado en Internet, no obstante, estos tutoriales que existen en Internet se utilizarán como material de apoyo pues el software tiene entre sus funcionalidades almacenar en la base de datos del mismo todos aquellos sitios, tutoriales o libros que traten el

tema y que hayan sido encontrados. Cuenta con un nivel de acceso, lo que permite que su contenido sólo pueda ser modificado por personal autorizado.

1.5 Fundamentación de la metodología utilizada.

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen sin éxito.

El objetivo de un proceso de desarrollo es garantizar la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Da igual si es un producto genérico o para un cliente, hay que producir lo esperado en el tiempo esperado y con el costo esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

Muchas veces no se toma en cuenta el utilizar una metodología adecuada, sin pensar lo importante de buscar cuál sería la metodología de desarrollo más apropiada. Muchas veces el éxito de un software radica en haber encontrado la metodología correcta, que sea capaz de cumplir con el objetivo del sistema.

1.5.1 Metodología RUP.

El Rational Unified Process (RUP) es un proceso de software de ingeniería habilitado para Internet que enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos los miembros del equipo. RUP es un mentor en línea fácil de usar, que hace el proceso práctico brindando amplias guías, plantillas y ejemplos para todas las actividades críticas del desarrollo electrónico. El contenido específico de negocios electrónicos de RUP proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web y pruebas para la Red. También se proporcionan guías para desarrollar en plataformas IBM WebSphere y Microsoft Web Solution para acelerar el desarrollo de sus proyectos de desarrollo electrónico. RUP es un framework ajustable, que puede adaptarse fácilmente a la forma en que usted trabaja. Se integra estrechamente con herramientas Rational, permitiendo a los equipos de desarrollo aprovechar todas las ventajas de las características de los productos Rational, el Unified Modeling Language (UML) y otras prácticas óptimas de la industria.

RUP divide el proceso de desarrollo en ciclos, donde se obtiene un producto al final de cada ciclo. Cada ciclo se divide en cuatro fases: concepción, elaboración, construcción, y transición. Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones. (Plasencia Rodríguez, 2005)

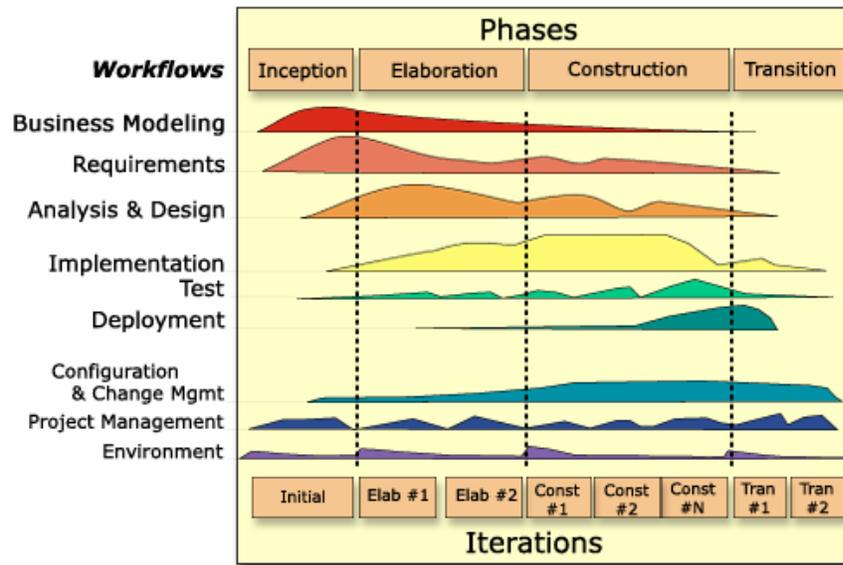


FIGURA 1 CICLOS DE DESARROLLO.

Las características principales que lo distingue son:

- ❖ Proceso Dirigido por Casos de Uso
- ❖ Proceso Iterativo e Incremental
- ❖ Proceso Centrado en la Arquitectura

El ciclo de vida iterativo se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes. En el ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. (Plasencia Rodríguez, 2005)

Cada iteración comprende:

- Planificar la iteración (estudio de riesgos)
- Análisis de los Casos de Uso y escenarios

- Diseño de opciones arquitectónicas
- Codificación y pruebas. La integración del nuevo código con el existente de iteraciones anteriores se hace gradualmente durante la construcción
- Evaluación de la entrega ejecutable (evaluación del prototipo en función de las pruebas y de los criterios definidos)
- Preparación de la entrega (documentación e instalación del prototipo)

1.5.2 Notación UML.

El Unified Modeling Language (UML) fue desarrollado en conjunto por Grady Booch, Ivar Jacobson, y Jim Rumbaugh en la Rational Software Corporation, con contribuciones de otros excelentes metodólogos, vendedores de software, y muchos usuarios. Basado en el amplio uso del Booch, OMT, y métodos de Jacobson, el UML es la evolución de estos y otros métodos para el moldeado de procesos de negocios, objetos y componentes. El UML brinda el lenguaje de aplicación de modelado para:

- ❖ Modelado de proceso de negocios con casos de uso.
- ❖ Modelado de clases y objetos.
- ❖ Modelado de componentes.
- ❖ Modelado de distribución y despliegues.

Es el lenguaje estandarizado en la industria para especificar, visualizar, construir y documentar los dispositivos de sistemas de software. Simplifica el complejo proceso de diseño de software, haciendo "planos" para su construcción. (Plasencia Rodríguez, 2005)

El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad). Los perfiles del UML son las extensiones a las notaciones estándares del UML usando los mecanismos de extensión del UML: los estereotipos, los valores etiquetados y las restricciones. Presenta nueve diagramas estándares:

diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue, si fuera necesario, el diagrama de objetos se puede crear usando los diagramas de colaboración. Las diferencias existen solamente con relación al soporte de una característica específica durante la creación de los diagramas UML y los perfiles extendidos del UML.

UML es un lenguaje estándar con el que es posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Con él sus protagonistas se propusieron que:

El método debía ser capaz de modelar no sólo sistemas de software sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).

Crear un lenguaje para modelado utilizable a la vez por máquinas y por personas.

Establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.

Manejar los problemas típicos de los sistemas complejos de misión crítica

Los modelos se utilizan en muchas actividades de la vida humana: antes de construir una casa el arquitecto utiliza un plano, los músicos representan la música en forma de notas musicales, los artistas pintan sobre el lienzo con carboncillos antes de empezar a utilizar los óleos, etc. Unos y otros abstraen una realidad compleja sobre unos bocetos, modelos al fin y al cabo, UML obtiene distintos puntos de vista de la realidad que modela mediante los distintos tipos de diagramas que posee. Con la creación del UML se persigue obtener un lenguaje que sea capaz de abstraer cualquier tipo de sistema, sea informático o no, mediante los diagramas, es decir, mediante representaciones gráficas que contienen toda la información relevante del sistema. Un diagrama es una representación gráfica de una colección de elementos del modelo, que habitualmente toma forma de grafo donde los arcos que conectan sus vértices son las relaciones entre los objetos y los vértices se corresponden con los elementos del modelo. Los

distintos puntos de vista de un sistema real que se quieren representar para obtener el modelo se dibuja de forma que se resaltan los detalles necesarios para entender el sistema.

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas. UML es un método independiente del proceso. Los procesos de desarrollo deben ser definidos dentro del contexto donde se van a implementar los sistemas. (Plasencia Rodríguez, 2005)

Desarrollo Iterativo

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Trabajo en Grupo

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Generador de Código

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Ingeniería Inversa

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Ingeniería de Código

La ingeniería de código consiste en la ingeniería directa -desde el modelo al código- y en la ingeniería reversa -desde el código al modelo-. Una vez que se completa el diseño (los modelos de diseño y de datos) se puede usar la información del modelo para generar el código fuente en un lenguaje de programación específico o los scripts DDL para una base de datos. A medida que los desarrolladores agregan o modifican el código o la implementación de la base de datos, se pueden sincronizar el diseño y el modelo de datos con el código o con el script DDL para mantenerlos consistentes entre sí. (Plasencia Rodríguez, 2005)

1.6 Tecnologías y herramientas que serán utilizados.

La solución de los problemas asociados al tratamiento de los datos en los sistemas tradicionales lleva a un cambio radical en el enfoque del sistema de información, en el cual los datos se organizan y mantienen en un conjunto estructurado que no está diseñado para una aplicación concreta, sino que, por el contrario, tiende a satisfacer las necesidades de información de la organización; necesidades cuya diversidad se ve acentuada con el transcurso del tiempo.

Concepto de base de datos: Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo. Los datos deben estar interrelacionados y estructurados. El concepto de base de datos ha ido cambiando y configurándose a lo largo del tiempo, en la actualidad, y de acuerdo con las características a las que deben responder las bases de datos, podemos definir a una base de datos como:

Colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien

determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos.

Las bases de datos al final, pueden brindar infinitudes de servicios a sus usuarios, gracias a su combinación con los lenguajes de programación. En esta unión está la magia del tratamiento de las bases de datos.

Lenguaje de programación es un conjunto de sintaxis y reglas semánticas que definen los programas del computador.

Es una técnica estándar de comunicación para entregarle instrucciones al computador. Un lenguaje le da la capacidad al programador de especificarle al computador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

Un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, es decir, ser traducido al lenguaje de máquina para que pueda ser ejecutado por el ordenador.

1.6.1 Arquitectura de desarrollo de N Capas.

La Arquitectura de Aplicaciones en n-capas se ha transformado en el modelo para el Software empresarial. Se determina por la descomposición de las aplicaciones.

- ❖ Provee una escalabilidad, capacidad de administración y manejo de recursos mejorados.
- ❖ Cada capa es un conjunto de elementos que realizan una función determinada.
- ❖ Se logra actualizar una capa sin recompilar otras capas.

Arquitectura de 3 capas

- ❖ Capa de presentación
- ❖ Capa de negocios
- ❖ Capa de datos

Este modelo de diseño es mostrado en la siguiente figura:

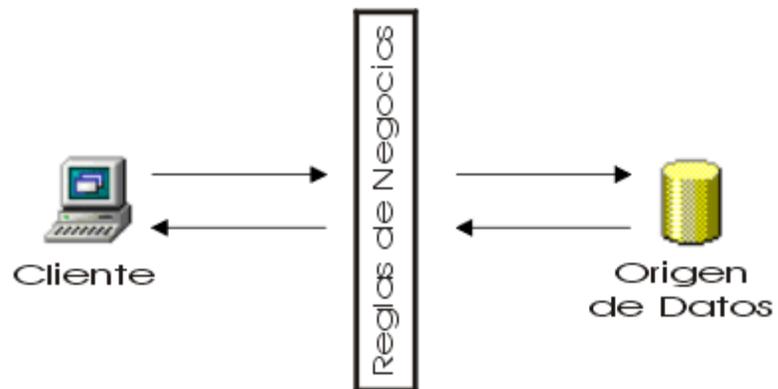


FIGURA 2 MODELO DE DISEÑO EN TRES CAPAS

Capas de Presentación

- ❖ Formularios
- ❖ Informes
- ❖ Respuestas al usuario

Por regla general, la capa de la presentación es una interfaz gráfica que muestra los datos a los usuarios. (Cordero Carrasco)

Capa de Negocio

- ❖ Reglas del negocio
- ❖ Validaciones
- ❖ Cálculos
- ❖ Flujos y procesos

La capa de la lógica de negocios es responsable de procesar los datos recuperados y enviarlos a la capa de presentación. (Cordero Carrasco)

Capa de datos

- ❖ Base de datos
- ❖ Tablas
- ❖ Procedimientos almacenados
- ❖ Componentes de datos

La capa de datos almacena los datos de la aplicación en un almacén persistente, tal como una base de datos relacional o archivos XML. (Cordero Carrasco)

Se pueden alojar todas las capas en el mismo servidor, pero también es posible alojar cada capa en varios servidores. (Cordero Carrasco)

1.6.2 Microsoft SQL Server 2000.

La forma recomendada de manipular bases de datos es utilizando un servidor de bases de datos, ya que es un sistema mucho más robusto y escalable que archivos individuales como las bases de datos de Access, archivos de texto plano, etc.

SQL Server es el servidor de bases de datos de Microsoft, seguro, robusto y con las más avanzadas prestaciones: transacciones, procedimientos almacenados, triggers, etc.

SQL Server 2000 proporciona a los usuarios una excelente plataforma de base de datos optimizada para procesamiento de transacciones en línea (OLTP, Online Transactional Processing) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. Ofrece a los usuarios un entorno de Lenguaje de marcado extensible (XML, Extensible Markup Language) totalmente integrado, agrega una nueva característica de minería de datos en Analysis Services y mejora la tecnología de depósito con Meta Data Services.

Analysis Services presenta la característica de minería de datos, que se puede emplear para descubrir información en cubos OLAP y bases de datos relacionales. SQL Server 2000 mejora los controles de seguridad y la conectividad del cliente, y hace posible el análisis de los datos en tiempo real. (Plasencia Rodríguez, 2005)

Algunas de sus características son:

Compatibilidad con XML: Simplifica la integración de los sistemas de servicios de fondo y la transferencia de datos a través de servidores de seguridad mediante XML.

Integración con Microsoft Windows Server System, Microsoft BizTalk y Microsoft Commerce Server: SQL Server 2000, en combinación con otros productos de Microsoft Server, proporciona aún mayor eficacia para los negocios de comercio electrónico. **Acceso Web a los datos:** conecta las bases de datos de manera flexible, mediante el Web sin necesidad de ninguna programación adicional.

Alojamiento de aplicaciones: gracias a la compatibilidad con varias instancias, SQL Server permite aprovechar totalmente sus inversiones en hardware de forma que múltiples aplicaciones se pueden ejecutar en un solo servidor o externamente.

Análisis del flujo de clientes: compresión más detallada del comportamiento del cliente en línea para tomar mejores decisiones.

Seguridad: las aplicaciones son seguras en cualquier entorno de red, con la seguridad basada en funciones y el cifrado de archivos y de la red.

Búsqueda de texto: Utilización y administración de los datos estructurados y sin estructurar, incluida la búsqueda a través de documentos de Microsoft Office.

Escalabilidad: aplicaciones hasta 32 CPU y 64 gigabytes (GB) de RAM. SQL.

Las características de ajuste y mantenimiento automáticos permiten a los administradores centrar su atención en otras tareas críticas. Las funciones definidas por el usuario, la integridad referencial en cascada y el depurador integrado de Transact SQL permiten la reutilización del código para simplificar el proceso de desarrollo. Permite a los usuarios realizar consultas en inglés, en lugar de utilizar expresiones multidimensionales (MDX).

SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor Web sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos. El servidor SQL se encarga de todo. Esta división de papeles es mucho más conveniente para la publicación Web.

Puede recibir conexiones (clientes que desean manipular datos) desde cualquier ordenador conectado; escucha un puerto TCP/IP. En este aspecto es igual que un servidor Web o un servidor de correo. La diferencia es el número de puerto y, naturalmente, el protocolo con el que se comunicará con el cliente. Un navegador es un cliente para un servidor Web, pero no conoce el protocolo para hablar con un servidor de bases de datos. Por ejemplo, Access 2000 o Access XP son clientes para SQL Server. (Plasencia Rodríguez, 2005)

SQL Server admite la programación cliente-servidor, que se utiliza cuando queremos realizar aplicaciones que utilicen redes y que comuniquen entre sí a varios ordenadores. Básicamente consiste en que el programa se divide en dos partes:

La parte Cliente, que reside en el equipo donde está el usuario y se encarga de la interacción con éste.

La parte Servidor, que reside en un ordenador conectado a la red permanentemente y se encarga de manipular los datos.

Ambas partes de la aplicación se comunican entre sí utilizando algún protocolo de red TCP/IP. La justificación de este paradigma es la minimización del tráfico de red, sobre todo para evitar ralentizaciones y economizar el ancho de banda.

SQL Server admite procedimientos almacenados (stored procedures) realizados en lenguaje SQL. Se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Los procedimientos guardados podrían ser la "parte servidor" de una aplicación cliente-servidor. La parte cliente podría ser una aplicación ASP o incluso una aplicación en Visual Basic o Access, que puede acceder directamente a SQL Server.

SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos. Es un servidor y está pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado. (Alejo Machado, 2006)

1.6.3 Borland Delphi 2006.

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 sólo funcionan sobre la plataforma de trabajo Windows.

Este entorno de desarrollo integrado incluye soporte para las aplicaciones Delphi/Delphi .NET, C, C++ y C#. Los clientes que adquieran cualquiera de estos productos obtendrán un soporte para los cuatro lenguajes sin costes adicionales, pudiendo personalizar las capacidades específicas del lenguaje.

Las nuevas mejoras de Delphi 2006, incluyen un sistema de plantillas en directo, refactorings avanzados que automatizan los cambios de código fuente y la integración con herramientas ALM de Borland.

Los usuarios de Delphi también pueden acelerar el desarrollo con soporte para ingeniería inversa, prototipaje rápido e integración avanzada de IDE con modelado UML, requerimientos, control de versiones, seguimiento de bugs y colaboración entre equipos. Delphi 2006 cuenta con soporte para Microsoft .NET Framework, ASP.NET, VCL.NET, VCL y Win32 SDK. Además, dado que Delphi soporta tanto Win32 como .NET, los clientes pueden continuar con el desarrollo y mantenimiento de las aplicaciones Win32 existentes, a la vez que avanzan hacia el marco de .NET y las aplicaciones Windows de próxima generación.

Principales mejoras en Delphi 2006.

- Nuevas funcionalidades de productividad para el IDE – El sistema de “plantillas en directo” permite que los usuarios naveguen por los campos e inserten puntos de cualquier plantilla, con una función añadida de finalización “Smart Block” que anticipa y añade cierres de

bloques a medida que usted escribe, y un nuevo editor “Change Bar” que indica las líneas modificadas y las líneas modificadas de sesiones guardadas.

- Librería visual de componentes (VCL) mejorada – La VCL mejorada de Delphi 2006 permite el desarrollo de interfaces gráficas de usuario (GUI) con una selección de componentes UI.
- IDE extendido / Integración ALM – Los clientes de Delphi 2006 pueden tomar el control del proceso de desarrollo mediante el apalancamiento de niveles de integración con los productos ALM de Borland, incluidas la gestión de cambios y configuración del software StarTeam, el modelado Together y las soluciones de gestión de requisitos de CaliberRM –ahora con la integración completa entre el StarTeam nativo y el IDE de Caliber. (10ab)

1.7 Conclusiones.

Partiendo del estudio del problema a resolver y la comparación de las soluciones existentes se llegó a la conclusión de que resulta novedoso e importante implementar un sistema que ofrezca una serie de beneficios en la enseñanza y aprendizaje de la programación lineal.

Después del análisis realizado en este capítulo sobre las tendencias y tecnologías actuales relacionadas con la propuesta, y de un estudio de las condiciones de la entidad para la cual va dirigida el sistema a implementar concluimos que será desarrollado con la herramienta de programación Borland Delphi 2006 como lenguaje, utilizando el gestor de base de datos SQL Server 2000, y para el análisis y diseño del sistema se utilizará la metodología RUP con UML como lenguaje de modelado.

Capítulo 2. Descripción de la solución propuesta.

2.1 Introducción.

En el presente capítulo se pretende brindar lo más claramente posible, una idea de la solución propuesta. Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Para lograr esos propósitos, el proceso de modelado permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso y de objetos.

Un modelo de casos de uso describe los procesos de una entidad en términos de casos de uso y actores que se corresponden con los procesos y los clientes, respectivamente. Un modelo de objetos es un modelo que describe cómo colaboran los trabajadores y las entidades dentro del flujo.

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.

La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema. (Alejo Machado, 2006)

Este dominio reside su funcionamiento en el continuo acceso de los usuarios al tutorial con el objetivo de asimilar conocimientos teóricos y prácticos sobre la Programación Lineal, haciendo uso para esto de disímiles fuentes teóricas que han explicado este tema, de ejercicios resueltos, artículos y ejemplos que ayuden a desarrollar las habilidades de los usuarios. Con la posibilidad de evaluarse.

2.2.1 Principales conceptos asociados al dominio del problema.

Estudiantes: Es la entidad que se refiere a los usuarios del tutorial. Los mismos son los estudiantes de Contabilidad y Finanzas de la UNISS.

Profesor: Es la entidad que se refiere a la persona con total acceso al tutorial, el encargado de la gestión de los datos del tutorial. Además puede controlar la actividad de los usuarios en el tutorial.

Evaluación: Se refiere a la clase principal que permite a los usuarios comprobar los conocimientos adquiridos y al administrador de gestionar todos los ejercicios por los cuales está compuesta.

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

2.2.2 Modelo de objetos del dominio.

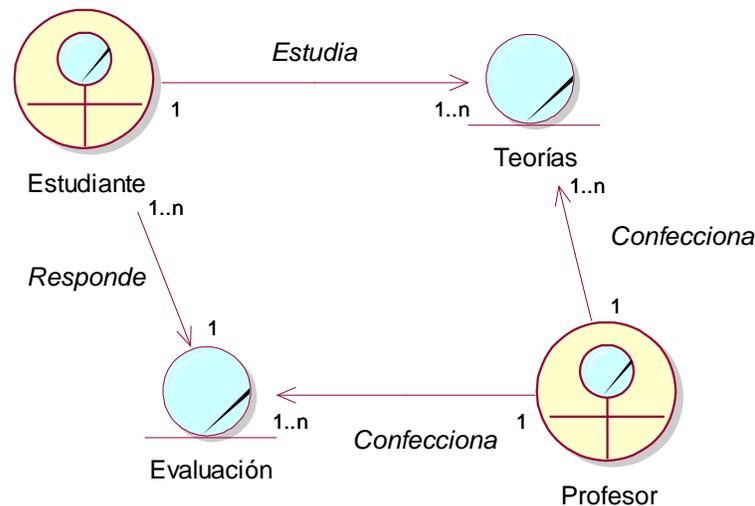


FIGURA 3 MODELO DE OBJETO

2.2.3 Reglas a considerar.

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

- Para tener acceso a la parte evaluativa 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 sea problemas propuestos, problemas resueltos, teoría, sitios web 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 o un usuario con privilegios.
- 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.

2.3 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.

1. Crear cuenta de usuario
2. Autenticarse
3. Cambiar contraseña
4. Gestionar usuario
 - 4.1 Asignar roles de seguridad
 - 4.2 Asignar funcionalidades a los roles
 - 4.3 Desactivar cuenta de usuario

- 5. Gestionar teoría
 - 5.1 Insertar teoría
 - 5.2 Modificar teoría
 - 5.3 Eliminar teoría
- 6. Mostrar teoría
- 7. Gestionar problemas propuestos
 - 7.1 Insertar problemas propuestos
 - 7.2 Modificar problemas propuestos
 - 7.3 Eliminar problemas propuestos
- 8. Mostrar problemas propuestos
- 9. Gestionar problemas resueltos
 - 9.1 Insertar problemas resueltos
 - 9.2 Modificar problemas resueltos
 - 9.3 Eliminar problemas resueltos
- 10. Mostrar problemas resueltos
- 11. Mostrar trazas
- 12. Gestionar sitios Web
 - 12.1 Insertar sitios Web
 - 12.2 Modificar sitios Web
 - 12.3 Eliminar sitios Web
- 13. Mostrar sitios Web
- 14. Gestionar Artículos
 - 14.1 Insertar Artículos
 - 14.2 Eliminar Artículos
- 15. Mostrar Artículos
- 16. Configurar el sistema
- 17. Hacer autoexamen
 - 17.1 Mostrar autoexamen

- 17.2 Responder autoexamen
- 17.3 Mostrar resultado
- 17.4 Guardar resultados del autoexamen
- 18. Gestionar ejercicios de autoexamen
 - 18.1 Gestionar ejercicios de selección
 - 18.1.1 Insertar ejercicios de selección
 - 18.1.2 Modificar ejercicios de selección
 - 18.1.3 Eliminar ejercicios de selección
 - 18.2 Gestionar ejercicios de enlazar
 - 18.2.1 Insertar ejercicios de enlazar
 - 18.2.2 Modificar ejercicios de enlazar
 - 18.2.3 Eliminar ejercicios de enlazar
 - 18.3. Gestionar ejercicios de verdaderos y falsos
 - 18.3.1 Insertar ejercicios de verdaderos y falsos
 - 18.3.2 Modificar ejercicios de verdaderos y falsos
 - 18.3.3 Eliminar ejercicios de verdaderos y falsos
- 19. Gestión de novedad
 - 19.1 Clasificar novedad
 - 19.2 Eliminar novedad
- 20. Entrar novedad
- 21. Mostrar novedad
- 22. Mostrar reportes
- 23. Mostrar ayuda
- 24. Mostrar listado de usuarios

Se consideraron como actores del sistema:

- Administrador

- Usuario con privilegio
- Usuario
- Invitado

2.4 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. (Alejo Machado, 2006)

Expresado de otra manera, 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 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.

- La interfaz se ajustará al estándar de ventanas que el sistema operativo Windows ha establecido e internacionalizado.

- La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible. La interfaz será uniforme de manera que a pesar de tener varios módulos, el usuario los identifique como un mismo sistema

Usabilidad: Creemos 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.

- La explotación del sistema agilizará la actividad científica del centro donde se ponga en práctica gracias a las comodidades brindadas por esta aplicación, la rapidez y organización.

- La información original del software sólo podrá ser modificada por especialistas en la materia. El sistema utiliza la política de impedir accesos no autorizados que pudieran introducir errores en la información original.

- El sistema podrá ser usado por aquellas personas que no tengan experiencia en el uso de la computadora, sólo necesitarían un ligero entrenamiento sobre el funcionamiento de los principales elementos de una interfaz estándar en el ambiente del sistema operativo Windows (uso del Mouse, manejo de menús, botones, cuadros de texto, etc.)

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.

Las pruebas del sistema se realizarán en la Universidad de Sancti-Spíritus. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.

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. Se permitirán accesos sin autorización al sistema, para ello se definirá una política de usuarios sin cuenta y usuarios con roles y privilegios diferentes para que la información pueda ser cambiada por las personas autorizadas. Para ello el sistema contará con cuatro tipos de usuarios, el administrador, usuarios con privilegios, usuarios y los invitados, 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 Windows 2000
Advanced Server o Windows 2003 Server

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.
- Framework 1.1

Restricciones en el diseño y la implementación: Se desea desarrollar una herramienta dinámica para la automatización de los procesos de enseñanza y aprendizaje 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.

2.5 Descripción del sistema propuesto.

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

Cuando cualquier usuario comienza a hacer uso del sistema a través de la computadora ubicada en el laboratorio, no necesita autenticarse al sistema para ver la información que ya está publicada y disponible a cualquier usuario, no obstante el estudiante si desea formar parte del sistema tiene la posibilidad de crearse su propia cuenta de usuario. Esta información se refiere inicialmente, a que cuando un usuario solicita ver el tutorial, se muestra todo lo referente a libros, teorías, ejercicios, 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 profesor podrá tener constancia de la participación del estudiante y su nivel alcanzado por mediación de los resultados de sus evaluaciones. El estudiante una vez usuario del sistema cuenta con la posibilidad de realizar autoexámenes.

El usuario cuenta con una parte de investigación donde encontrará los libros existentes en el sistema en formato digital los cuales pueden ser vistos desde la propia aplicación o bien pueden ser copiados para la computadora o memoria flash. Puede además ver la dirección de sitios vinculados al tema facilitándole la búsqueda de información que pueda servir de utilidad.

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

- Ejercicios resueltos.
- Ejercicios propuestos.

Los ejercicios resueltos serán mostrado al usuario con el procedimiento que se fue llevando a cabo para su solución y los propuestos son para que el usuario sea quien lo responda, aunque en estos últimos puede ver su respuesta, ya sea para comprobarla con el resultado obtenido o para conocerla si no fue capaz de resolverlo. Es por ello que estos ejercicios se dividen en niveles de dificultad que van desde fáciles, medios hasta difíciles, donde el estudiante debe ser capaz de saber en cual nivel se encuentra para poder hacer un buen uso del tutorial.

El tutorial tendrá un módulo en el cual los usuarios pueden evaluarse mediante autoexámenes a diferentes niveles de dificultad donde el estudiante puede escoger un nivel y realizar tres tipos de preguntas, una de verdaderos y falsos para evaluar los principales conceptos que deben ser dominado, una de enlazar y una de seleccionar, estas dos últimas evaluarían los conocimientos prácticos. Una vez realizado el autoexamen se mostrará la nota final, así como las preguntas contestadas incorrectamente con las respuestas correctas que deberían haber sido marcadas. Los autoexámenes serán preguntas generadas aleatoriamente.

En la parte que comprende a lo novedoso aparecerán todas aquellas reflexiones que pueden ser útiles, realizadas por los usuarios y validadas por el administrador. En esta parte el usuario podrá expresar libremente que le gustó o que no le gustó, e incluso si tiene algo que aportar al sistema, o bien un artículo reciente que trate la temática y quiera mostrárselo a los demás usuarios. Aquí el administrador del sistema verá todas las anotaciones realizadas por los estudiantes y publicará las que él considere apropiadas.

Por parte de la administración del sistema comprendido por los administradores o usuarios con privilegios, se tiene lo siguiente:

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 todos los autoexámenes desarrolladas por los estudiantes en la aplicación, así como los resultados obtenidos en los mismos. También es el encargado de asignar roles a los usuarios, los cuales según el rol asignado podrán desempeñar tareas en el sistema que como usuario normal no pueden hacer; como es el caso de entrar ejercicios o libros, entre otras funciones en las cuales el administrador le puede dar permiso.

Actores del Sistema

Un actor representa a cualquier elemento que interactúa con el sistema, que puede: intercambiar información con él, ser un recipiente pasivo de información, representar el rol que juega una o varias personas, un equipo o un sistema automatizado. Estas operaciones pueden ser realizadas por una persona, un software, o un hardware.

A continuación se describe los actores identificados en el sistema:

TABLE 2 ACTORES DEL SISTEMA

Administrador	Es la persona encargada de llevar el control del estado de los usuarios y su seguridad en el acceso a la información. Es el encargado de modificar, insertar, o eliminar el contenido del sistema Es el que tiene el control total del sistema
Usuario con privilegios	Es el usuario que tiene la posibilidad de hacer los cambios en el sistema, para los cuales el administrador le ha dado derecho.
Usuario	Es el que interactúa con el sistema pero no tiene derechos para modificarlo. Es miembro de la aplicación por lo que tiene el derecho de hacer autoexámenes.
Invitado	Es el usuario que interactúa con el sistema sin tener cuenta, no tiene ningún tipo de privilegio para modificar el sistema y no puede evaluarse.

2.6 Modelo de casos de uso del sistema.

Los casos de uso forman parte de los artefactos del sistema, son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado a los actores, sus atributos representan valores que una instancia de caso de uso utiliza durante su ejecución, estas instancias se consideran atómicas, es decir indivisibles. Con un caso de uso se representa cada forma en que los actores usan el sistema y se especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

Para una mejor organización de los elementos podemos agruparlos a través de paquetes, así se subdivide el modelo en otros más pequeños que colaboran entre sí. Por eso decidimos realizar el diagrama de casos de uso del sistema dividido por paquetes, pues de lo contrario resultaría muy extenso y por ende difícil de comprender.

La descripción de los casos de uso se hace para describir en detalle el flujo de sucesos, de una manera precisa pero a la vez fácil de leer. Se describen todas las excepciones y alternativas, incluyéndose pre y pos condiciones. Es importante definir cómo comienza, termina e interactúa cada caso de uso con sus actores.

Diagrama de Paquetes.

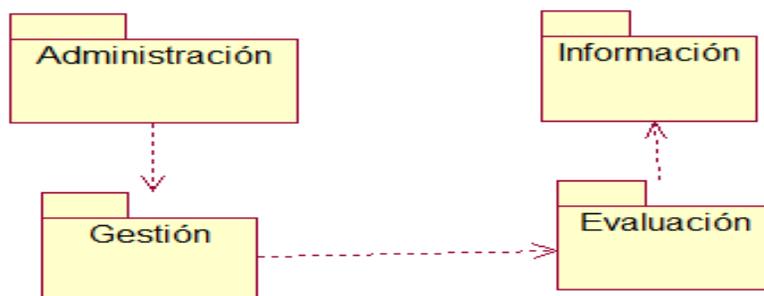


FIGURA 4 DIAGRAMA DE PAQUETE

2.6.1 Paquete Administración.

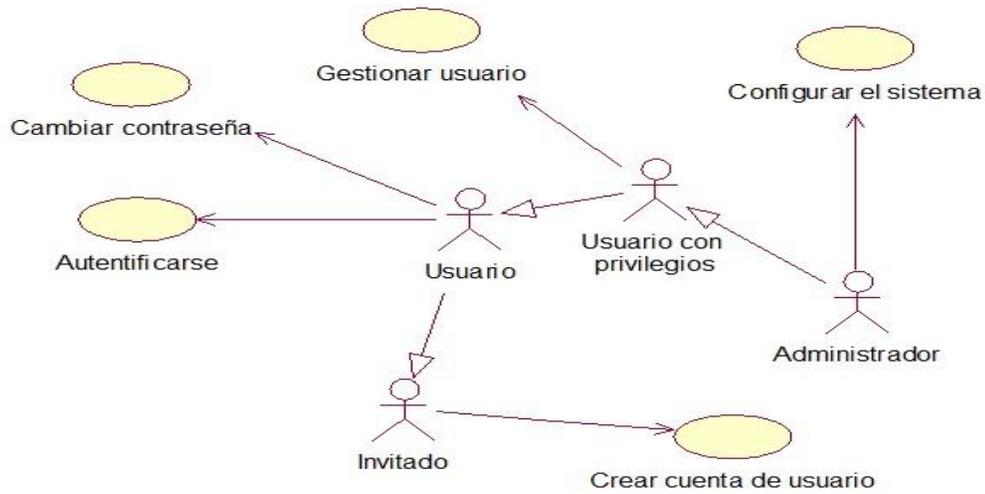


FIGURA 5 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE ADMINISTRACIÓN

2.6.2 Paquete Gestión.

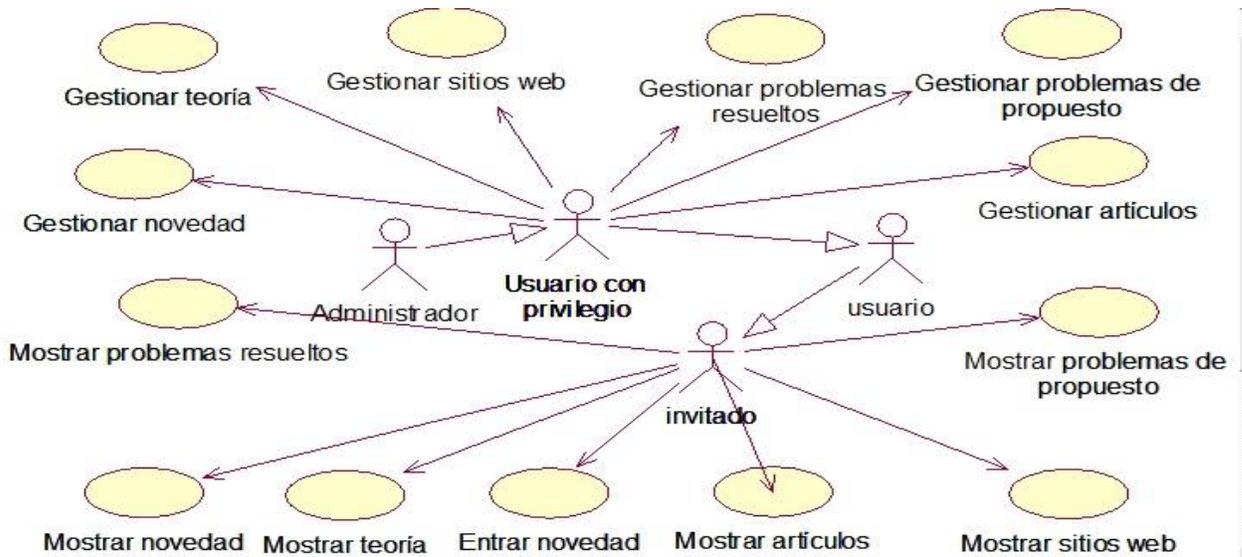


FIGURA 6 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE GESTIÓN

2.6.3 Paquete Información

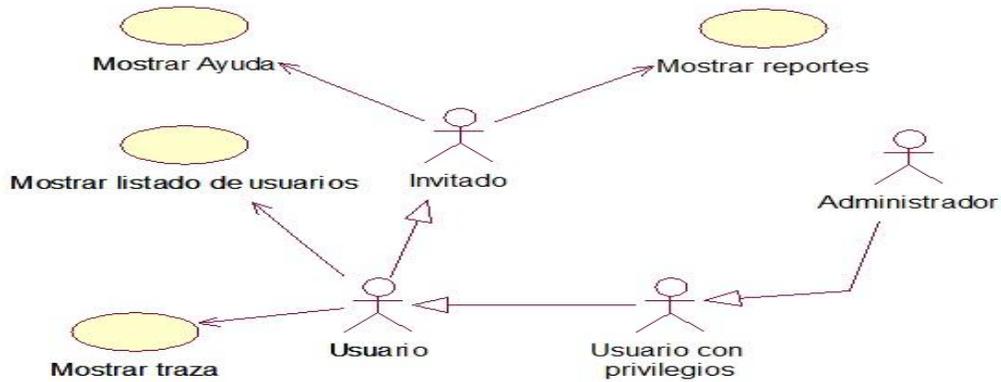


FIGURA 7 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE INFORMACIÓN

2.6.4 Paquete Evaluación

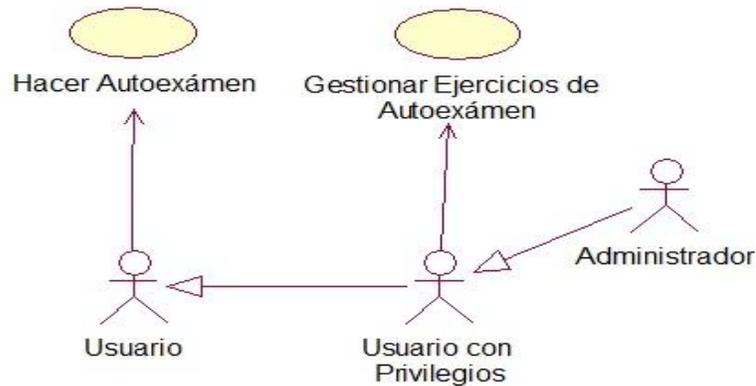


FIGURA 8 DIAGRAMA DE CASOS DE USO DEL SISTEMA PAQUETE EVALUACIÓN

2.7 Descripción de los casos de uso del sistema

Caso de Uso	Mostrar teoría
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar toda la teoría
Resumen	<p>Cuando cualquier usuario solicita ver la teoría existente en la aplicación. Se le muestra el formulario con toda la teoría organizada por un orden específico mostrándole el autor, el título y la teoría en materia, una vez aceptada la teoría seleccionada se le muestra y finaliza el caso de uso.</p>

Referencias	6
Prototipo	
Ver anexo 39	

Caso de Uso	Mostrar problemas propuestos
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar todos los problemas propuestos
Resumen	
<p>Cuando cualquier usuario solicita ver los problemas propuestos existente en la aplicación. Se le muestra el formulario con todos los problemas propuestos organizados por un orden específico mostrándole el título, el nivel de dificultad, y el problema, una vez aceptado uno en particular se le muestra.</p> <p>El caso de uso termina en cuanto acaba el paso anterior.</p>	
Referencias	8
Prototipo	
Ver anexo 38	

Caso de Uso	Mostrar problemas resueltos
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar todos los problemas resueltos
Resumen	
<p>Cuando el administrador solicita ver los problemas resueltos existente en la aplicación. Se le muestra el formulario con todos los problemas resueltos organizados por un orden específico mostrándole el nivel de dificultad, una vez aceptado un problema resuelto se le muestra.</p> <p>El caso de uso termina en cuanto acaba el paso anterior y es válido para los demás usuarios.</p>	
Referencias	10
Prototipo	
Ver anexo 38	

Caso de Uso	Mostrar sitios Web
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar todos los sitios Web
Resumen	
<p>Cuando el administrador solicita ver los sitios Web existente en la aplicación. Se le muestra el formulario con todos los sitios Web organizados por un orden específico mostrándole la dirección de los sitios Web que abordan temas relacionados con los tratados en la aplicación.</p> <p>El caso de uso termina en cuanto acaba el paso anterior y es válido para los demás usuarios.</p>	
Referencias	13
Prototipo	
Ver anexo 45	

Caso de Uso	Mostrar novedad
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar todas las novedades
Resumen	
<p>Cuando el administrador solicita ver las novedades existente en la aplicación. Se le muestra el formulario con todas las novedades organizados por un orden específico mostrándole el autor y la novedad. Todo lo descrito anteriormente es válido para los demás usuarios</p> <p>El caso de uso termina en cuanto acaba el paso anterior.</p>	
Referencias	21
Prototipo	
Ver anexo 37	

Caso de Uso	Mostrar artículos
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Mostrar todos los artículos

Resumen	
<p>Cuando el administrador solicita ver los artículos existentes en la aplicación. Se le muestra el formulario con todos los artículos organizados por un orden específico mostrándole el autor, descripción y el título. Una vez seleccionado un artículo en específico tiene la oportunidad de verlo en la misma aplicación o bien descargarlo para cualquier medio de almacenamiento ya sea el disco duro de la maquina en la que se encuentra trabajando o para una memoria flash. Todo lo descrito anteriormente es válido para los demás usuarios</p> <p>El caso de uso termina en cuanto acaba el paso anterior.</p>	
Referencias	15
Prototipo	
Ver anexo 36	

Caso de Uso	Gestionar teoría.
Actores	Administrador, Usuario con privilegios
Propósito	Adicionar, modificar o eliminar teoría.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> 1. Cuando el administrador solicita adicionar una nueva teoría. Se le muestra el formulario para la entrada de datos, una vez aceptada la operación se le muestra el listado de teoría con los datos actualizados. 2. Cuando el administrador solicita modificar una teoría. Se le muestra el formulario con los datos de la teoría, donde el administrador puede cambiar los datos, una vez aceptado se pasa al listado actualizado. 3. Cuando el administrador solicita eliminar una teoría. Se muestra el formulario con todas las teorías, una vez aceptada la operación, se elimina y se muestra el listado actualizado. <p>El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios que posean este privilegio.</p>	
Referencias	5, 5.1, 5.2, 5.3
Prototipo	
Ver anexo 46	

Caso de Uso	Gestionar problemas propuestos.
Actores	Administrador, Usuario con privilegios
Propósito	Adicionar, modificar o eliminar problemas propuestos.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> 1. Cuando el administrador solicita adicionar un nuevo problema propuesto. Se le muestra el formulario para la entrada de datos, una vez aceptada la operación se le muestra el listado de los problemas propuestos con los datos actualizados. 2. Cuando el administrador solicita modificar un problema propuesto. Se le muestra el formulario con los datos del problema propuesto, donde el administrador puede cambiar los datos, una vez aceptado se pasa al listado actualizado. 3. Cuando el administrador solicita eliminar un problema propuesto. Se le muestra el formulario con los problemas propuestos, una vez aceptada, se elimina y se muestra el listado actualizado. <p>El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios con este privilegio.</p>	
Referencias	7, 7.1, 7.2, 7.3
Prototipo	
Ver anexo 40	

Caso de Uso	Gestionar problemas resueltos.
Actores	Administrador, Usuario con privilegios
Propósito	Adicionar, modificar o eliminar problemas resueltos.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> 1. Cuando el administrador solicita adicionar un nuevo problema resuelto. Se le muestra el formulario para la entrada de datos, una vez aceptada la operación se le muestra el listado de los problemas resueltos con los datos actualizados. 2. Cuando el administrador solicita modificar un problema resueltos. Se le muestra el formulario con los datos del problema resuelto, donde el administrador puede cambiar los datos, una vez aceptado se pasa al listado actualizado. 3. Cuando el administrador solicita eliminar un problema resuelto. Se le muestra el listado de los problemas resueltos, una vez aceptada, se elimina y se muestra el 	

listado actualizado.	
El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios con este tipo de privilegio.	
Referencias	9, 9.1, 9.2, 9.3
Prototipo	
Ver anexo 40	

Caso de Uso	Gestionar sitios Web.
Actores	Administrador, Usuario con privilegios
Propósito	Adicionar, modificar o eliminar sitios Web.
Resumen	
Se inicia por tres vías distintas:	
1. Cuando el administrador solicita adicionar un nuevo sitio Web. Se le muestra el formulario para la entrada de datos, una vez aceptada la operación se le muestra el listado de los sitios Web con los datos actualizados.	
2. Cuando el administrador solicita modificar un sitio Web. Se le muestra el formulario con los datos del sitio Web, donde el administrador puede cambiar los datos, una vez aceptado se pasa al listado actualizado.	
3. Cuando el administrador solicita eliminar un sitio Web. Se le muestra el formulario con los sitio Web, una vez aceptada, se elimina y se muestra el listado actualizado.	
El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios con este tipo de privilegio.	
Referencias	12, 12.1, 12.2, 12.3
Prototipo	
Ver anexo 45	

Caso de Uso	Gestionar novedades.
Actores	Administrador, Usuarios con privilegios
Propósito	Clasificar o eliminar novedades.
Resumen	
Se inicia por tres vías distintas:	

1. Cuando el administrador solicita clasificar una nueva novedad. Se le muestra el formulario con los datos de la novedad, donde el administrador puede clasificarla como adecuada, una vez aprobada por el administrador la novedad pasa a ser parte visible de la aplicación.	
2. Cuando el administrador solicita eliminar una novedad. Se muestra las novedades para que seleccione la deseada, una vez aceptada, se elimina y se muestra el listado actualizado.	
El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios que cuenten con este privilegio.	
Referencias	19, 19.1, 19.2
Prototipo	
Ver anexo 35	

Caso de Uso	Gestionar artículos.
Actores	Administrador, Usuarios con privilegios
Propósito	Adicionar y eliminar artículos.
Resumen	
Se inicia por dos vías distintas:	
1. Cuando el administrador solicita adicionar un nuevo artículo. Se le muestra el formulario para la entrada de datos, una vez aceptada la operación se le muestra el listado de artículos con los datos actualizados.	
2. Cuando el administrador solicita eliminar un artículo. Se muestra un formulario con los artículos una vez seleccionado el deseado y aceptada la operación, se elimina y se muestra el listado actualizado.	
El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios con este tipo de privilegio.	
Referencias	14, 14.1, 14.2
Prototipo	
Ver anexo 32	

Caso de Uso	Hacer autoexamen.
Actores	Administrador, Usuario con privilegios, Usuarios

Propósito	Evaluar los conocimientos adquiridos en el tutorial.
Resumen	
<p>Se inicia cuando el administrador solicita realizar un autoexamen, donde se le muestra un formulario para elegir el nivel de dificultad que desea hacer, una vez elegido el nivel de dificultad se le generan tres tipos de ejercicios que son seleccionados por un orden aleatorio acorde al nivel de dificultad especificado, los cuales pueden ser fácil, medio o difícil. Los tipos de ejercicios mostrados son, uno de selección, uno de enlazar y uno de verdaderos y falsos. Al terminar de resolverlos se les muestra la calificación obtenida y se guardan los resultados.</p> <p>El caso de uso termina una vez terminado el autoexamen y es válido para los demás usuarios, exceptuando los invitados.</p>	
Referencias	17, 17.1, 17.2, 17.3, 17.4
Prototipo	
Ver anexo 25	

Caso de Uso	Gestionar ejercicios del autoexamen.
Actores	Administrador, Usuario con privilegios
Propósito	Adicionar, modificar o eliminar ejercicios de autoexamen.
Resumen	
<p>Se inicia por tres vías distintas:</p> <ol style="list-style-type: none"> 1. Cuando el administrador solicita adicionar un ejercicio para el autoexamen, se le mostrara un formulario con los tres tipos de ejercicios empleados y sus respectivos niveles de dificultad, donde debe seleccionar un tipo en particular y una vez realizada esta operación debe entrar los datos correspondientes. Al concluir se actualizan los ejercicios de preparación. 2. Cuando el administrador solicita modificar un ejercicio de los de autoexamen se le mostraran los tres tipos de ejercicios de los cuales por tipo de ejercicio y por nivel de dificultad debe seleccionar el que se desea modificar, una vez aceptada esta operación se actualiza el ejercicio mostrándose con los cambios realizados. 3. Cuando el administrador solicita eliminar un ejercicio de autoexamen. Se muestra un mensaje para confirmar la operación, una vez aceptada, se elimina y se muestra el listado actualizado. <p>El caso de uso termina en cualquiera de los pasos anteriores y es válido para usuarios</p>	

que tengan estos privilegios.	
Referencias	18, 18.1, 18.1.1, 18.1.2, 18.1.3, 18.2, 18.2.1, 18.2.2, 18.2.3, 18.3, 18.3.1, 18.3.2, 18.3.3
Prototipo	
Ver anexo 30	

Caso de Uso	Mostrar trazas
Actores	Administrador, Usuario con privilegios, Usuarios
Propósito	Mostrar el transcurso de un usuario en el sistema en cuanto a sus evaluaciones
Resumen	
Se inicia cuando un administrador desea ver los resultados obtenidos por los evaluados desde que comenzaron a trabajar en el sistema, una vez aceptada la opción se le muestra un formulario con cada una de las evaluaciones realizadas por estudiantes, incluyendo la fecha y los ejercicios realizados. El caso de uso termina una vez terminada esta operación y es válido para todos los usuarios exceptuando los invitados.	
Referencias	11
Prototipo	
Ver anexo 44	

Caso de Uso	Mostrar reportes
Actores	Administrador, Usuarios con privilegios, Usuarios, Invitados
Propósito	Mostrar información de usuarios y del sistema
Resumen	
Se inicia cuando un administrador desea ver resultados de autoexámenes de estudiantes o conocer datos específicos de la aplicación, una vez seleccionada esta operación se les mostraran los diferentes tipos de reportes que pueden ser de interés, de los cuales puede además de verlos imprimirlo o simplemente guardarlos. El caso de uso termina una vez concluido este paso y es válido para cualquier usuario.	
Referencias	22
Prototipo	

Ver anexo 41

Caso de Uso	Gestionar Usuario.
Actores	Administrador, Usuario con privilegios
Propósito	Controlar el nivel de acceso de los usuarios a realizar cambios en el sistema.
Resumen	
Se inicia cuando el administrador o un usuario que posea este tipo de privilegio desea asignarle a un usuario determinado un rol con una determinada funcionalidad; para lo cual se le muestra un formulario con todos los usuarios y todas las acciones que pueden ser realizadas como es el caso también de la opción que permite desactivar una cuenta de usuario del sistema; una vez terminada las acciones efectuadas se actualiza el sistema y termina el caso de uso.	
Referencias	(17)
Prototipo	
Ver anexo 42	

Caso de Uso	Cambiar contraseña
Actores	Administrador, Usuarios con privilegios, Usuarios
Propósito	Modificar la contraseña para entrar al sistema.
Resumen	
El caso de uso comienza cuando un usuario del sistema desea cambiar su contraseña, se le muestra un formulario para que escriba la nueva contraseña y una vez aceptada la operación se actualiza el sistema y culmina el caso de uso.	
Referencias	3
Prototipo	
Ver anexo 27	

Caso de Uso	Autenticarse.
Actores	Administrador, Usuarios con privilegios, Usuarios
Propósito	Entrar al sistema por mediación de una cuenta.
Resumen	

Se inicia cuando un determinado individuo desea entrar, hacer uso del sistema y quiere hacerlo por su cuenta ya previamente creada; se le muestra un formulario donde debe escribir su nombre de cuenta y su contraseña, una vez aceptada esta opción se verifican los datos y se le abre la aplicación con todos sus privilegios como usuario del sistema. El caso de uso termina una vez concluida esta operación y es válida para todos los usuarios exceptuando aquellos para los cuales no existe una cuenta creada.	
Referencias	2
Prototipo	
Ver anexo 33	

Caso de Uso	Crear cuenta de usuario.
Actores	Administrador, Usuario con privilegios, Usuario, Invitado
Propósito	Crear una cuenta para aquellas personas que la soliciten.
Resumen	
El caso de uso comienza cuando cualquier persona que entre al sistema quiera formar parte del mismo; selecciona la opción de crear cuenta y se le muestra un formulario con todos los datos que debe entrar; una vez aceptada esta operación se crea la cuenta nueva por la cual el usuario ya puede entrar al sistema y termina el caso de uso	
Referencias	1
Prototipo	
Ver anexo 29	

Caso de Uso	Configuración del sistema.
Actores	Administrador
Propósito	Determinar el número de ejercicios con que contará el autoexamen; así como la dirección del fichero donde se guardaran los artículos del sistema.
Resumen	
El caso de uso se inicia cuando el administrador desea configurar el sistema de manera que los estudiantes a la hora de hacer el autoexamen respondan tantas	

preguntas como el administrador desee; así como ubicar la dirección de la carpeta donde se guardaran los artículos del sistema una vez seleccionada esta operación se le mostrará un formulario con los datos que deben ser llenados, al terminar se actualiza el sistema y termina el caso de uso	
Referencias	16
Prototipo	
Ver anexo 28	

Caso de Uso	Mostrar ayuda.
Actores	Administrador, Usuario con privilegios, Usuarios, Invitados
Propósito	Aclarar cualquier duda que pueda tener el usuario del sistema indicándole que hace el sistema en cada una de sus partes
Resumen	
Se inicia cuando cualquier persona desea conocer alguna información del funcionamiento del sistema una vez aceptada esta operación se le mostrará al usuario todos los contenidos que posee el sistema; así como la forma de uso del mismo una vez terminado lo anteriormente explicado termina el caso de uso.	
Referencias	23
Prototipo	
Ver anexo 26	

Caso de Uso	Entrar novedad.
Actores	Administrador, Usuario con privilegios, Usuarios, Invitados
Propósito	Suministrarle al sistema toda aquella información que puede ser novedosa para las personas que lo utilizan.
Resumen	
El caso de uso se inicia cuando cualquier persona que se encuentre trabajando en el sistema desee incorporarle algo nuevo y novedoso, una vez aceptada esta opción se le mostrará un formulario con los datos que deben ser llenados; al aceptar el sistema guarda la novedad y culmina el caso de uso.	
Referencias	20
Prototipo	

Ver anexo 35

Caso de Uso	Mostrar listado de usuarios.
Actores	Administrador, Usuario con privilegios, Usuarios.
Propósito	Mostrar todos aquellos usuarios que forman parte del sistema
Resumen	
Se inicia cuando cualquier usuario desea conocer cuáles son las personas que se encuentran registradas en el sistema; una vez aceptada esta opción se le mostrará un formulario con la lista de todos los usuarios del sistema, culminando así el caso de uso.	
Referencias	24
Prototipo	
Ver anexo 34	

2.8 Conclusiones.

En este capítulo se hizo un estudio detallado del problema a tratar, permitiendo una mejor comprensión de la dinámica, estructura de la organización y del problema a resolver por el software propuesto. Esto facilitará la eficiencia del producto final.

El estudio facilitó la definición de los requerimientos del sistema, el cual presenta 60 requisitos funcionales y 11 requisitos no funcionales, con una explicación de los mismos.

Se definieron finalmente 4 actores del sistema asociados a 24 casos de uso, que fueron agrupados por paquetes de acuerdo a las funcionalidades del sistema.

Capítulo 3. Construcción de la solución propuesta.

3.1 Introducción.

El diseño forma parte del proceso de desarrollo de software y su función es decidir cómo el sistema se llevará a cabo. En el diseño, se toman decisiones para cumplir los requerimientos funcionales y de calidad. Mediante el diseño se le da respuesta a la pregunta de cómo diseñar.

En este capítulo se moldean los resultados de la etapa del diseño del sistema, empleando UML para su modelado. Se describe la puesta en práctica de la construcción de la solución propuesta, dada su descripción en el capítulo anterior. Se presenta el diagrama de clases del diseño. Se plantea también el modelo de datos a partir del diagrama las clases persistentes, el diagrama de componentes y se hace referencia a los estándares de diseño y de programación que se tuvieron en cuenta.

3.2 Diagrama de clases.

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

Ver diagramas de clases desde el anexo 1 hasta el 23.

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 capaces de mantener su valor en el espacio y en el tiempo.

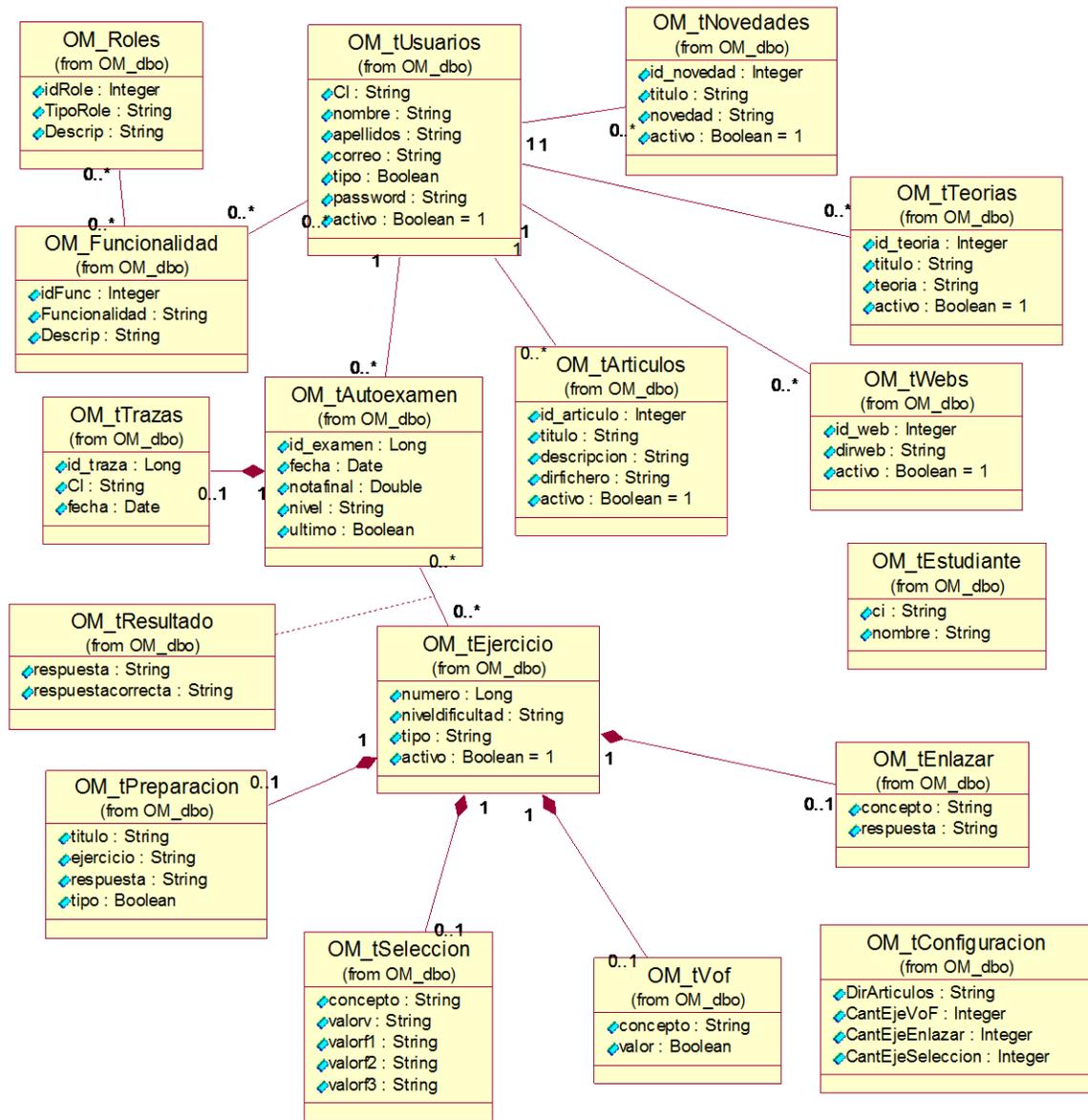


FIGURA 9 DIAGRAMA DE CLASES PERSISTENTES.

3.3.2 Modelos de datos.

La calidad de la interfaz de usuario puede ser uno de los motivos que lleve a un sistema al éxito o al fracaso, es por ello que uno de los aspectos más significativos de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

La interfaz estará diseñada siguiendo el estándar de ventanas de Windows. El tipo de letra a utilizar será Arial, de estilo regular y tamaño 8. Se empleará un lenguaje de fácil comprensión para los usuarios en las diferentes opciones. El icono asociado a la aplicación será el logotipo que identificará al sistema. Los mensajes de error e informativos que se mostrarán serán cortos y fáciles de entender.

3.5 Concepción general de la ayuda.

Para hacer más fácil el empleo del tutorial, este constará con una ayuda que estará disponible al usuario en todo momento y constará con una explicación funcional. La ayuda contará además con explicaciones detalladas de cómo funciona cada parte del tutorial, explicará el funcionamiento de cada ventana, con sus opciones y operaciones. En la descripción de cada una de las partes se utilizan imágenes que facilitan la comprensión de lo que se está explicando.

3.6 Tratamiento de errores.

En el tutorial presentado se tratarán los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que se gestiona. El usuario tiene muy pocas posibilidades de entrar información errónea, se conservará un nivel de validación de la información y en caso de errores se le comunicará el error a través de mensajes de error. Los mensajes de error que emitirán, se mostrarán en un lenguaje de fácil comprensión para los usuarios y fueron utilizados la clase de excepciones de Delphi.

3.7 Estándares de codificación.

Los estándares de codificación en cualquier aplicación son de suma importancia, por facilitar el mantenimiento del tutorial, garantizar un código claro, y reducir los errores que se puedan presentar durante el desarrollo del mismo. Los aspectos que logran hacer un código más legible son; empleo de nombres descriptivos, uso de una identificación coherente y de comentarios

informativos. Se describen a continuación convenciones tomadas con relación a estos aspectos para el código escrito en Object Pascal en el sistema.

Los nombres de las variables, los controles, los procedimientos y funciones fueron adaptados 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. El comentario brinda información de una manera clara y ayuda a entender el código. Se tomó como regla, comentar todos aquellos procedimientos y funciones que puedan resultar complejos. Se comentan además algunos algoritmos que pudieran resultar de difícil comprensión. 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.

3.8 Diagrama de componentes.

Un diagrama de componentes ilustra un conjunto de componentes y sus relaciones.

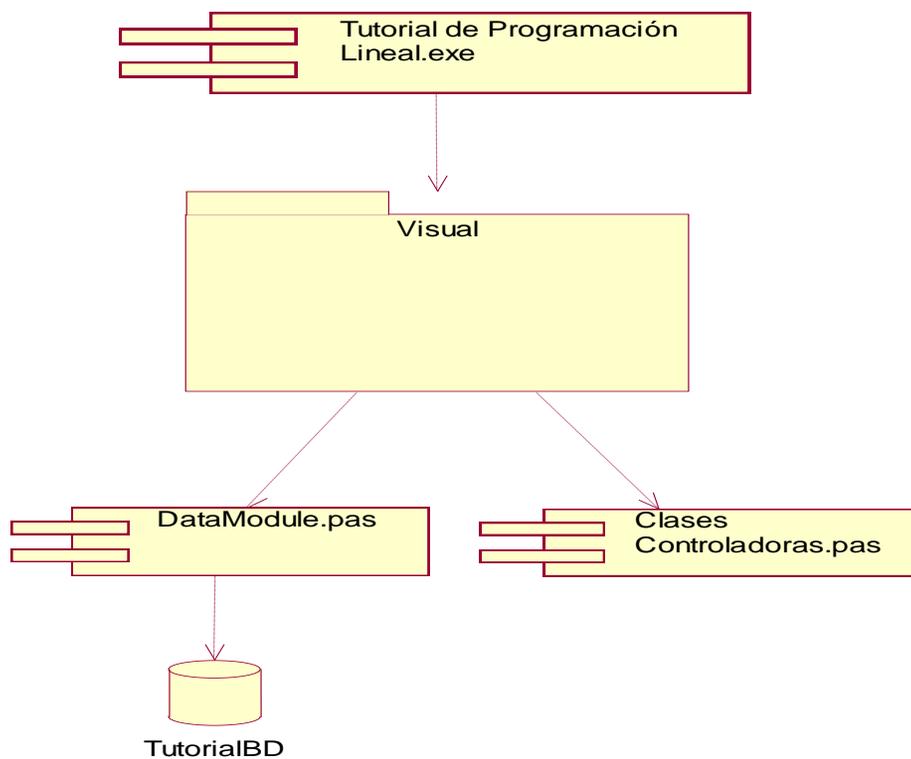


FIGURA 11 DIAGRAMA DE COMPONENTES.

3.9 Modelo de despliegue.

En el diagrama de despliegue se señala la situación física de los componentes lógicos desarrollados. Expresado de otra forma se sitúa el software en el hardware que lo contiene.

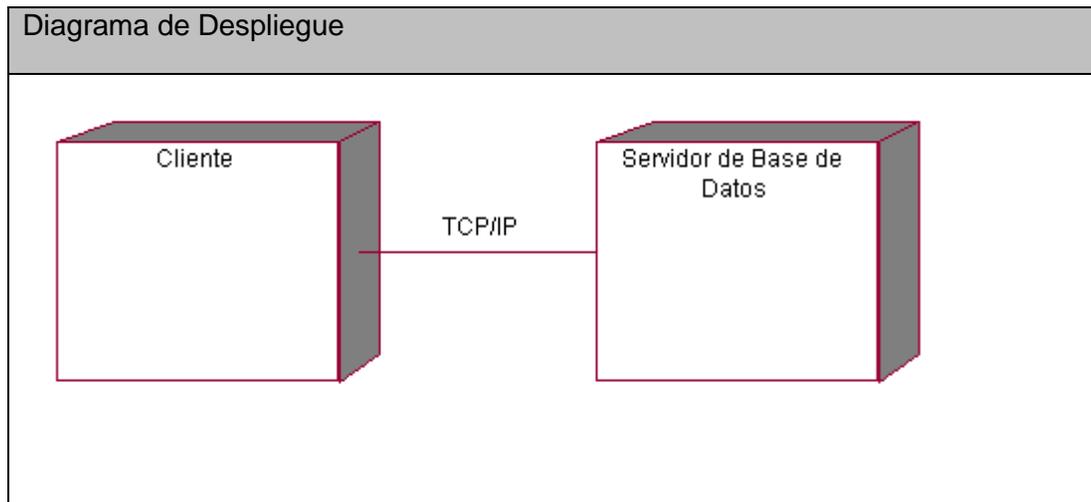


FIGURA 12 DIAGRAMA DE DESPLIEGUE.

3.10 Conclusiones.

En el 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.

TABLE 3 ENTRADAS EXTERNAS

Nombre de la entrada externa	Cant.de ficheros	Cant.de elem. de datos	Clasificación
Crear Usuario	1	7	Simple
Modificar Usuario	1	5	Simple
Eliminar Usuario	1	7	Simple
Insertar Teoría	1	3	Simple
Modificar Teoría	1	3	Simple
Eliminar Teoría	1	3	Simple
Insertar Ejercicio Resuelto	1	4	Simple
Modificar Ejercicio Resuelto	1	4	Simple
Eliminar Ejercicio Resuelto	1	4	Simple
Insertar Ejercicio Propuesto	1	5	Simple
Modificar Ejercicio Propuesto	1	5	Simple

Eliminar Ejercicio Propuesto	1	5	Simple
Insertar Sitio de Internet	1	2	Simple
Modificar Sitio de Internet	1	2	Simple
Eliminar Sitio de Internet	1	2	Simple
Insertar Ejercicio de Verdadero o Falso	1	3	Simple
Modificar Ejercicio de Verdadero o Falso	1	3	Simple
Eliminar Ejercicio de Verdadero o Falso	1	3	Simple
Insertar Ejercicio de Selección Simple	1	7	Simple
Modificar Ejercicio de Selección Simple	1	7	Simple
Eliminar Ejercicio de Selección Simple	1	7	Simple
Insertar Ejercicio de Enlace	1	4	Simple
Modificar Ejercicio de Enlace	1	4	Simple
Eliminar Ejercicio de Enlace	1	4	Simple
Insertar Novedad	1	3	Simple
Modificar Novedad	1	3	Simple
Eliminar Novedad	1	3	Simple
Insertar Artículos	1	4	Simple
Eliminar Artículos	1	4	Simple
Total	Simple:29	Media: 0	Compleja: 0

Salidas externas.

TABLE 4 SALIDAS EXTERNAS

Nombre de la salida externa	Cant. de ficheros	Cant. elem. de datos	de de	Clasificación
Teoría	1	3		Simple
Ejercicios Preparación	1	4		Simple
Novedades	1	3		Simple
Páginas Web	1	3		Simple
Artículos	1	5		Simple
Configuración General del Sistema	1	10		Simple
Cantidad Exámenes Usuario	1	5		Simple
Cantidad Exámenes por Nivel	1	6		Simple
Ultimo Examen	1	5		Simple
Total	Simple:9	Media: 0		Compleja: 0

Ficheros internos.

TABLE 5 FICHEROS INTERNOS

Nombre del fichero interno	Cant. de records	Cant. elem. de datos	de de	Clasificación
Funcionalidad	1	3		Simple
Roles	1	3		Simple
Roles_Func	1	2		Simple
TArtículos	1	6		Simple
TAutoexamen	1	6		Simple

TConfiguración	1	4	Simple
TEjercicios	1	4	Simple
TEnlazar	1	3	Simple
TEstudiante	1	2	Simple
TNovedades	1	5	Simple
TPreparación	1	5	Simple
TResultado	1	4	Simple
TSelección	1	6	Simple
TTeorías	1	5	Simple
TTrazas	1	4	Simple
TUsuarios	1	7	Simple
TVoF	1	3	Simple
TWebs	1	4	Simple
Usuario_Role_Func	1	3	Simple
Total	Simple:19	Media: 0	Compleja: 0

Puntos de Función sin ajustar.

TABLE 6 PUNTOS DE FUNCIÓN SIN AJUSTAR

Elementos	S	X Peso	M	X Peso	C	X Peso	PF. Subtotal
Ficheros lógicos internos	19	(*7)	0	(*10)	0	(*15)	133
Ficheros de interfaces externas	0	(*5)	0	(*7)	0	(*10)	0
Entradas externas	29	(*3)	0	(*4)	0	(*6)	87

Elementos	S	X Peso	M	X Peso	C	X Peso	PF. Subtotal
Salidas externas	9	(*4)	0	(*5)	0	(7)	36
Peticiones	0	(*3)	0	(*4)	0	(6)	
Total							256

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.

Instrucciones Fuentes.

TABLE 7 INSTRUCCIONES FUENTES

Características	Valor	
Puntos de función desajustados	256	
Lenguaje	Object Pascal	SQL
% de utilización en la aplicación	70% (≈ 179)	30% (≈ 77)
Instrucciones fuentes por puntos de función	29	39
Instrucciones fuentes	5191	3003
Total Instrucciones fuentes (LOC)	8194	

4.3 Costos.

Multiplicadores de esfuerzo:

TABLE 8 MULTIPLICADORES DE ESFUERZO

Multiplicador	Descripción	Valor

RCPX	Considerando el énfasis en la documentación, la complejidad del producto y el tamaño de la base de dato se determinó que el nivel de este multiplicador es nominal.	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 nominal para este multiplicador.	1.00
PERS	La capacidad de los especialistas (analistas-programadores) es nominal.	1
PREX	La experiencia en este tipo de aplicación es alta.	0.87
FCIL	Se utilizan herramientas e instrumentos de programación modernos.	1.10
SCED	Se estima que el producto se hará en el tiempo establecido.	1.00
		1.02

7

$$EM = \prod_{i=1}^7 EM_i = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 1.02$$

i=1

Factores de Escala:

TABLE 9 DEFINICIÓN DE LOS VALORES DE LOS FACTORES DE ESCALA (SFI)

Factor	Descripción	Valor
PREC	Aspectos novedosos.	2.48
FLEX	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las	3.04

	especificaciones de interfaz externa.	
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

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

$$i=1$$

Valores calibrados:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

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

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.02 - 0.91) = 0.262$$

Cálculo del esfuerzo (PM):

$$PM = A * (MSLOC)^E * \Pi Emi = 2.94 * (8.194)^{1.02} * 1.02 = 25 \text{ Hombres/Mes.}$$

Cálculo del tiempo de desarrollo:

$$TDEV = C * PM^F = 3.67 * (25)^{0.262} = 8.52 \approx \mathbf{9 \text{ meses (Estimado)}}$$

Cálculo de la cantidad de hombres:

$$CH = PM / TDEV = 25/9 \approx \mathbf{3 \text{ hombres}}$$

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

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

$$TEDV = PM/CH^* = 25/1 = \mathbf{25 \text{ meses.}}$$

Cálculo del costo:

Asumiendo como salario promedio mensual (SP) \$275.00

$$CHM = CH * SP = 1 * \$275.00 = \$275.00$$

$$\text{Costo} = CHM * PM = \$275 * 25 = \$ 6875$$

- Cálculos

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

TABLE 10 CÁLCULO DEL ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO

Cálculo de:	Valor
Esfuerzo (PM: Hombres - mes)	25 Hombres/Mes
Tiempo de Desarrollo(meses)	25 Meses
Cantidad de Hombres	1
Costo	\$ 6875

Salario medio	\$ 275.00
---------------	-----------

4.4 Beneficios tangibles e intangibles.

Beneficios intangibles.

Los beneficios que se lograrán con el desarrollo del sistema propuesto son principalmente intangibles, dado a que la aplicación diseñada es para la educación; permitiendo a los usuarios ampliar sus conocimientos sobre la Programación Lineal, más particularmente en la modelación matemática del problema.

Disminuirá el tiempo que se invierte en buscar información sobre la temática pues en el tutorial aparecerán muchas teorías y artículos que aborden el tema; así como sitios Web vinculados con la programación lineal.

4.5 Análisis de costos y beneficios.

La utilización de este nuevo sistema para apoyar la enseñanza y aprendizaje de la modelación matemática en el proceso de preparación de los estudiantes en el tema programación lineal, permitirá disminuir el tiempo en el proceso de búsqueda de información para el estudio de la temática. Se contará con un mayor control sobre el estado actual de los conocimientos de los estudiantes, los cuales se verán beneficiados por todo lo que pueden aprender en el tutorial dado a que el mismo muestra todas aquellas explicaciones teóricas necesarias para el buen entendimiento de la materia estudiada.

Con la información que brindará el sistema, se podrá buscar de forma rápida todos los contenidos que pueden aportar conocimiento del tema, así como conocer, mediante reportes el desarrollo alcanzado por parte de los usuarios.

Analizando el costo del proyecto que es de \$ 6875.00 contra los numerosos beneficios que reportará, detallados con anterioridad; y la necesidad, de desarrollar una herramienta para apoyar el proceso de enseñanza y aprendizaje de la formulación matemática en el tema programación lineal, se considera factible el desarrollo del mismo.

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 25 hombre/mes, para un tiempo de desarrollo de 25 meses, para una persona desarrollándolo, el costo que implica es de \$6875.00, calculado para un salario mensual de \$275.

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

Conclusiones.

1. Se evidenció, a través del análisis bibliográfico que, el desarrollo de tutoriales donde se emplean las TIC, da lugar a herramientas informáticas que apoyan los procesos de enseñanza-aprendizaje en diferentes entornos. La enseñanza de la formulación matemática de un problema de programación lineal, debido a complejidades propias, precisa de medios y herramientas que la apoyen.
2. Se realizó el diseño de un tutorial para apoyar el proceso de enseñanza y aprendizaje de la formulación matemática, de un problema de programación lineal, siguiendo la metodología RUP y teniendo en cuenta los requisitos funcionales y no funcionales. El diseño se hizo de forma tal, que cada una de las pantallas de la aplicación fueran lo mas explicativas posibles, fáciles de usar por los usuarios y buscando la mayor interactividad.
3. El estudio de la factibilidad del sistema propuesto, arrojó como resultado que los beneficios que aporta a la enseñanza y aprendizaje de la formulación matemática en el tema de programación lineal son mayores que los costos en que se incurre. Por tanto, resulta factible su implementación.
4. Se implementó un tutorial para apoyar el proceso de enseñanza y aprendizaje de la formulación matemática, de un problema de programación lineal, utilizando la herramienta de programación Borland Delphi 2006 y SQL Server 2000.

Recomendaciones.

1. Incorporar nuevos tipos de preguntas para el autoexamen de modo que enriquezca la diversidad de su realización.
2. incluir en el tutorial opciones de estímulos para los estudiantes que obtengan buenos resultados en los autoexámenes.

Bibliografía.

- 227:13 UNESCO: *Proyecto general de educación para América Latina y el Caribe*, Pedagogía 2003, La Habana, 2003.
- 44:45 Castellanos, Doris y otros: *El proceso de enseñanza-aprendizaje desarrollador en la Secundaria Básica*. Centro de Estudios Educativos. I.S.P. "E. J. Varona". Ciudad de la Habana, 2001.
- 43:125 Castellanos, Doris y otros: *Aprender y enseñar en la escuela: una concepción desarrolladora*, Ciudad de la Habana, 2002.
- 187:15 Prieto, Daniel: "Mediación pedagógica y Nuevas tecnologías", *Nuevas tecnologías aplicadas a la Educación Superior*, # 1, ICFES. ARTE Y FOTOLITO "ARFO" LTDA, Santafé de Bogotá, 1995, ISBN: 958-9279-18-X.
- 33:49 Blanco, Lázaro: "Hiperuniversidad: nueva alternativa educativa", *GIGA: La revista cubana de Computación*, Colombus conectividad, 3, 48-51, Ciudad de la Habana, 1999, ISBN 1028-270x
- 225:5 Twigg, Carol : "The changing Definition of learning", *Educom Review*, Julio-Agosto, 1994.
- 58:81 Colom, A, Jesús Salinas, y J. Sureda: *Tecnología y medios educativos*, Cincel-kapelusz, Madrid, 1988
- 95:6 Forneiro, Rolando, Victoria Arencibia y René Hernández: "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, 2002.
- 77:18 Delgado, Mayra: *Tesis presentada en opción al Título de Master en Informática Educativa*, Ciudad de la Habana, 2002.
- 216:92 Schank, Roger: *Engines for Education*, 1995.

Alejo Machado, O. (2006). *Portal de Psicología de la Universidad de Cienfuegos*.

Almeida Campos, S. (2009). *Evolución de la enseñanza asistida por computadoras*.

Castillo Lorenzo, J. (2008). Software educativos ., (p. 6).

Conociendo algo de software educativo. (2009, diciembre 25). Retrieved enero 3, 2010, from Conociendo algo de software educativo: <http://www.enlaces.udec.cl>

Cordero Carrasco, R. (n.d.). Retrieved febrero 20, 2010, from [http://www.nielsoft.com/Seminario/3 capas](http://www.nielsoft.com/Seminario/3%20capas)

(n.d.). Retrieved abril 20, 2010, from <http://www.programlenguaje.delphi.com>

Alonso Moles, E. (2007). *Software Educativo para enseñar a utilizar e implementar los Servicios de Redes*.

Batista Pacheco, A. L. (2008). *Medios para apoyar la enseñanza primaria*.

Conociendo algo de software educativo. (2009, diciembre 25). Retrieved enero 3, 2010, from Conociendo algo de software educativo: <http://www.enlaces.udec.cl>

García Suarez, L., & López Delgado, L. (2007). *Medios para aprender a aprender*.

Gómez Martínez, I. (2009). *Medios de transmisión de las información*.

Hombre y naturaleza. (n.d.). Retrieved diciembre 26, 2009, from <http://www.ense.tlcp.com>

Jímenes Valle, Y. (2008). *Soluciones para mejorar los medios educativos*.

Felipe, P. (1987). Aspectos fundamentales de la programación lineal. In P. Felipe, *Programación Matemática 1* (p. 69). Pueblo y Educación.

Felipe, P. (1987). Construcción de la función objetivo. In P. Felipe, *Programación Matemática 1* (p. 85). Pueblo y Educación.

Felipe, P. (1987). Construcción del modelo de programación lineal. In P. Felipe, *Programación Matemática 1* (p. 78). Pueblo y Educación.

Felipe, P. (1987). Construcción del sistema de restricciones. In P. Felipe, *Programación Matemática 1* (p. 82). Pueblo y Educación.

- Felipe, P. (1987). Definición de las Variables de decisión. In P. Felipe, *programación matemática 1* (pp. 78-79). pueblo y educación.
- Marquez, P. (2010). *El software educativo*. Universidad Autónoma de Barcelona, Barcelona.
- Plasencia Rodríguez, M. (2005). *Sistema Generador de Foros de Discusión*.
- Rodríguez Terreno, N. (n.d.). Retrieved marzo 24, 2010, from <http://www.elguille.info/colabora/net2005/aplicaciones3capas.htm>
- Rosario, J. (2005). *La Tecnología de la Informática y las Comunicación(TIC)*.
- Soler Castillo, F. (2008). *La investigación sobre medios de enseñanza: Pasado y Presente*.
- Toledo Osorio, Y. (2009, noviembre 5). *Monografías*. Retrieved diciembre 15, 2009, from <http://www.Monografías.com>
- Verano Catoní, M. L. (2009, enero 25). *Software y Educación*. Retrieved febrero 2010, 12, from <http://blogs.rimed.cu/infoedu/category/software-educativo/>
- (n.d.). Retrieved abril 20, 2010, from <http://www.programlenguaje.delphi.com>
- (n.d.). Retrieved abril 20, 2010, from <http://www.programlenguaje.delphi.com>
- Alonso Moles, E. (2007). *Software Educativo para enseñar a utilizar e implementar los Servicios de Redes*.
- Batista Pacheco, A. L. (2008). *Medios para apoyar la enseñanza primaria*.
- Caraballosa Rafael, Y. (2007). *Tutorial para la enseñanza-aprendizaje de los contenidos de SQL Server en las universidades pedagógicas de Sancti Spíritus*.
- Conociendo algo de software educativo*. (2009, diciembre 25). Retrieved enero 3, 2010, from *Conociendo algo de software educativo*: <http://www.enlaces.udec.cl>
- Echemendia Sierra, J. D. (2007). *Multimedia "Microsoft Access más fácil" para contribuir a desarrollar la*.
- García Suarez, L., & López Delgado, L. (2007). *Medios para aprender a aprender*.
- Gómez Martínez, I. (2009). *Medios de transmisión de las información*.
- Hombre y naturaleza*. (n.d.). Retrieved diciembre 26, 2009, from <http://www.ense.tlcp.com>

Jímenes Valle, Y. (2008). *Soluciones para mejorar los medios educativos*.

Osiel Martínez, V. (2007). *Diseño de un Tutorial para contribuir al aprendizaje del tema clasificación y manejo de malezas en la carrera de ingeniería agronómica*.

Perez Perez, J. C. (2007). *TUTORIAL PARA APOYAR EL APRENDIZAJE DE LOS ESTUDIANTES EN LA ASIGNATURA INFORMÁTICA I PERTENECIENTE A LA TAREA ÁLVARO REINOSO DEL MINAZ*.

Rodríguez Borges, T. (2007). *MULTIMEDIA "LA EVOLUCIÓN DE LAS TELECOMUNICACIONES EN SANCTI SPIRITUS" PARA CONTRIBUIR A LA PREPARACIÓN CULTURAL DE LOS TRABAJADORES*.

Castells M. *The Rise of the Network Society*. Cambridge. Mass: Blackwell; 1996.

Cabero, J Duarte, A. (1994): CD-ROM en la enseñanza e investigación: una tecnología en aumento, Pixel.bit, nº1, pp. 83-101.

BELLOCH; 2000. Disponible en: <http://cfv.uv.es/belloch>.

García,D;López,E. Sistema tutor para la enseñanza del Algebra Lineal. Revista Ingeniería Industrial. Vol XII.No 2. 1991.Cuba Pag 69.

Rodríguez,R. Informe final sobre Trabajo de Investigación; Búsqueda y estudio de los sistemas tutoriales para la importación de la computación. Escuela Militar de Pilotos. 1991. Cuba.

Microsoft ® Encarta ® 2006. © 1993-2005 Microsoft Corporation. Reservados todos los derechos.

Fernández Manjón, Baltasar: Sistema de ayuda inteligente para entornos informáticos complejos, 2000.

Rodríguez Lamas, Raúl y otros: Introducción a la Informática Educativa. Cuba, 2000.p.46.

Matos, Rosa María. *Introducción al trabajo con Base de Datos*. Asignatura Sistemas de Gestión de Base de Datos.

Manual de SQL. www.lobocom.es/~claudio.

Introducción a SQL. www.lafacu.com/apuntes/informatica/sql

SQL. www.arsys.es/soporte/programacion.

Manual de SQL. <http://walter.freesevers.com>

Díaz Antón Maria Gabriela, Angélica Pérez Maria, C. Grimmán Anna, E. Mendoza Luis. *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática.* <http://www.academia-interactiva.com/ise.pdf>

Ferrá Grau, Xavier. *Desarrollo orientado a objetos con UML.* Universidad

Politécnica de Madrid. <http://www.clikear.com/manuales/uml/introduccion.asp>,

Lenguaje Unificado de Modelamiento. <http://www.creangel.com/uml/intro.php>

Conociendo algo de software educativo. MODULO 3. Centro Zonal Sur - Area Pedagógica. [en línea]. (accedido el 16 de agosto de 2006). Disponible en: <http://www.enlaces.udec.cl/centrozonal-sur/pedagogica/Modulo3.pdf>.

Gómez, Luz del Carmen; Moncayo, José de Jesús. La Importancia de Promover en el Aula Estrategias de Aprendizaje para elevar el Nivel Académico en los Estudiantes. [on line]. (accedido el 20 de diciembre de 2006). Disponible en: <http://www.congreso.unam.mx/ponsemloc/ponencias/1224.html>

Bravo Romero, Silvia. (2004). "El Mapa Conceptual como estrategia de enseñanza y aprendizaje en la resolución de problemas". Facultad de Química. Universidad de la Habana.

Jacobson, I. "El Proceso Unificado de Desarrollo de software" / I Jacobson .-- EUA:Addison-Wesley, 2000.--p.132.

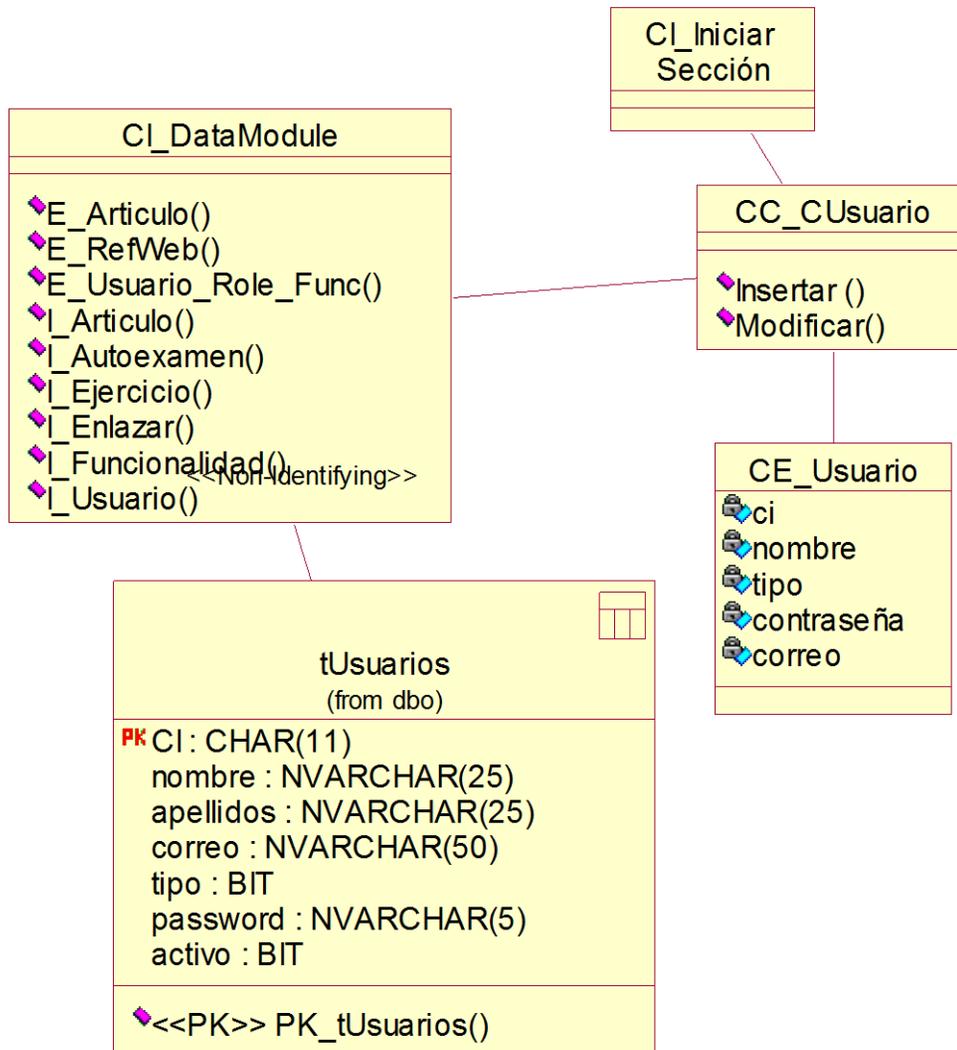
Ruíz Bravo, Danaysi. Software para la elaboración de Mapas Conceptuales/ Danaysi Ruíz Bravo; Boris Piñero Suárez; Alfredo J. Simón Cuevas, tutor. —Trabajo deDiploma; ISPJAE (C.H), 2004. --p.105.

Lauro, Soto. "Especificación de requerimientos" / Soto Lauro. Tomado de: <http://mitecnologico.com/Main/EspecificacionesDeRequerimientos>, 4 de marzo del 2006.

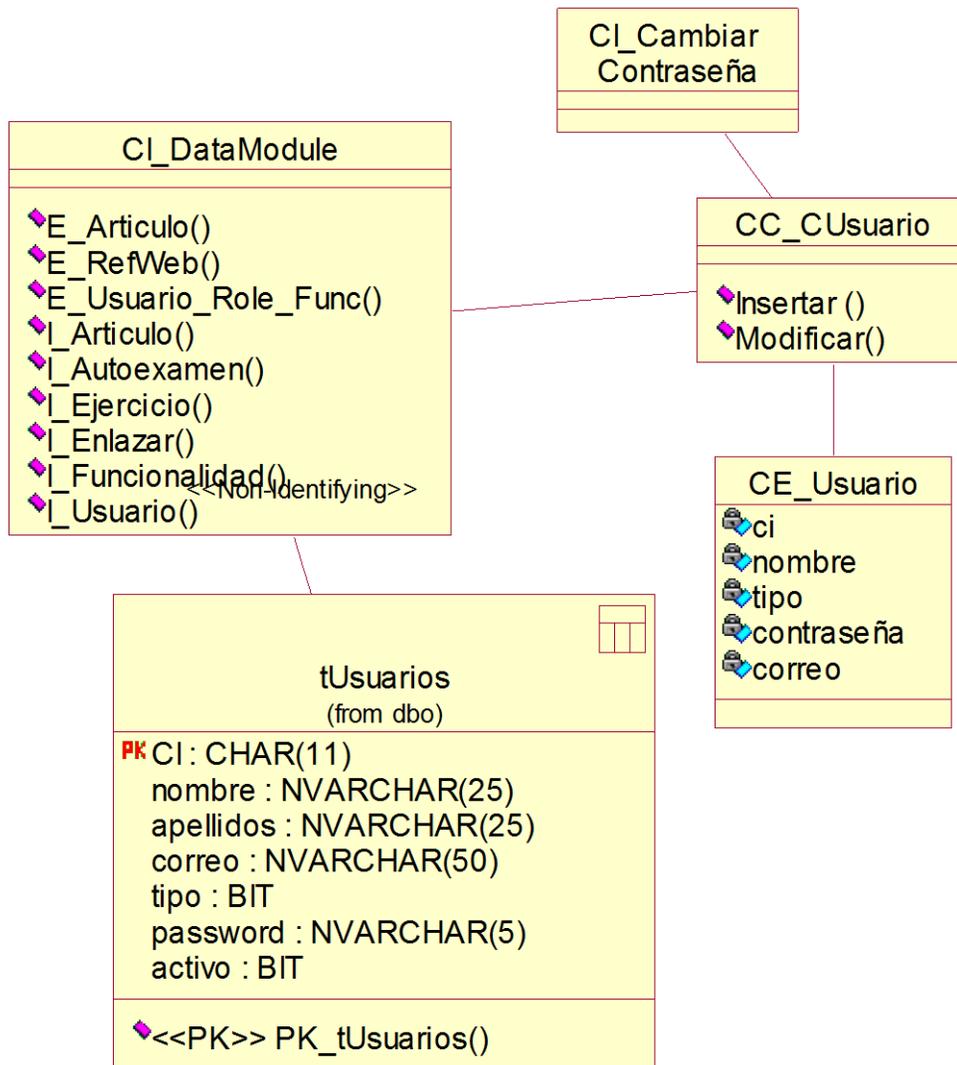
Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software / Ivar Jacobson;Grady Booch; Jame Rumbaugh.-- La Habana: Editorial Félix Varela, 2004.Tomo I y II.

Anexos

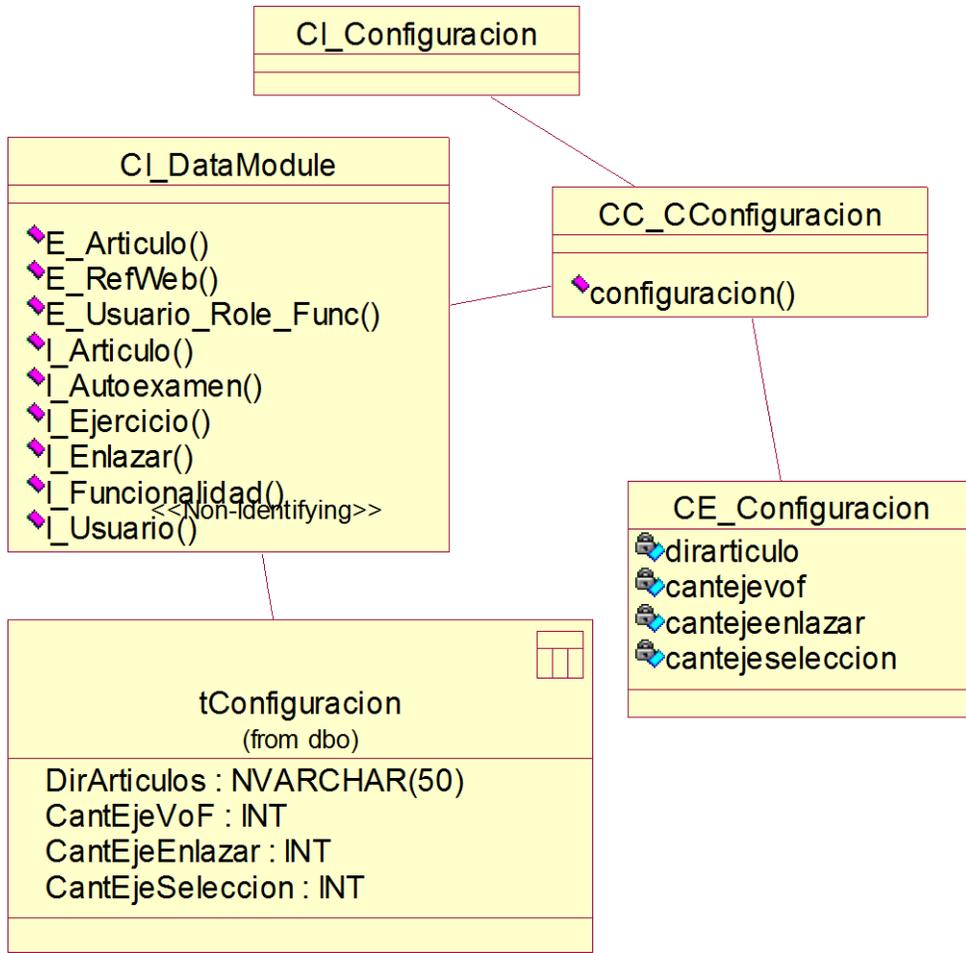
Anexo 1: Diagrama de clases. Caso de uso autenticarse.



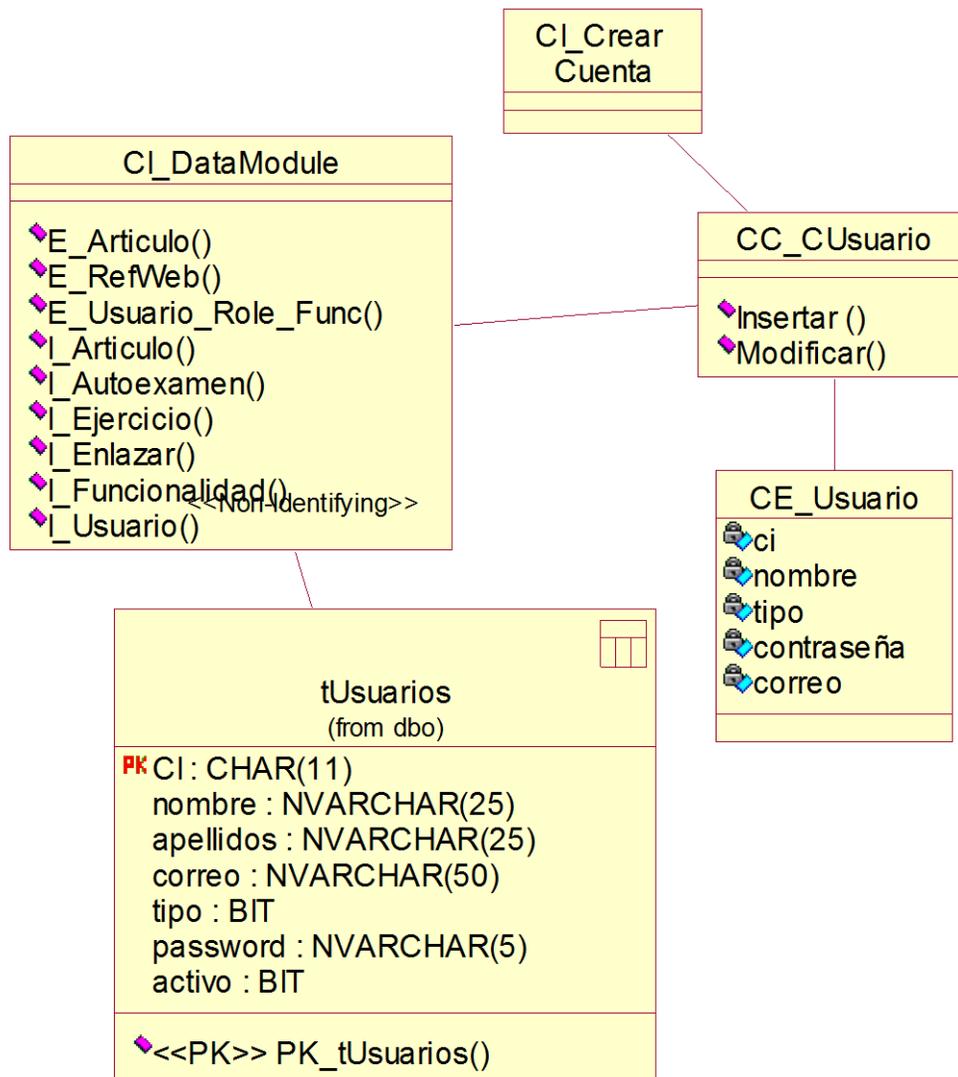
Anexo 2: Diagrama de clases. Caso de uso cambiar contraseña.



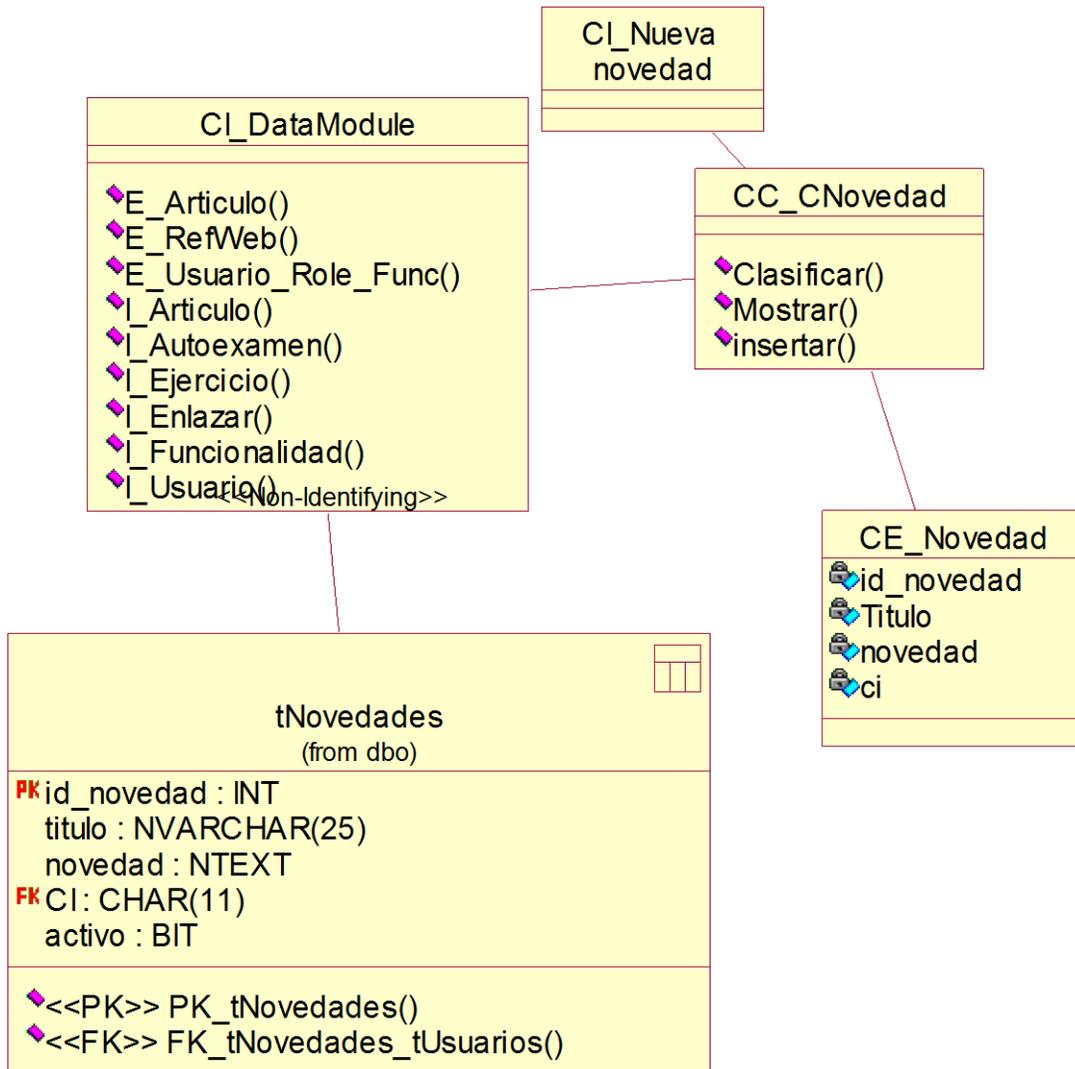
Anexo 3: Diagrama de clases. Caso de uso configurar el sistema.



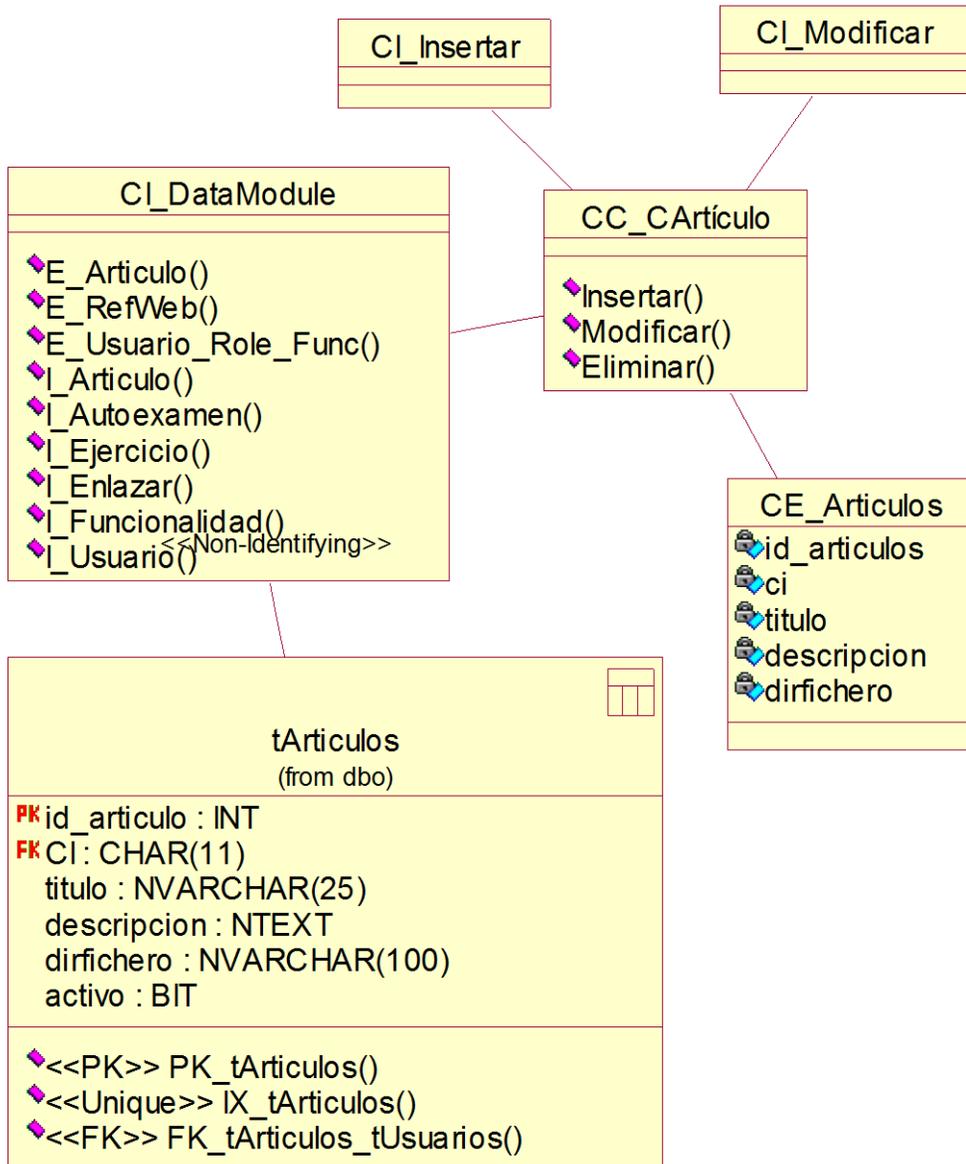
Anexo 4: Diagrama de clases. Caso de uso crear cuenta de usuario.



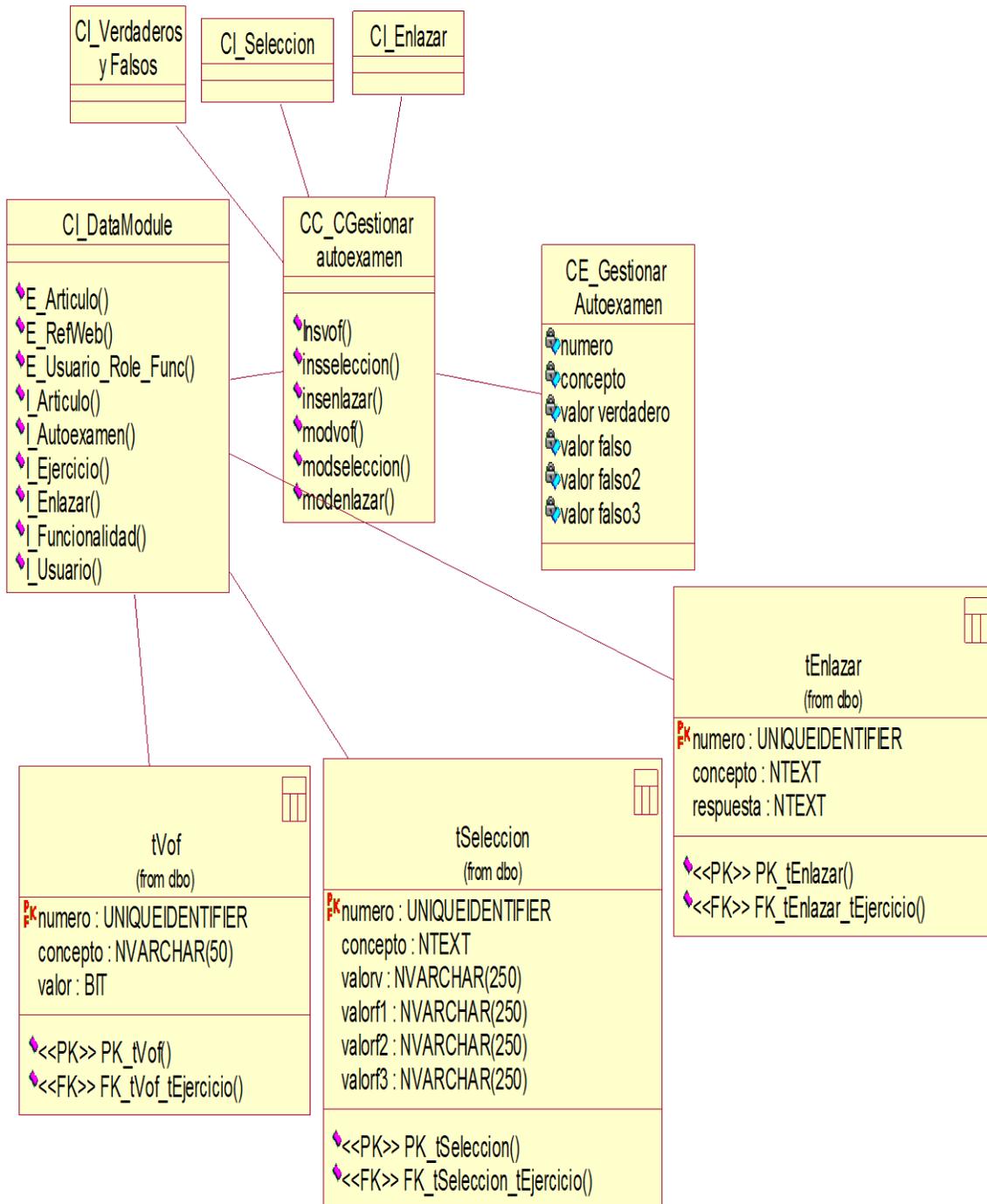
Anexo 5: Diagrama de clases. Caso de uso entrar novedad.



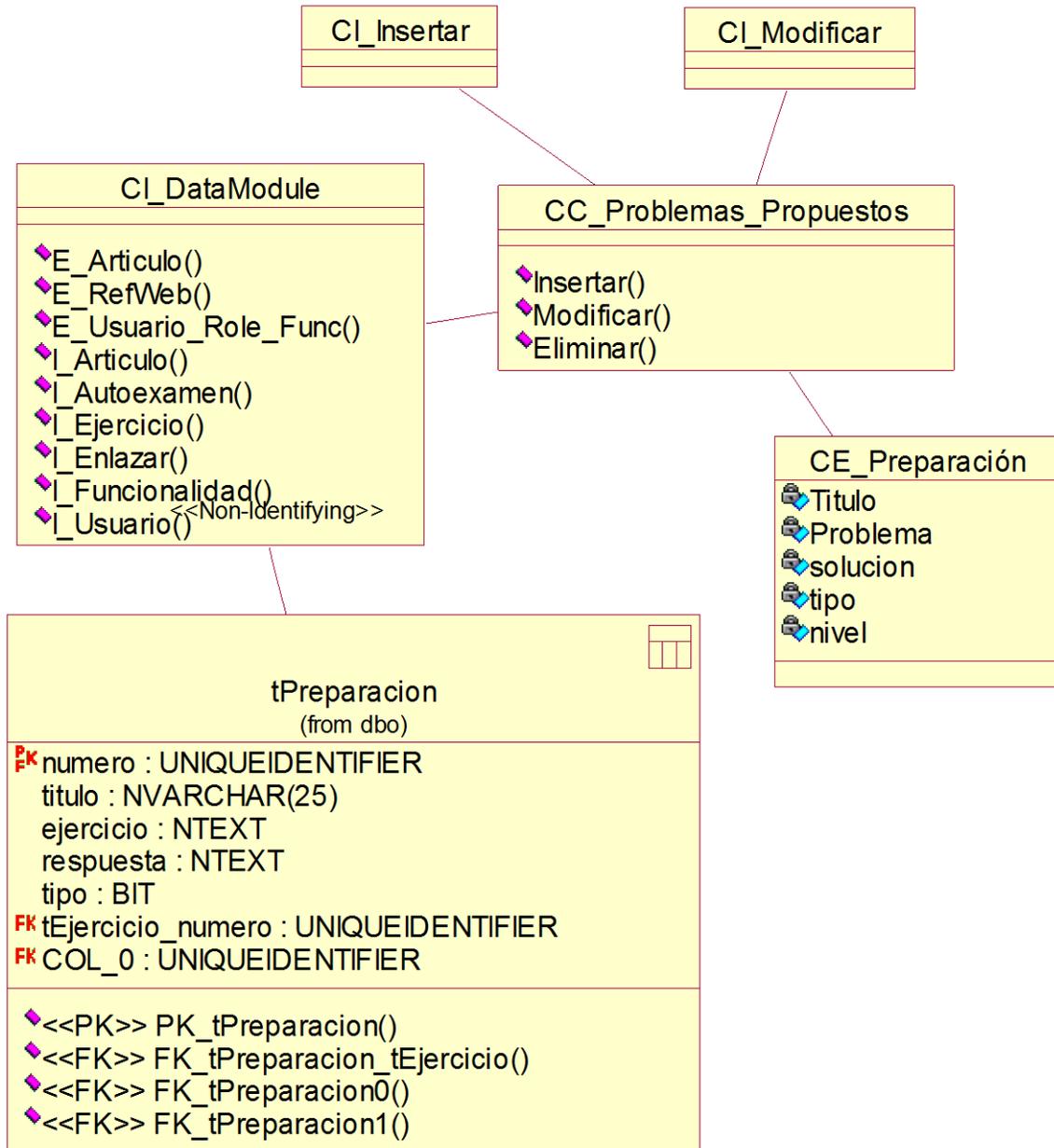
Anexo 6: Diagrama de clases. Caso de uso gestionar artículo.



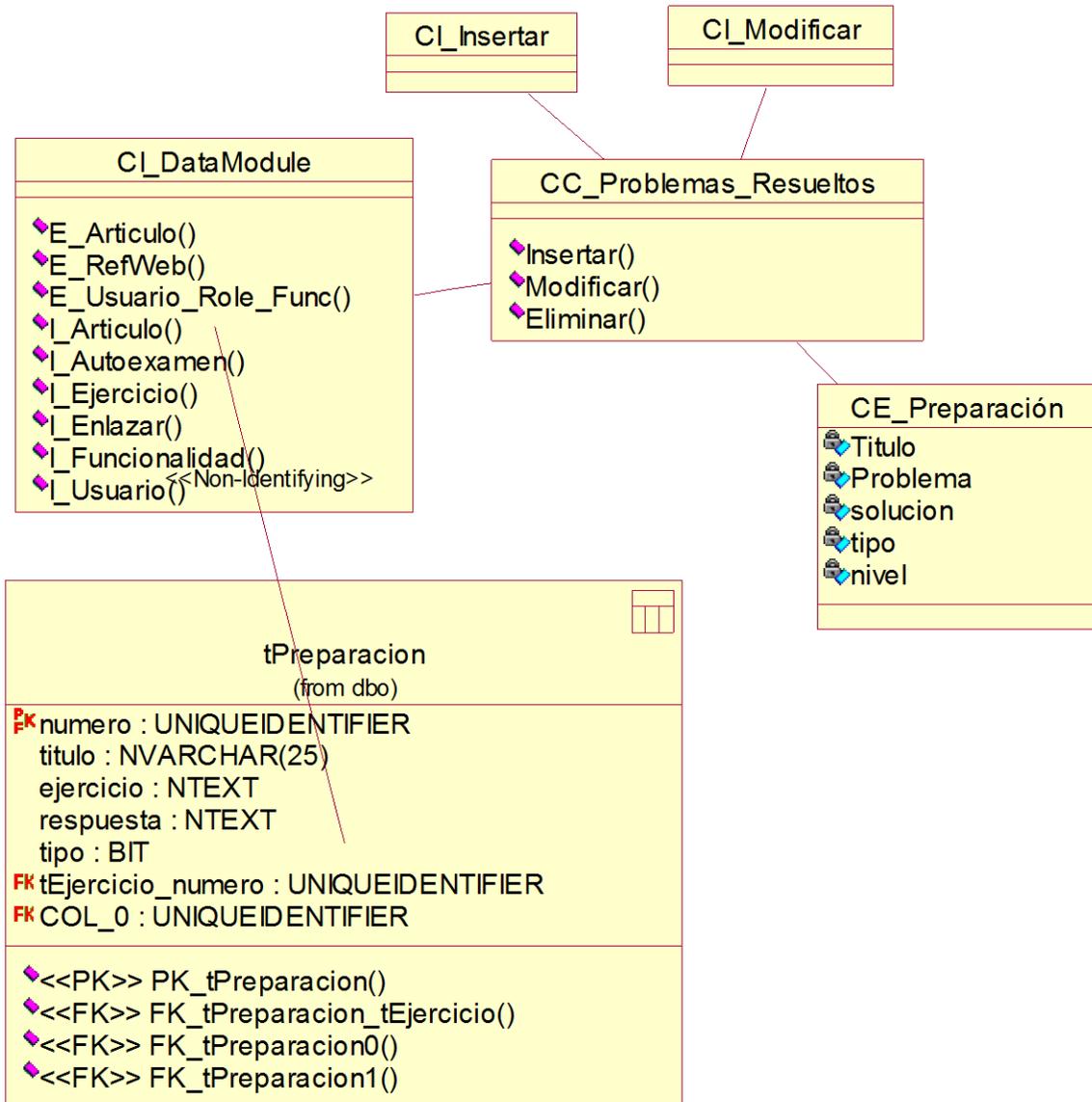
Anexo 7: Diagrama de clases. Caso de uso gestionar ejercicios de autoexamen.



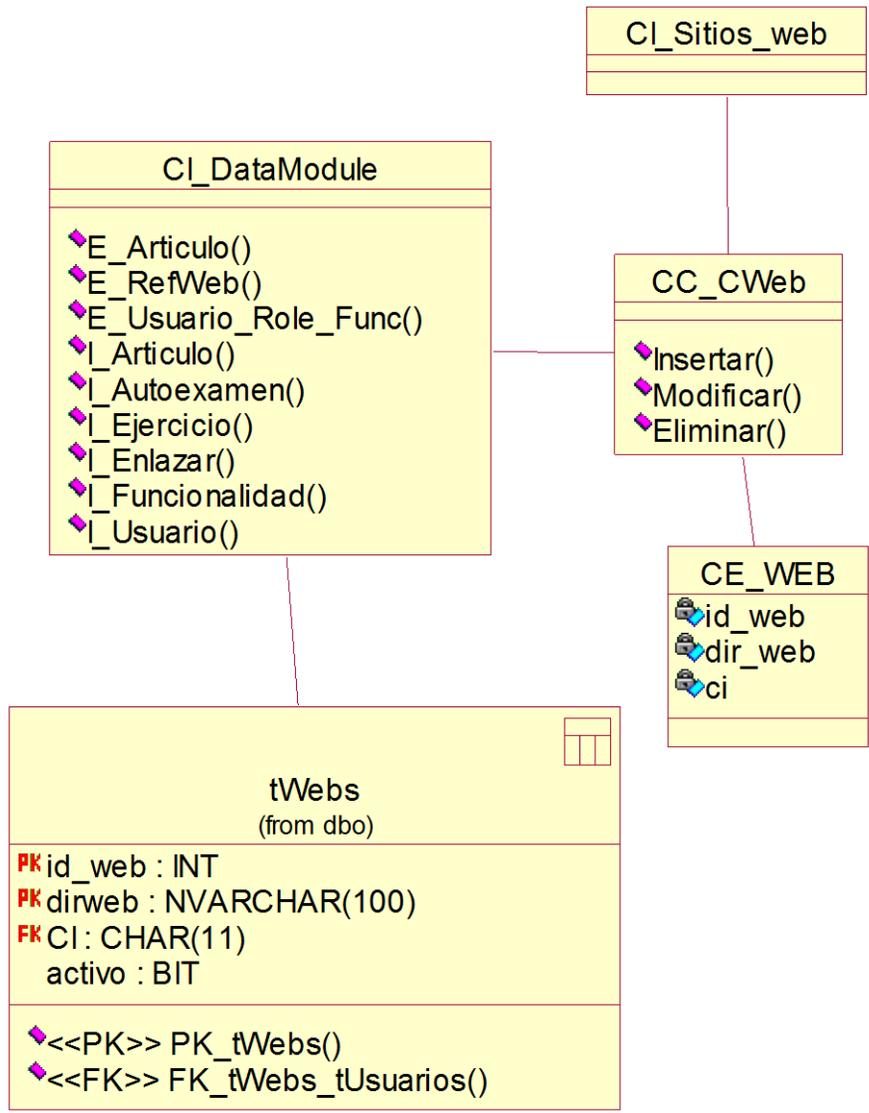
Anexo 8: Diagrama de clases. Caso de uso gestionar problemas propuestos.



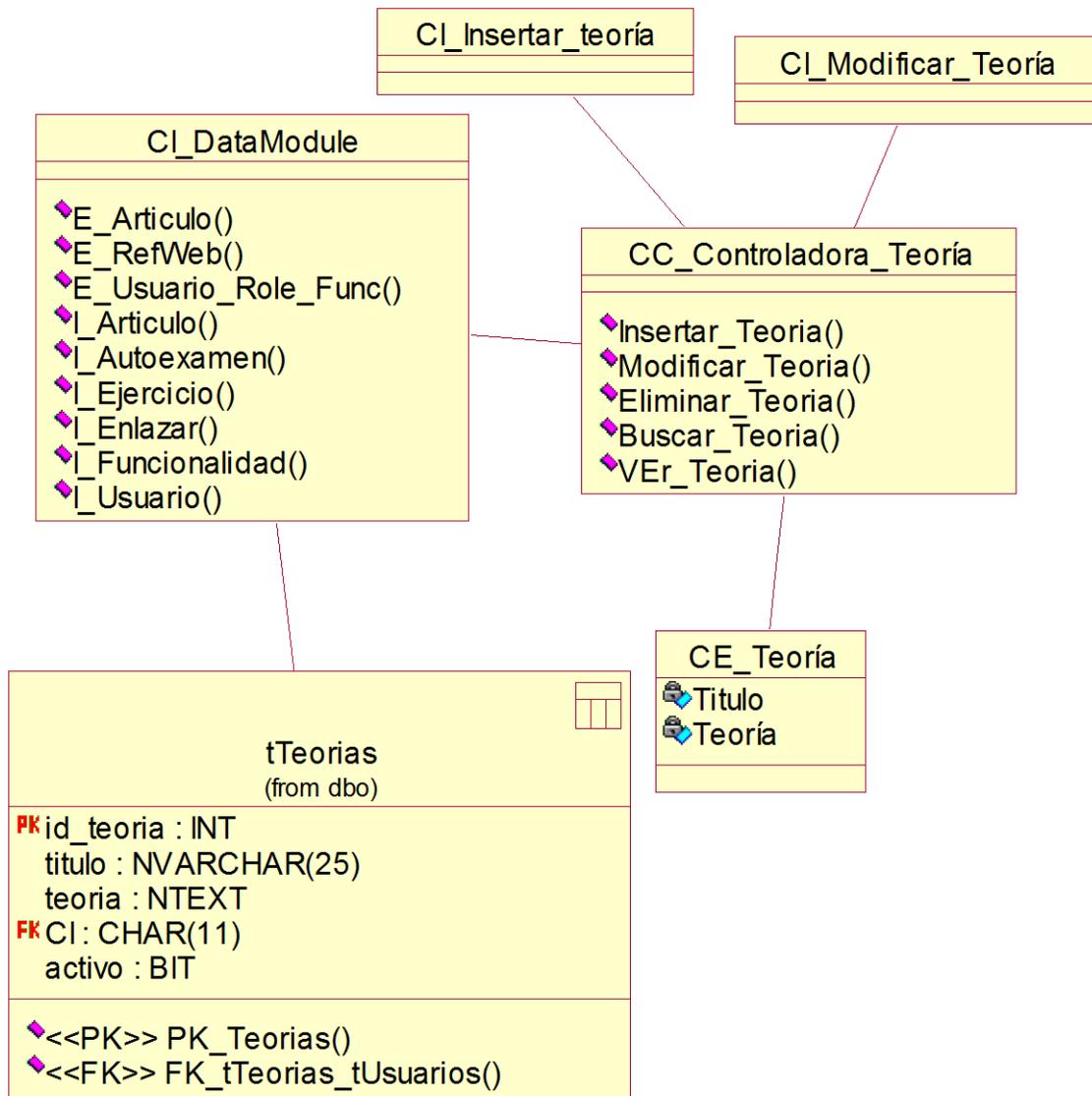
Anexo 9: Diagrama de clases. Caso de uso gestionar problemas resueltos.



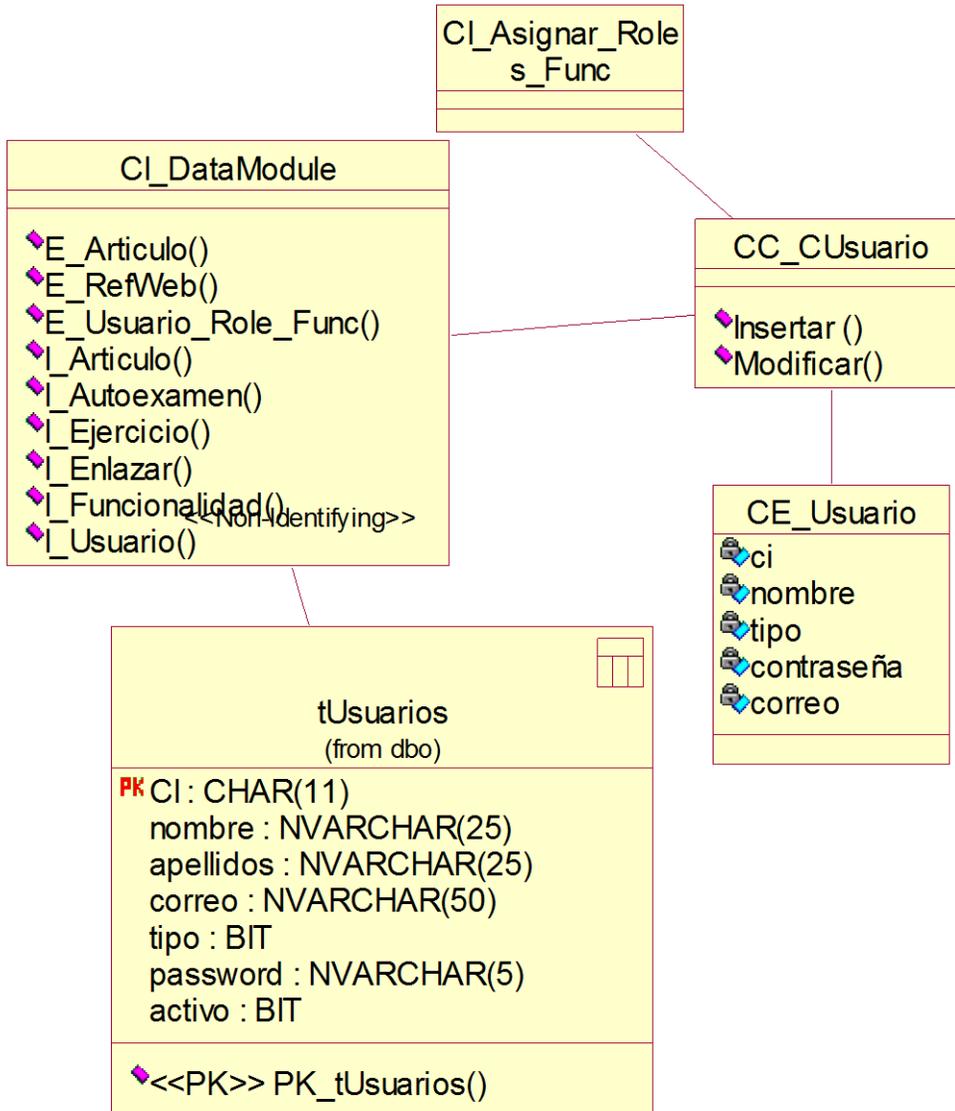
Anexo10: Diagrama de clases. Caso de uso gestionar web.



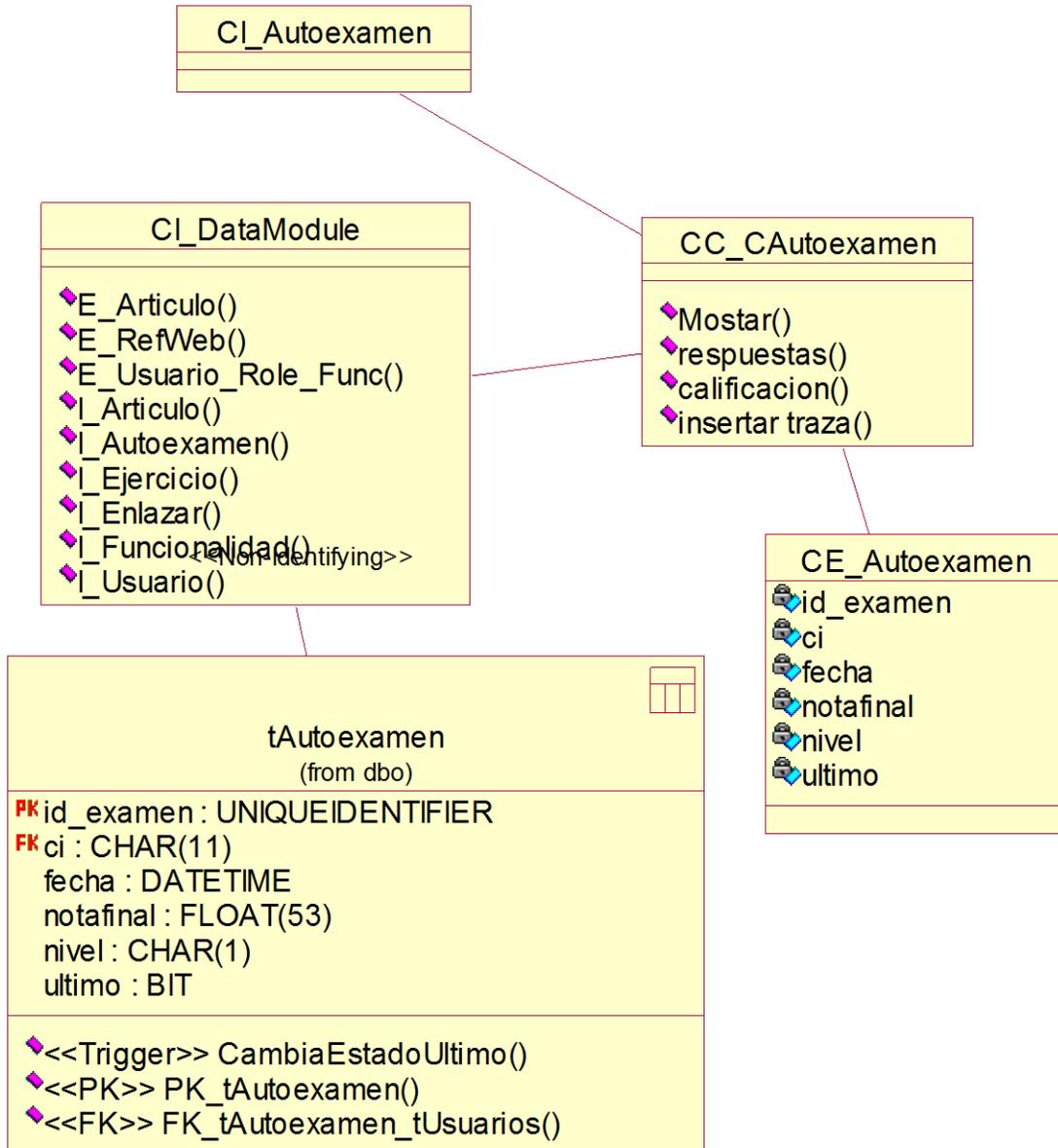
Anexo 11: Diagrama de clases. Caso de uso gestionar teoría.



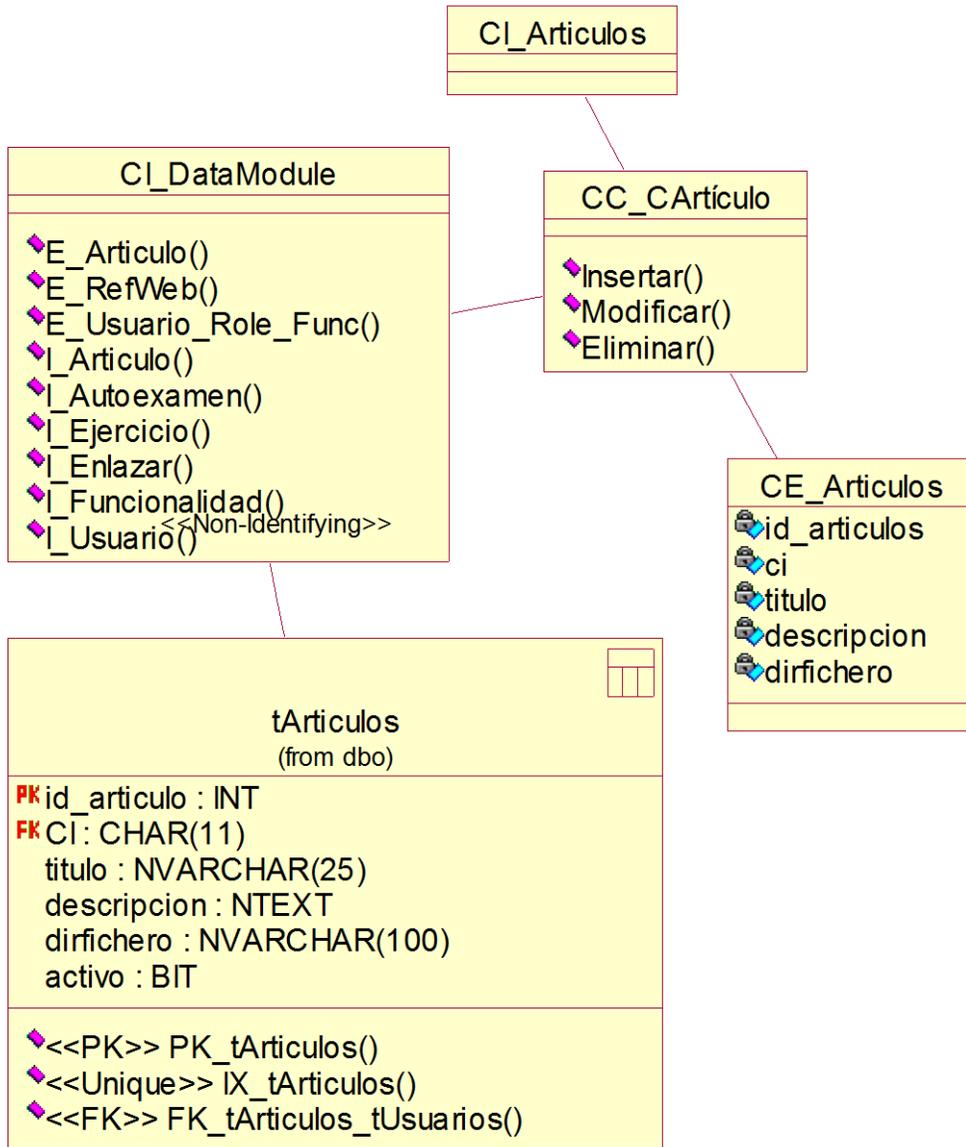
Anexo 12: Diagrama de clases. Caso de uso gestionar usuario.



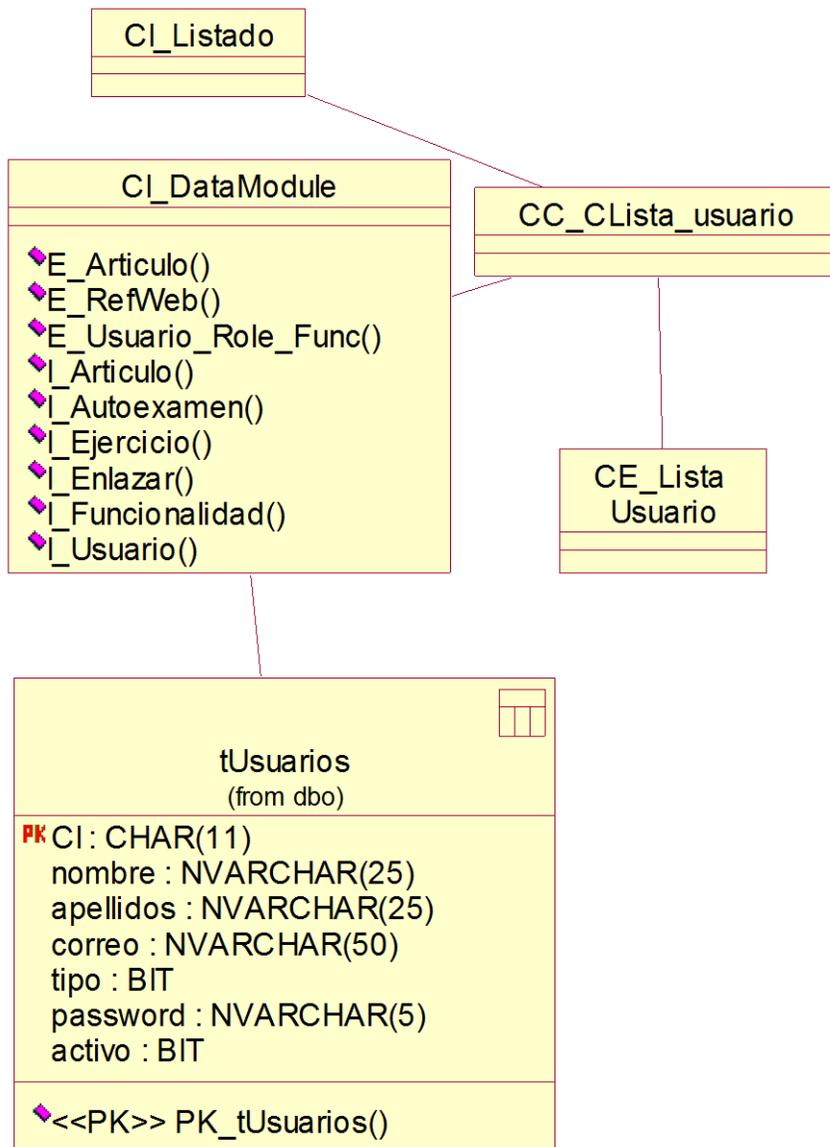
Anexo 13: Diagrama de clases. Caso de uso hacer autoexamen.



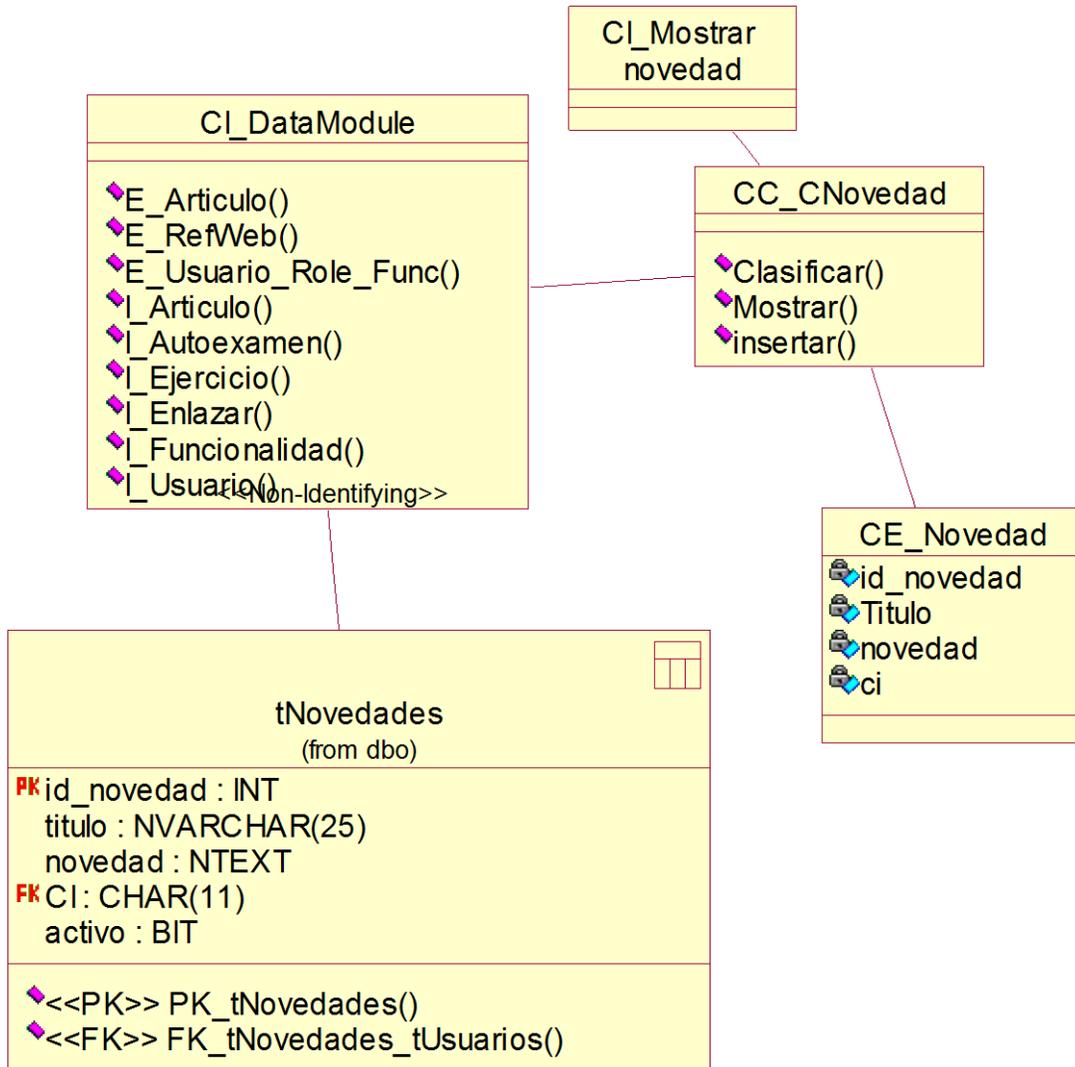
Anexo 14: Diagrama de clases. Caso de uso mostrar artículos.



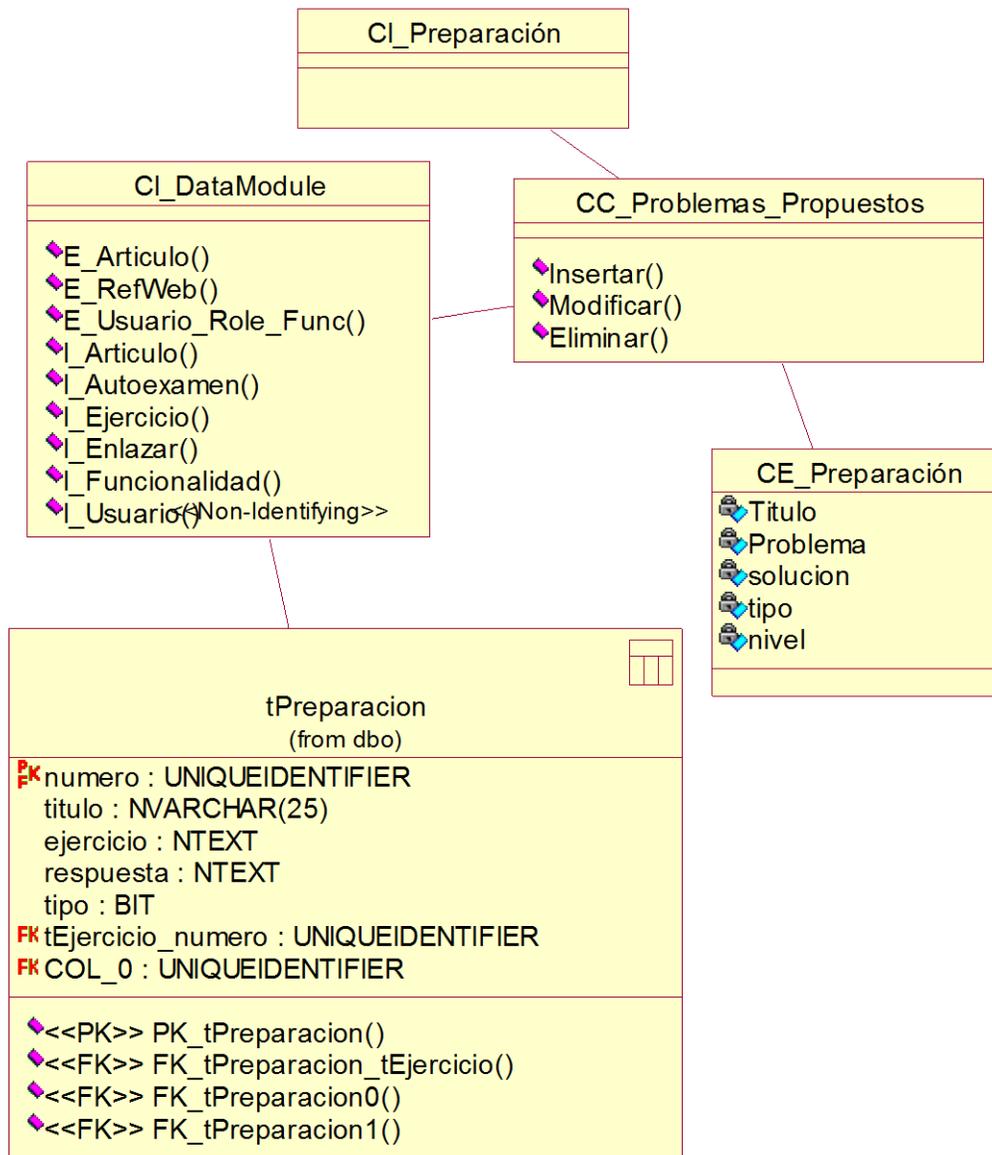
Anexo 15: Diagrama de clases. Caso de uso mostrar listado de usuarios.



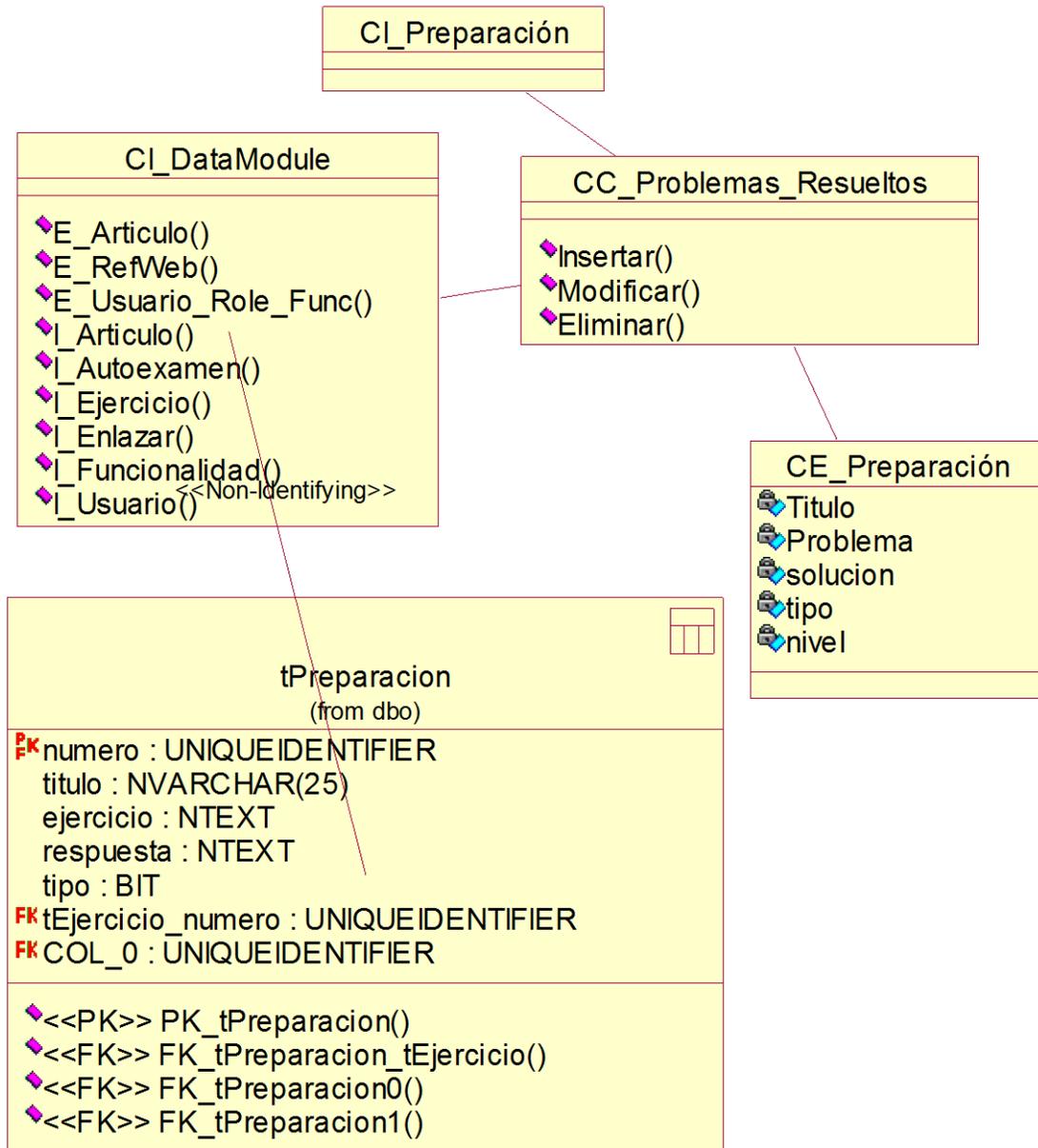
Anexo 16: Diagrama de clases. Caso de uso mostrar novedad.



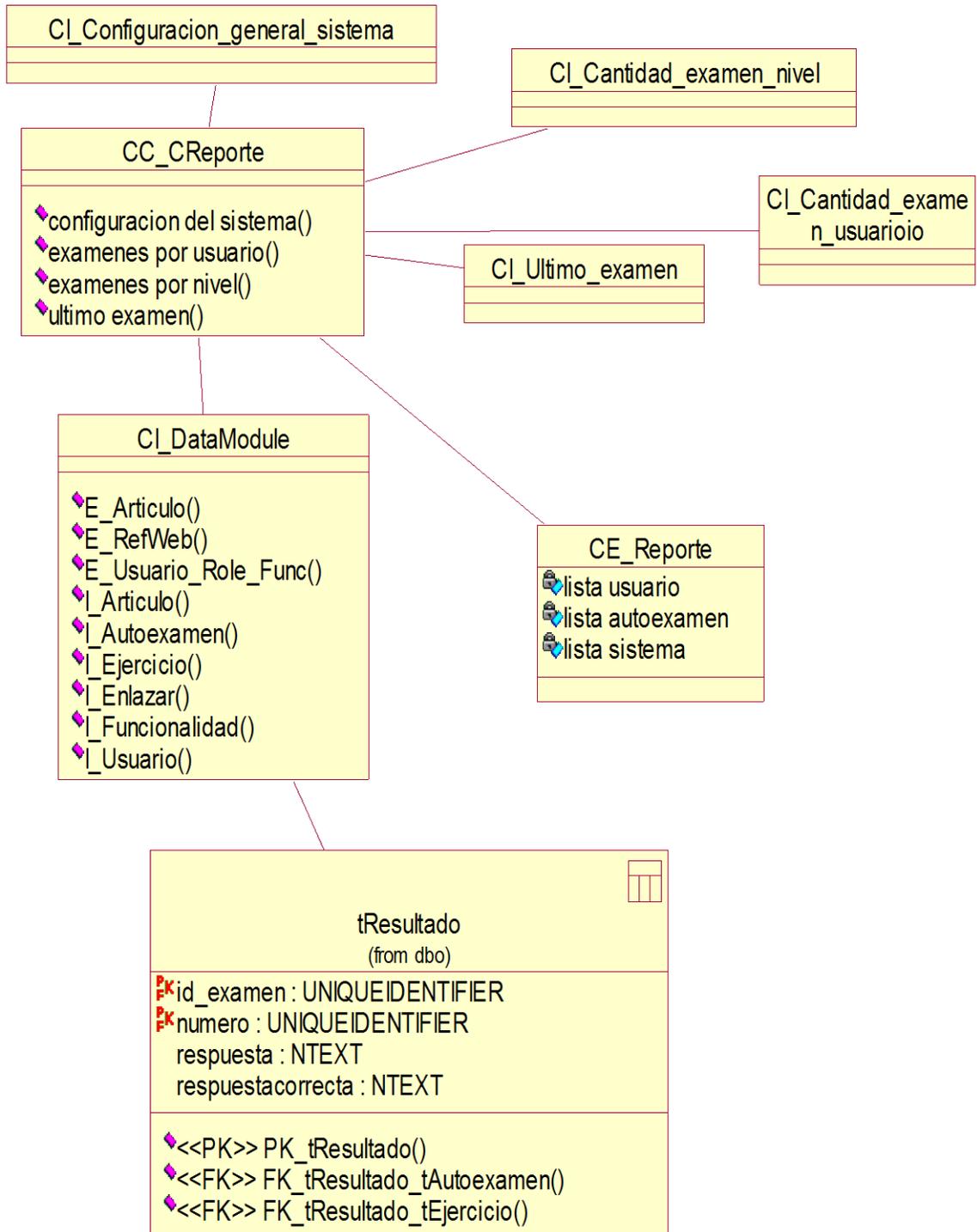
Anexo 17: Diagrama de clases. Caso de uso mostrar problemas propuestos.



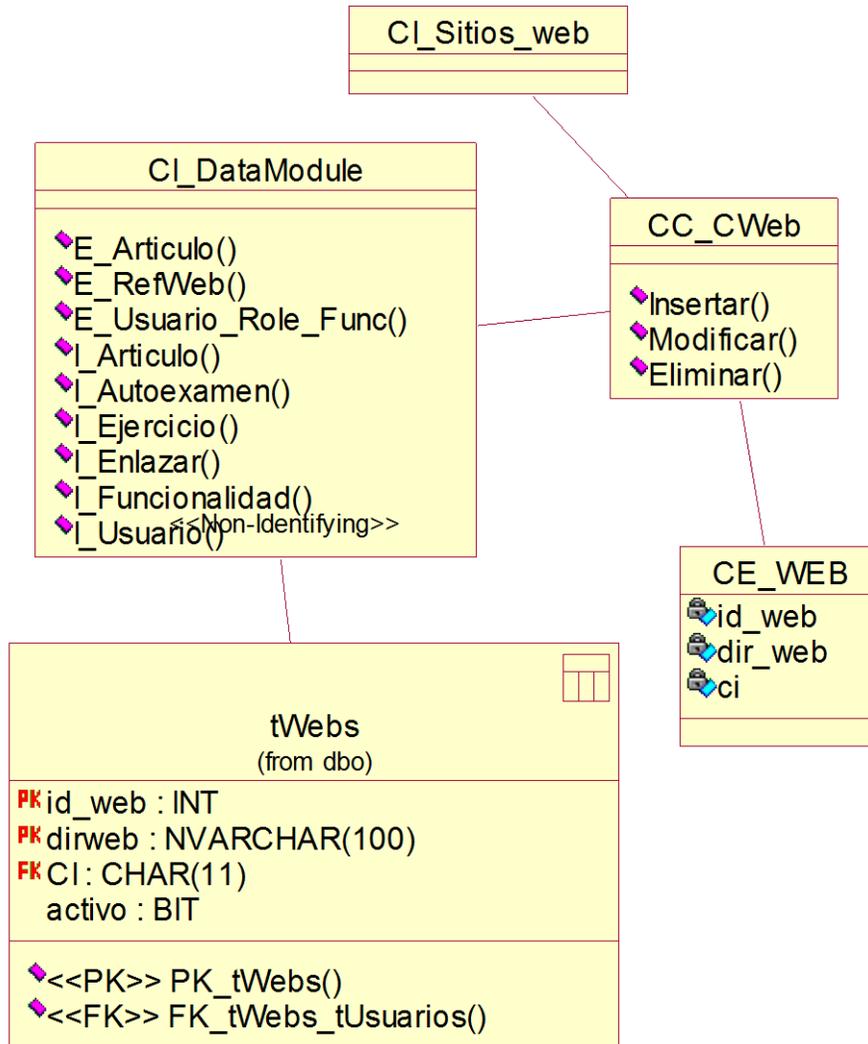
Anexo 18: Diagrama de clases. Caso de uso mostrar problemas resueltos.



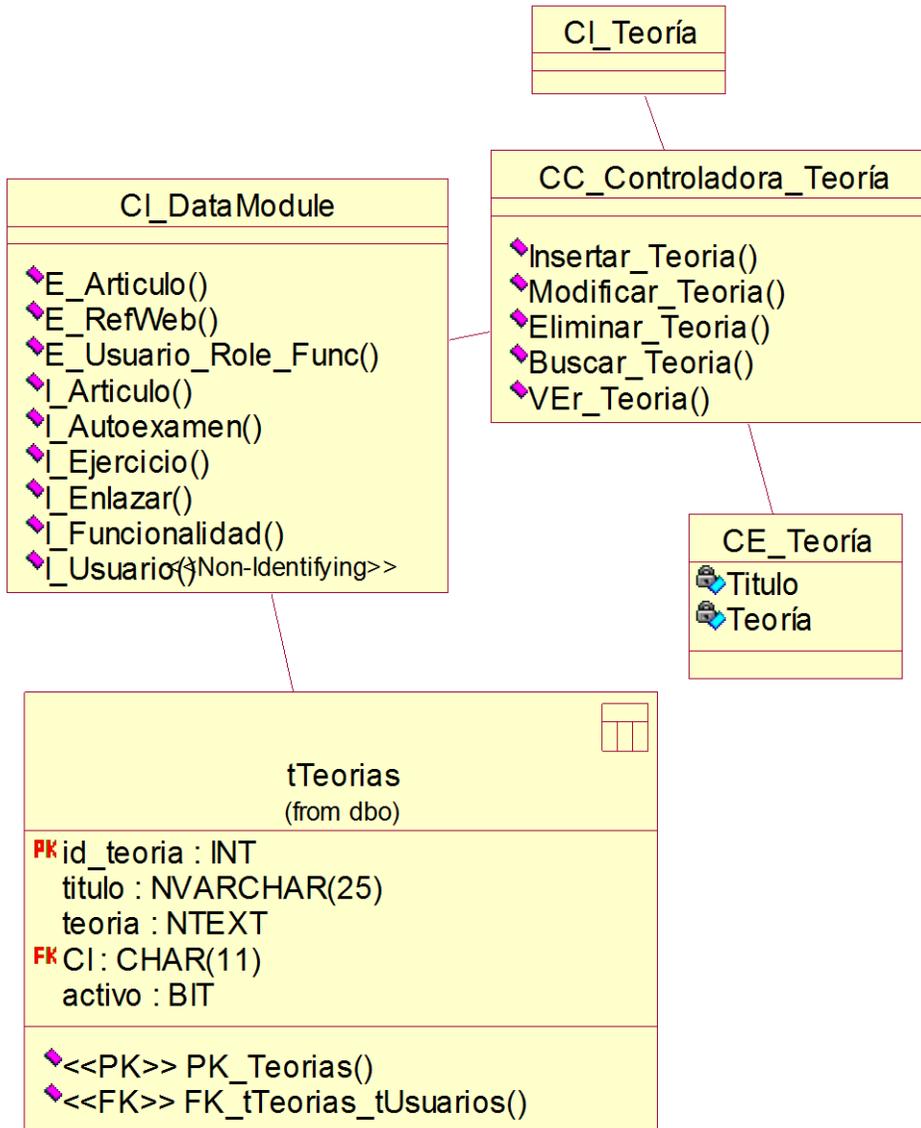
Anexo 19: Diagrama de clases. Caso de uso mostrar reporte.



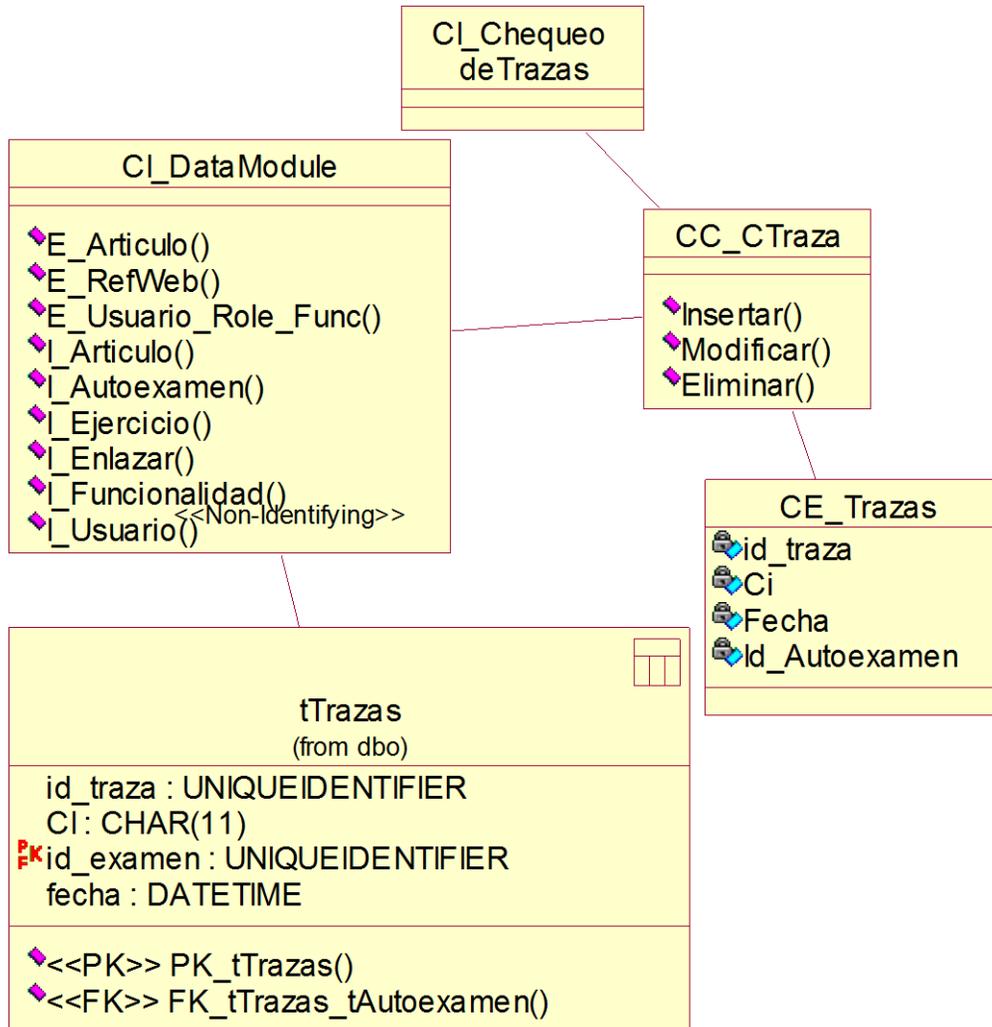
Anexo 20: Diagrama de clases. Caso de uso mostrar sitios web.



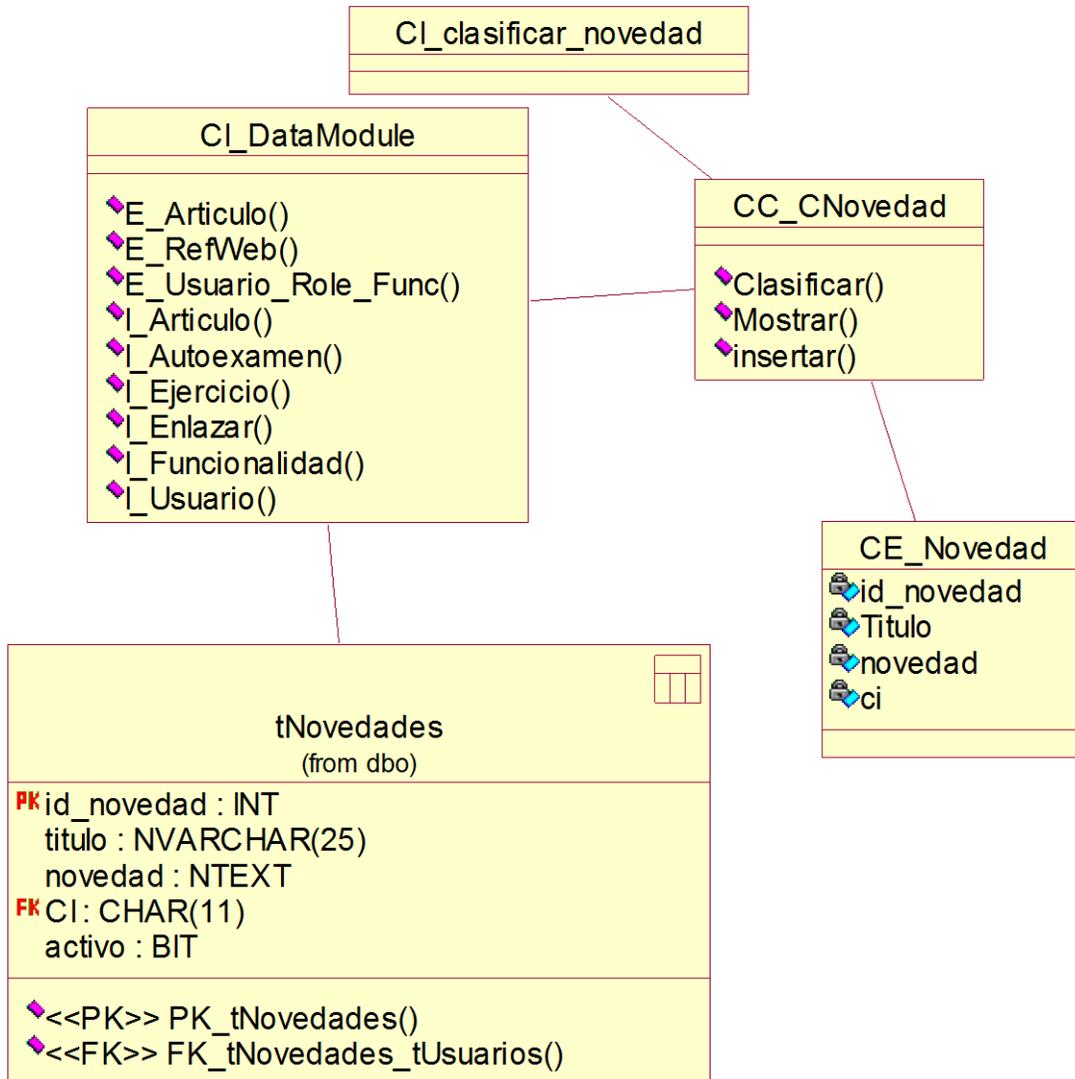
Anexo 21: Diagrama de clases. Caso de uso mostrar teoría.



Anexo 22: Diagrama de clases. Caso de uso mostrar traza.



Anexo 23: Diagrama de clases. Caso de uso gestionar novedad.



Anexo 24: Pantalla principal.



Anexo 25: Pantalla autoexamen.

Nuevo Autoexamen - □ X

Nivel Fácil Nuevo Examen Notas Final: 4.4

Verdadero o Falso Enlazar Selección Historico

Seleccionar la respuesta correcta

2 Siguiente Anterior

Concepto 1. Una empresa constructora dispone de tablas de 50 cm de ancho para satisfacer los pedidos que se hacen desde sus construcciones a pie de obra, de ellas se reciben las solicitudes de: 300 tablas de 28 cm, 500 tablas de 20 cm y 200 de 16 cm. ¿Cuántas tablas deberán utilizarse y en que forma deberán cortarse para lograr un mínimo de desperdicio de madera y satisfacer las solicitudes recibidas? Construya el modelo.

Solución (construcción del modelo o planteamiento matemático del problema):

Nota aclaratoria : En un problema de corte óptimo de materiales en la Industria, lo primero que se debe encontrar para resolver el problema son las variantes de corte, pues ellas se convertirán a la postre en variables esenciales del problema. Para ello se trata de analizar todas las formas posibles de cortar las tablas aun cuando parezcan poco beneficiosas. Si lo son o no la máquina se encargará de tomarlas o desecharlas.

Se dispone de _____ se quieren tablas de _____ desperdicio _____

R1

Definición de variables:

- X 1 -> Cantidad de tablas a cortar según la VC1.
- X 2 -> Cantidad de tablas a cortar según la VC2.
- X 3 -> Cantidad de tablas a cortar según la VC3.
- X 4 -> Cantidad de tablas a cortar según la VC4.
- X 5 -> Cantidad de tablas a cortar según la VC5
- X_j = 0 con j = 1 ...5 (condición de no negatividad)

Construcción del sistema de restricciones:

X 1 + X 2 = 300 para t

R3

Definición de variables:

- X 1 -> Cantidad de tablas a cortar según la VC1.
- X 2 -> Cantidad de tablas a cortar según la VC2.
- X 3 -> Cantidad de tablas a cortar según la VC3.
- X 4 -> Cantidad de tablas a cortar según la VC4.
- X 5 -> Cantidad de tablas a cortar según la VC5
- X_j = 0 con j = 1 ...5 (condición de no negatividad)

Construcción del sistema de restricciones:

X 1 + X 2 = 300 para t

R2

Definición de variables:

- X 1 -> Cantidad de tablas a cortar según la VC1.
- X 2 -> Cantidad de tablas a cortar según la VC2.
- X 3 -> Cantidad de tablas a cortar según la VC3.
- X 4 -> Cantidad de tablas a cortar según la VC4.
- X 5 -> Cantidad de tablas a cortar según la VC5
- X_j = 0 con j = 1 ...5 (condición de no negatividad)

Construcción del sistema de restricciones:

X 1 + X 2 = 300 para t

R4

Definición de variables:

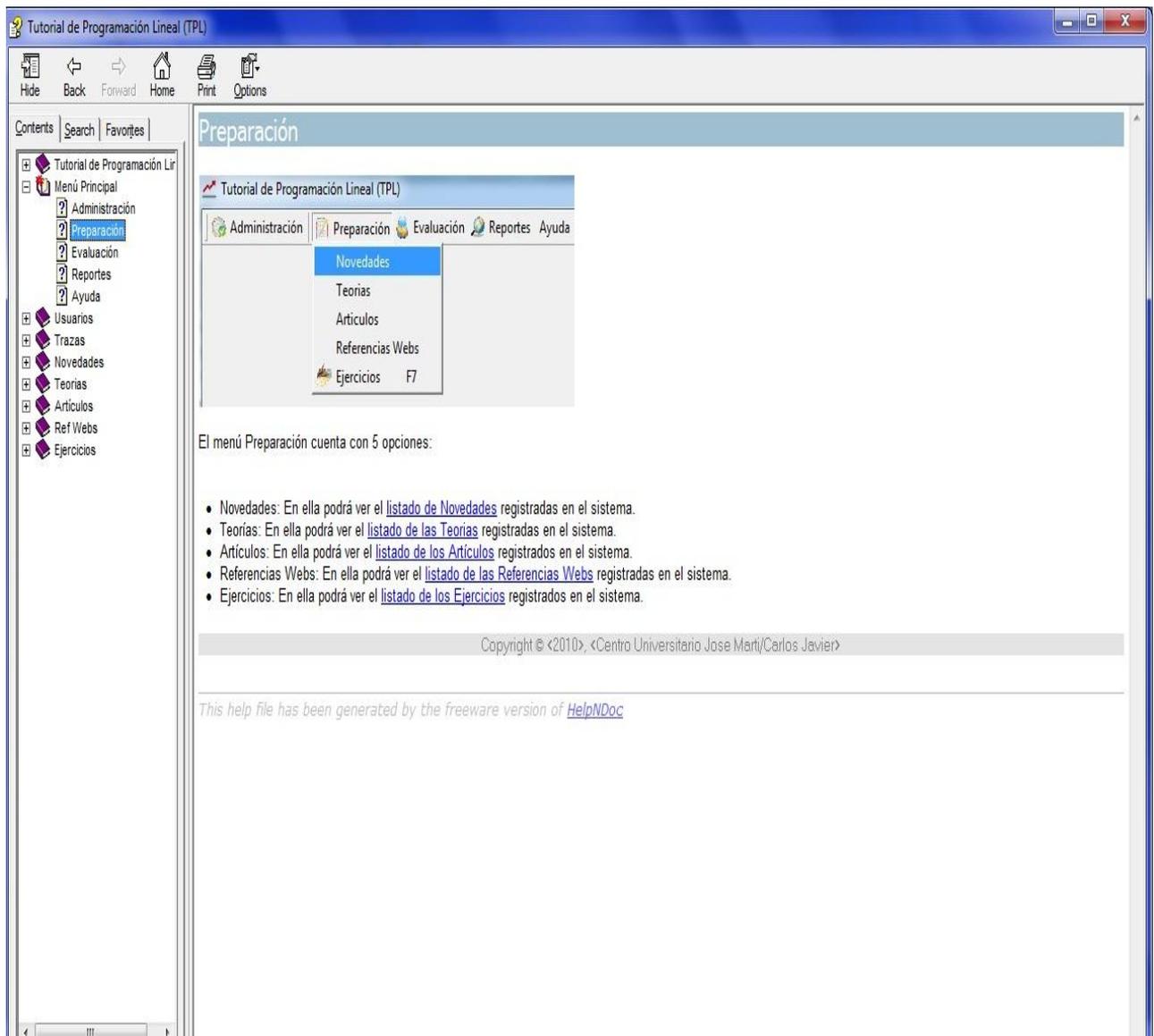
- X 1 -> Cantidad de tablas a cortar según la VC1.
- X 2 -> Cantidad de tablas a cortar según la VC2.
- X 3 -> Cantidad de tablas a cortar según la VC3.
- X 4 -> Cantidad de tablas a cortar según la VC4.
- X 5 -> Cantidad de tablas a cortar según la VC5
- X_j = 0 con j = 1 ...5 (condición de no negatividad)

Construcción del sistema de restricciones:

X 1 + X 2 = 300 seste

Aceptar Cerrar

Anexo 26: Pantalla de la ayuda.



The screenshot shows a web browser window titled "Tutorial de Programación Lineal (TPL)". The browser's address bar and navigation buttons (Hide, Back, Forward, Home, Print, Options) are visible at the top. The page content is divided into a left sidebar and a main area.

Left Sidebar (Contents):

- Tutorial de Programación Lir
- Menú Principal
 - Administración
 - Preparación**
 - Evaluación
 - Reportes
 - Ayuda
- Usuarios
- Trazas
- Novedades
- Teorias
- Articulos
- Ref Webs
- Ejercicios

Main Content Area:

Preparación

Tutorial de Programación Lineal (TPL)

Administración | Preparación | Evaluación | Reportes | Ayuda

- Novedades**
- Teorias
- Articulos
- Referencias Webs
- Ejercicios F7

El menú Preparación cuenta con 5 opciones:

- **Novedades:** En ella podrá ver el [listado de Novedades](#) registradas en el sistema.
- **Teorias:** En ella podrá ver el [listado de las Teorias](#) registradas en el sistema.
- **Articulos:** En ella podrá ver el [listado de los Articulos](#) registrados en el sistema.
- **Referencias Webs:** En ella podrá ver el [listado de las Referencias Webs](#) registradas en el sistema.
- **Ejercicios:** En ella podrá ver el [listado de los Ejercicios](#) registrados en el sistema.

Copyright © <2010>, <Centro Universitario Jose Martí/Carlos Javier>

This help file has been generated by the freeware version of [HelpNDoc](#)

Anexo 27: Pantalla cambiar contraseña.

Cambiar Contraseña



0000000000

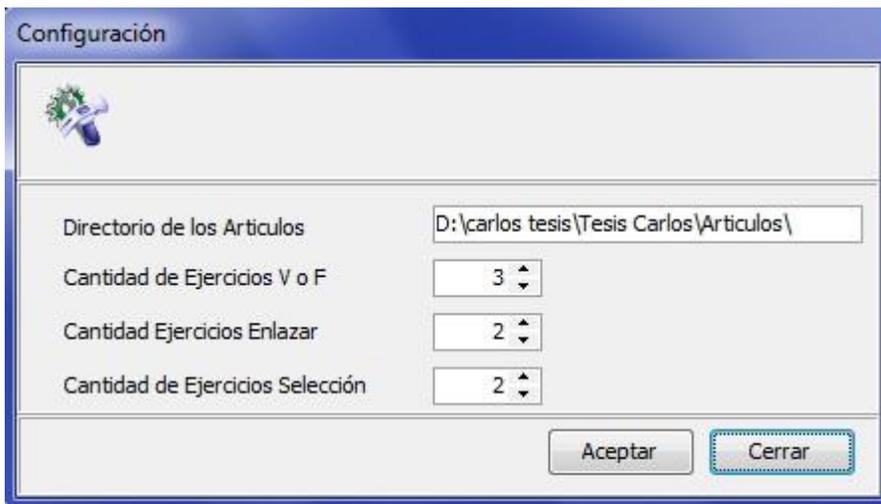
Administrador Administrador

Nueva Contraseña

Repetir

Aceptar Cancelar

Anexo 28: Pantalla configuración.



The image shows a configuration window titled "Configuración" with a blue header bar. Inside the window, there is a small icon of a globe and a blue arrow. Below the icon, there are four configuration items:

- Directorio de los Articulos: A text box containing the path "D:\carlos tesis\Tesis Carlos\Articulos\".
- Cantidad de Ejercicios V o F: A spin box with the value "3".
- Cantidad Ejercicios Enlazar: A spin box with the value "2".
- Cantidad de Ejercicios Selección: A spin box with the value "2".

At the bottom right of the window, there are two buttons: "Aceptar" and "Cerrar".

Anexo 29: Pantalla crear cuenta de usuario.

The image shows a software dialog box titled "Registrar Usuario" (Register User). The dialog has a blue title bar with a close button (X) in the top right corner. Below the title bar is a small icon of a person with a document. The main area contains several input fields and checkboxes:

- CI: A text input field.
- Nombre: A text input field.
- Apellidos: A text input field.
- Correo: A text input field.
- Contraseña: A text input field.
- Activo: A checked checkbox.
- Administrador: An unchecked checkbox.

At the bottom of the dialog, there are three buttons: "Listado", "Nuevo", and "Cancelar". The "Nuevo" button is highlighted with a blue border.

Anexo 30: Pantalla ejercicios de enlazar.

Listado de Ejercicios

Verdadero o Falso Selección **Enlazar** Preparación

Drag a column header here to group by that column

concepto	respuesta	NivelDif	activo
aaaaaa	rrrrr	Fácil	<input checked="" type="checkbox"/>
1. Las producciones en toneladas de tres fábricas de cemento pueden ser absorbidas por dos empresas constructoras, los costos en cientos de pesos de transportación de las producciones de las fábricas a cada uno de las empresas constructoras, las demandas de	Definición de variables: X 1 -> Es la cantidad de toneladas a transportar de la fábrica 1 a la empresa constructora 1. X 2 -> Es la cantidad de toneladas a transportar de la fábrica 2 a la empresa constructora 1. X 3 -> Es la cantidad de toneladas a transportar de la fábrica 3 a la empresa constructora 1. X 4 -> Es la cantidad de toneladas a transportar de la fábrica 1 a la empresa constructora 2.	Fácil	<input checked="" type="checkbox"/>
1. El establecimiento "Planta Habana", perteneciente al combinado cárnico se dedica a la elaboración de productos derivados de la carne, tanto frescos como en conserva. El surtido de	Definición de variables: X 1 -> Cantidad de TM de perro elaborado con el extensor caseína de sodio. X 2 -> Cantidad de TM de perro elaborado con el extensor harina de soya. X 3 -> Cantidad de TM de salchicha elaborada con el extensor caseína de Sodio de soya. X 4 -> Cantidad de TM de salchicha elaborada con el extensor harina de soya. $X_j = 0$ con $j = 1 \dots 4$	Fácil	<input checked="" type="checkbox"/>

3

Nuevo Cerrar

Anexo 31: Pantalla entrar ejercicios de enlazar.

Modificar Ejercicio

Activo

Verdadero o Falso Selección Enlazar Preparación Nivel Fácil

Concepto
aaaaaa

Respuesta
rrrrr

Aceptar Cancelar

Anexo 32: Pantalla entrar artículos.

Nuevo Artículo

Título

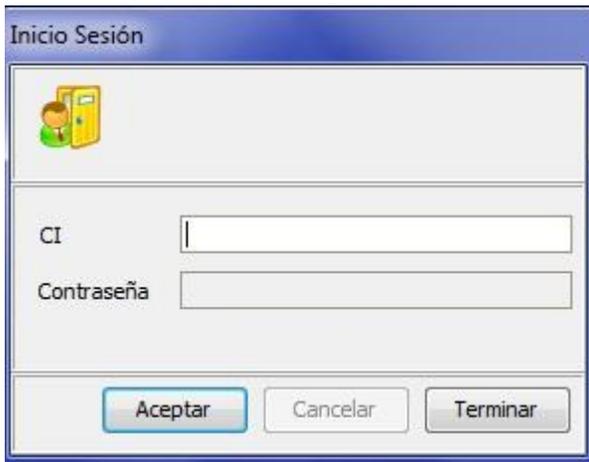
Descripción

<<Seleccione el fichero del artículo>> Limpia

Activo

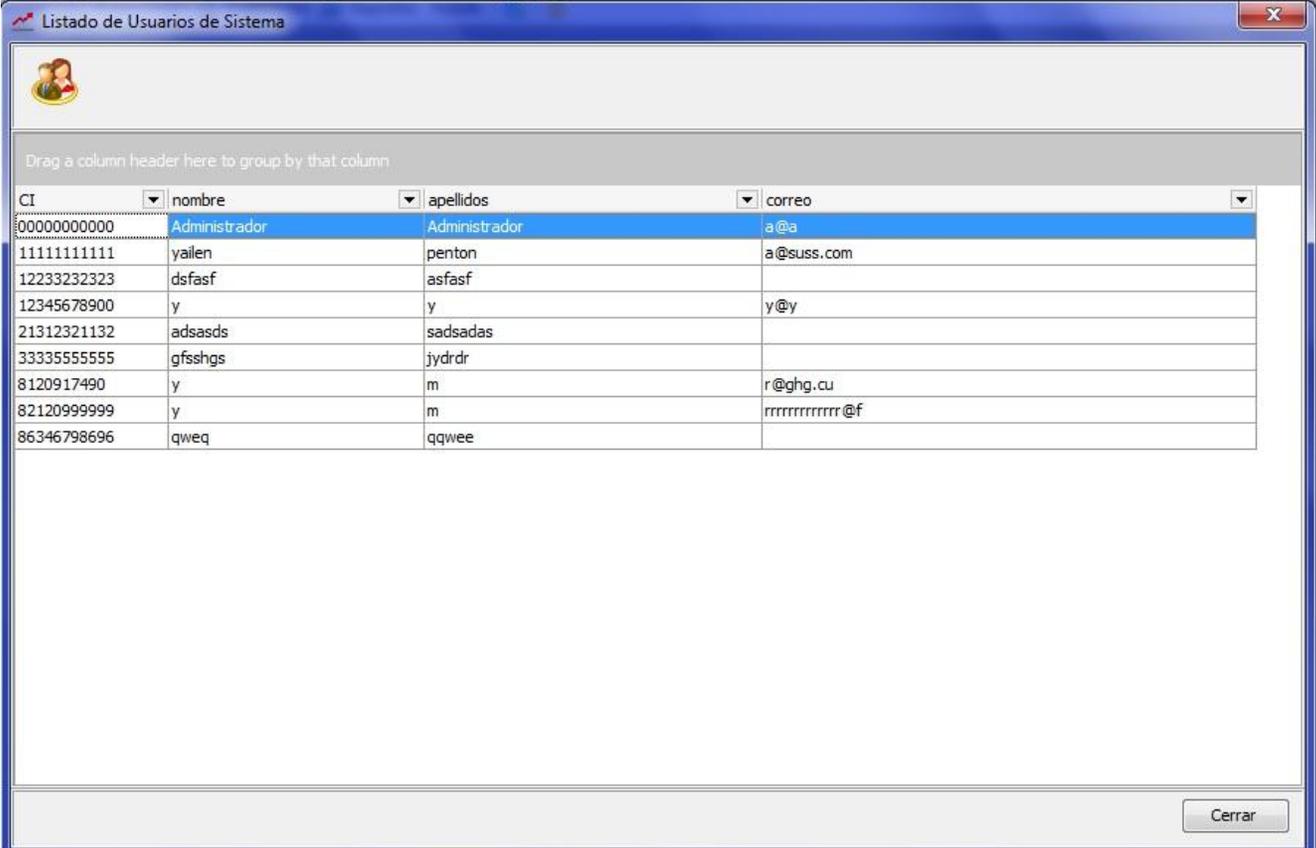
Guardar Cancelar

Anexo 33: Pantalla iniciar sección.



The image shows a login window titled "Inicio Sesión". At the top left, there is a small icon of a person and a document. Below the icon, there are two input fields: one labeled "CI" and another labeled "Contraseña". At the bottom of the window, there are three buttons: "Aceptar", "Cancelar", and "Terminar".

Anexo 34: Pantalla listado de usuarios.



Drag a column header here to group by that column

CI	nombre	apellidos	correo
0000000000	Administrador	Administrador	a@a
1111111111	yailen	penton	a@suss.com
12233232323	dsfasf	asfasf	
12345678900	y	y	y@y
21312321132	adsasds	sadsadas	
33335555555	gfsshgs	jydrdr	
8120917490	y	m	r@ghg.cu
82120999999	y	m	rrrrrrrrrrr@f
86346798696	qweq	qqwee	

Cerrar

Anexo 35: Pantalla gestionar novedad.

The image shows a software window titled "Modificar Novedad". It features a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is divided into two sections: a "Titulo" section with a text input field containing "este es un titulo", and a "Novedad" section with a large text area containing "este es un comentario para probar la forma de representacion de las novedades". At the bottom left, there is a checkbox labeled "Activo" which is checked. At the bottom right, there are two buttons: "Guardar" and "Cancelar".

Modificar Novedad

Titulo
este es un titulo

Novedad
este es un comentario para probar la forma de representacion de las novedades

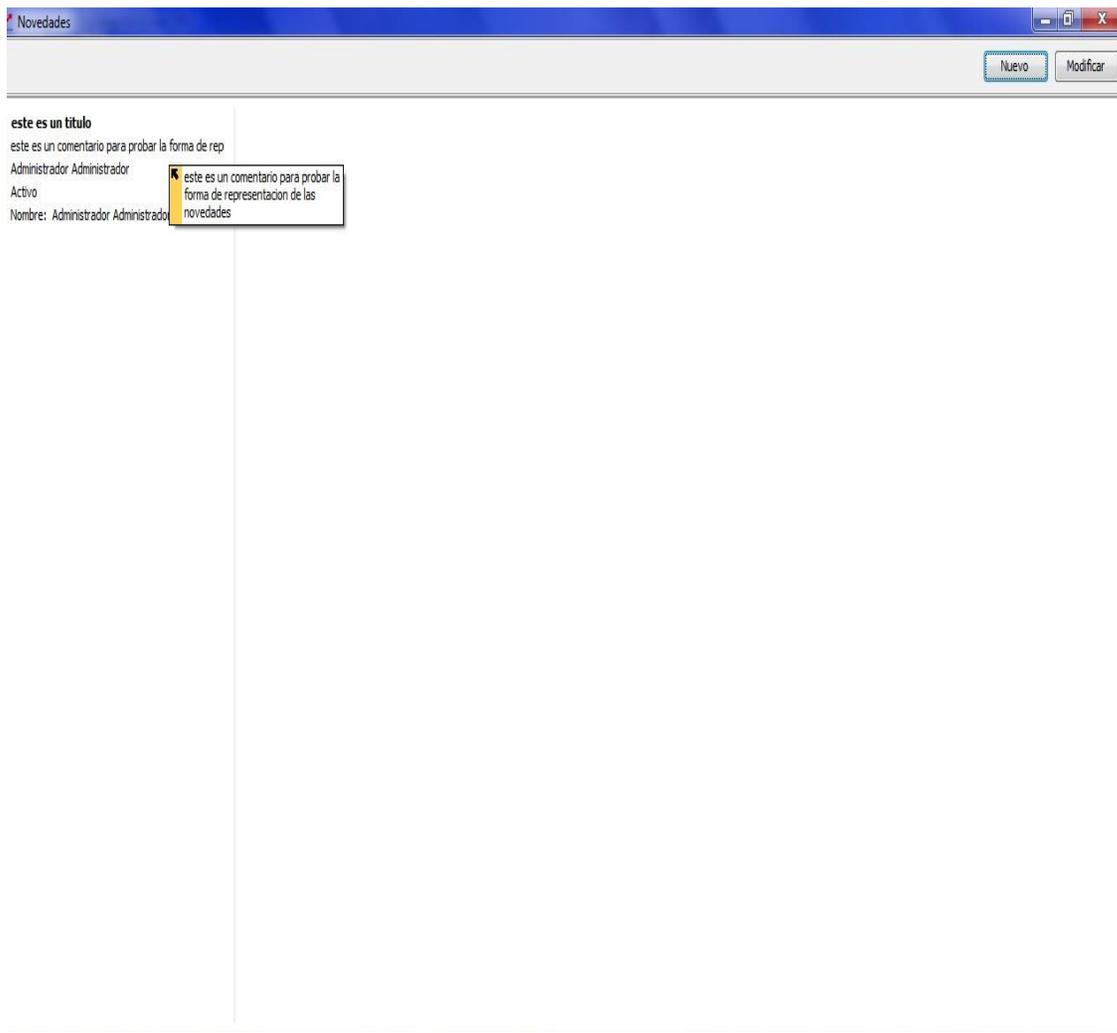
Activo

Guardar Cancelar

Anexo 36: Pantalla mostrar artículos.



Anexo 37: Pantalla mostrar novedad.



Anexo 39: Pantalla mostrar teoría.

The screenshot shows a window titled "Teorias" with a menu bar containing "Nuevo" and "Modificar". The main content area displays a document with the following text:

teoria 1
Aspectos fundamentales de la programación lineal
Administrador Administrador
Activo
Nombre: Administrador Administrador

Aspectos fundamentales de la programación lineal

Se denominan problemas de optimización aquellos tipos de problemas en los cuales se busca el máximo o mínimo de una función con un número determinado de variables, estando los valores de las mismas sujetos a ciertas limitaciones. (a)

•Construcción del modelo de programación lineal

El procedimiento para la construcción del modelo puede resumirse de la forma siguiente :

Paso 1- Definición de las variables de decisión.
Paso 2- Construcción del sistema de restricciones.
Paso 3- Construcción de la función objetivo. (a)

•Definición de las variables de decisión

La definición de las variables de decisión es el primer paso en la construcción del modelo de programación lineal. Cada variable de decisión se identifica con cada una de las actividades en que se descompone el problema que se estudia.

El nombre de variables de decisión que se da a las actividades responde al hecho de que son variables sobre las cuales se puede tomar una decisión directamente para resolver el problema que nos ocupa, en otras palabras, son los elementos a través de los cuales se logra el objetivo que se persigue con la solución del problema. Por ejemplo si el objetivo consiste en minimizar el costo total de producción, el mismo se puede lograr a través del establecimiento de un surtido de productos, el cual confirmaría en este caso el contenido

The Windows taskbar at the bottom shows the Start button, several application icons, and the system tray with the date and time: 12:02 AM 6/6/2010.

Anexo 40: Pantalla entrar problemas de preparación.

Modificar Ejercicio

Verdadero o Falso Selección Enlazar Preparación Nivel: Fácil

Ejercicio Resuelto Ejercicio Normal

Título:

Ejercicio

1. Su Empresa dedicada al transporte recibe la solicitud de distribuir dentro de los límites de la ciudad un producto único, homogéneo, desde tres orígenes hasta tres destinos. La oferta total es de 30 t, distribuidas de la siguiente forma; el almacén 1 puede ofertar hasta 12 t, el 2 hasta 8t y el 3 hasta 10t. La demanda total también es de 30t que pueden ser asimiladas de la siguiente forma: el destino 1 hasta 6t, el 2 hasta 9t y 3 hasta 15t. Se conocen los gastos de combustibles en litros de los orígenes hasta los destinos. Como es sabido por todos, nuestro país está enfrascado en la lucha del ahorro del preciado líquido, por lo que su equipo de trabajo debe elaborar un plan de transportación que garantice el menor gasto posible de combustible. Realice el planteamiento matemático del problema.

	D1	D2	D3	Ofert
Alm 1	1	3	4	12
Alm 2	2	5	3	8
Alm 3	6	7	4	10
Deman	6	9	15	30

Solución (construcción del modelo o planteamiento matemático del problema):

Nota aclaratoria: Resolvamos este problema de transporte como un problema de Programación Lineal.

Definamos para ello las variables con un solo subíndice. (puede hacerse c

Respuesta

Definición de variables:

X 1 -> Es la cantidad de toneladas a transportar del almacén 1 al destino 1. X 3 -> Es la cantidad de toneladas a transportar del almacén 1 al destino 3.
X 2 -> Es la cantidad de toneladas a transportar del almacén 1 al destino 2.
X 4 -> Es la cantidad de toneladas a transportar del almacén 2 al destino 1.
X 5 -> Es la cantidad de toneladas a transportar del almacén 2 al destino 2.
X 6 -> Es la cantidad de toneladas a transportar del almacén 2 al destino 3.
X 7 -> Es la cantidad de toneladas a transportar del almacén 3 al destino 1.
X 8 -> Es la cantidad de toneladas a transportar del almacén 3 al destino 2.
X 9 -> Es la cantidad de toneladas a transportar del almacén 3 al destino 3.

$X_j = 0$ con $j = 1..9$ (condición de no negatividad)

Construcción del sistema de restricciones: Aquí se deben hacer dos grupos de restricciones, un grupo perteneciente a las ofertas (orígenes) y otro perteneciente a las demandas (destinos):

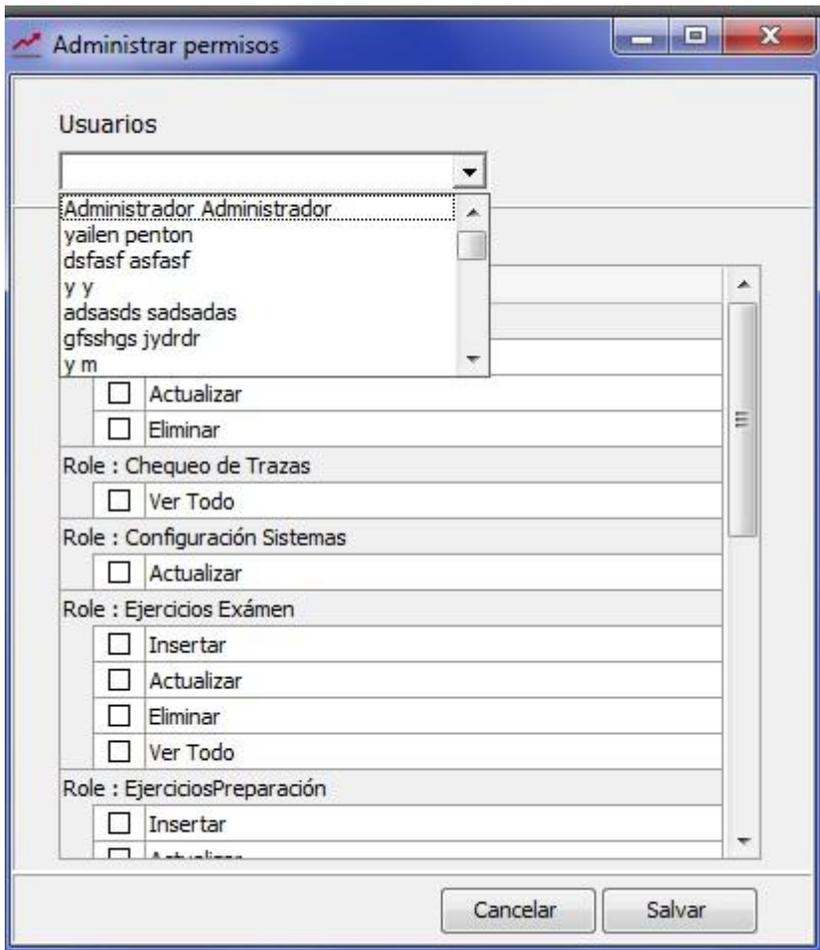
Aceptar Cancelar

Anexo 41: Pantalla reporte del sistema.

Listado General del Sistema	
6/6/2010	
Cant Ejercicios Verdadero o Falso	15
Cant Ejercicios Enlazar	3
Cant Ejercicios Selección	2
Cant Ejercicios Preparación	3
Cant Artículos	2
Cant Referencias Webs	1
Cant Teorias	1
Cant Usuarios	9
Cant Novedades	1

Page 1 of 1

Anexo 42: Pantalla administrar permisos de usuario.



Anexo 43: pantalla modificar ejercicios de selección.

Modificar Ejercicio

Activo

Verdadero o Falso
 Selección
 Enlazar
 Preparación
 Nivel: **Fácil**

Concepto 1. La fábrica de muebles "Mártires de Barbados" desea establecer un proyecto de plan de producción para un trimestre con el objetivo de maximizar las ganancias. Realice el planteamiento matemático del problema. De los recursos disponibles el más deficitario es la madera, de la que se cuenta con 960 pies³ y los insumos de este material por la construcción por unidad, son de 12 pies³ para las mesas y 8 pies³ para el juego de sillas. Por otra parte la mano de obra debido a la calificación necesaria de los obreros está limitada a 720 horas y se conoce además que una mesa consume 6 horas, mientras que para el juego de sillas 12. La carpintería por un pedido que quedó pendiente de la etapa anterior está obligada a producir por lo menos 2 mesas y 2 juegos de sillas. Se debe significar además que las ganancias por cada mesa o juego de silla que se construya es de 50 pesos.

Verdadero

F1

F2

F3

Definición de variables:
 X 1 -> Cantidad de mesas a fabricar
 X 2 -> Cantidad de juegos de sillas a fabricar
 X_j = 0 con j = 1 y 2 (condición de no negatividad)

Construcción del sistema de restricciones:
 $12 X_1 + 8 X_2 = 960$ pies³ de madera
 $6 X_1 + 12 X_2 = 720$ horas de mano de obra
 $X_1 = 2$
 $X_2 = 2$

Construcción de la función Objetivo:
 Max Z = 50 X

Definición de variables:
 X 1 -> Cantidad de mesas a fabricar
 X 2 -> Cantidad de juegos de sillas a fabricar
 X_j = 0 con j = 1 y 2 (condición de no negatividad)

Construcción del sistema de restricciones:
 $12 X_1 + 8 X_2 = 960$ pies³ de madera
 $6 X_1 + 12 X_2 = 720$ horas de mano de obra
 $X_1 = 2$
 $X_2 = 2$

Construcción de la función Objetivo:
 Max Z = 50 X

Definición de variables:
 X 1 -> Cantidad de mesas a fabricar
 X 2 -> Cantidad de juegos de sillas a fabricar
 X_j = 0 con j = 1 y 2 (condición de no negatividad)

Construcción del sistema de restricciones:
 $12 X_1 + 8 X_2 = 960$ pies³ de madera
 $6 X_1 + 12 X_2 = 720$ horas de mano de obra
 $X_1 = 2$
 $X_2 = 4$

Construcción de la función Objetivo:
 Max Z = 50 X

Definición de variables:
 X 1 -> Cantidad de mesas a fabricar
 X 2 -> Cantidad de juegos de sillas a fabricar
 X_j = 0 con j = 1 y 2 (condición de no negatividad)

Construcción del sistema de restricciones:
 $12 X_1 + 8 X_2 = 960$ pies³ de madera
 $6 X_1 + 12 X_2 = 720$ horas de mano de obra
 $X_1 = 2$
 $X_2 = 2$

Construcción de la función Objetivo:
 Max Z = 50 X

Aceptar Cancelar

Anexo 44: Pantalla trazas del autoexamen.

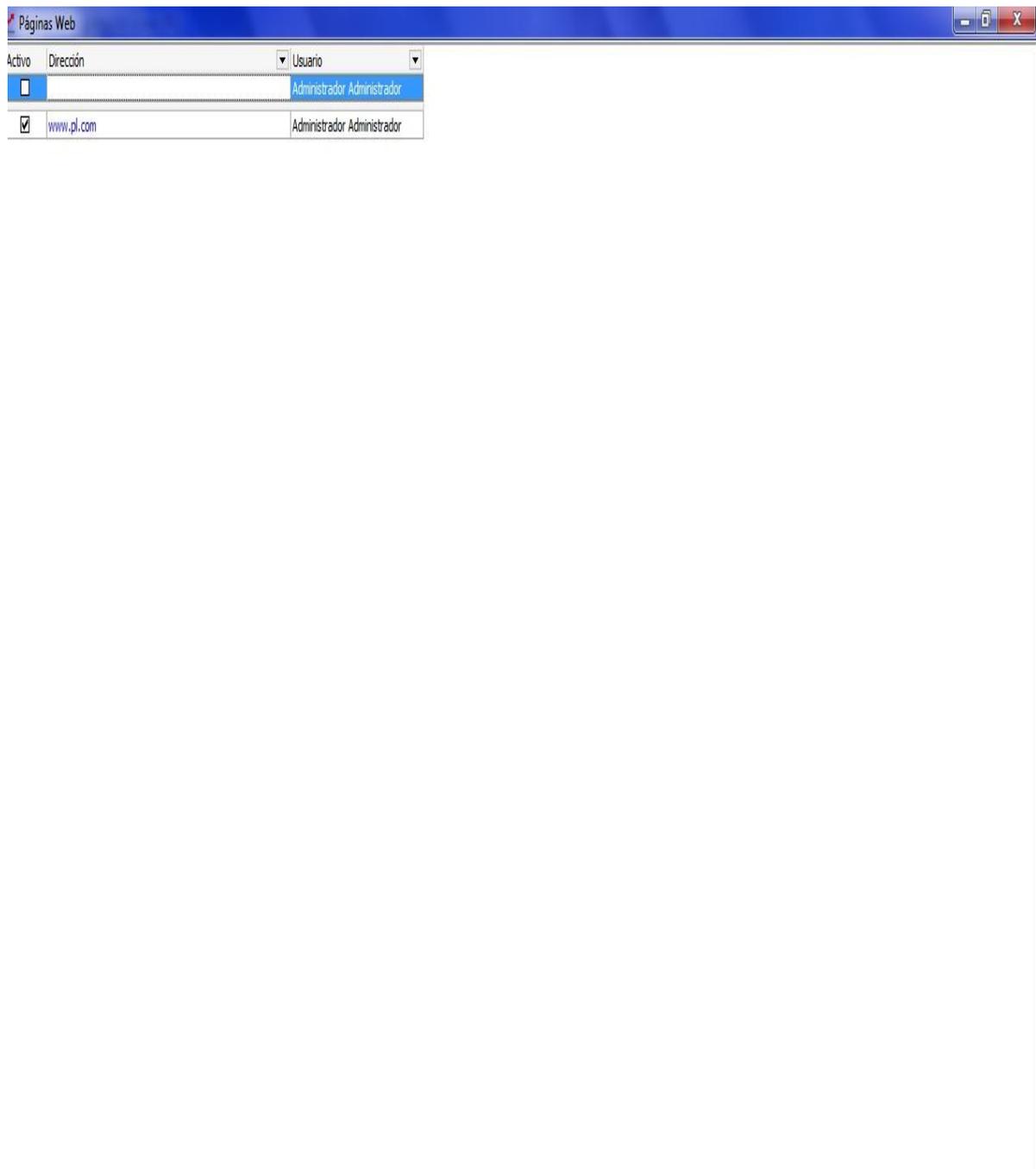
Trazas Autoexamen

Drag a column header here to group by that column

Nombre	Apellidos	Nota Final	Fecha Examen	Correo
Administrador	Administrador	5	5/24/2010 10:24:23 AM	a@a
Administrador	Administrador	3.8	5/19/2010 10:30:50 PM	a@a
Administrador	Administrador	3.8	5/19/2010 10:06:26 PM	a@a
Administrador	Administrador	4.4	6/6/2010 12:17:24 AM	a@a
yallen	penton	2	5/20/2010 12:09:32 AM	a@suss.com
Administrador	Administrador	5	5/19/2010 10:19:54 PM	a@a
yallen	penton	5	5/20/2010 12:08:39 AM	a@suss.com
Administrador	Administrador	4.7	5/19/2010 10:14:00 PM	a@a
Administrador	Administrador	2	5/19/2010 10:02:40 PM	a@a
Administrador	Administrador	2	5/24/2010 10:26:36 AM	a@a
Administrador	Administrador	2	5/19/2010 10:33:35 PM	a@a
Administrador	Administrador	4.7	5/19/2010 10:28:25 PM	a@a
Administrador	Administrador	3.8	5/19/2010 10:11:24 PM	a@a
Administrador	Administrador	2.2	6/6/2010 12:16:23 AM	a@a
Administrador	Administrador	3.3	5/20/2010 12:07:38 AM	a@a
Administrador	Administrador	2	5/20/2010 12:43:44 AM	a@a
Administrador	Administrador	4.7	5/19/2010 10:29:16 PM	a@a
Administrador	Administrador	4.7	5/19/2010 10:15:55 PM	a@a
Administrador	Administrador	2.2	5/19/2010 10:03:23 PM	a@a
Administrador	Administrador	4.7	5/19/2010 10:20:46 PM	a@a
Administrador	Administrador	4.7	5/19/2010 10:27:19 PM	a@a
Administrador	Administrador	2	5/20/2010 12:28:55 AM	a@a

Cerrar

Anexo 45: Pantalla páginas web.



The screenshot shows a window titled "Páginas Web" with a table of web pages. The table has columns for "Activo", "Dirección", and "Usuario". The first row is highlighted in blue and has an unchecked checkbox, an empty "Dirección" field, and "Administrador" in the "Usuario" column. The second row has a checked checkbox, the URL "www.pl.com", and "Administrador" in the "Usuario" column.

Activo	Dirección	Usuario
<input type="checkbox"/>		Administrador
<input checked="" type="checkbox"/>	www.pl.com	Administrador

Anexo 46: Pantalla teoría.

Modificar Teoría

Título
teoría 1

Teoría

Aspectos fundamentales de la programación lineal

Se denominan problemas de optimización aquellos tipos de problemas en los cuales se busca el máximo o mínimo de una función con un número determinado de variables, estando los valores de las mismas sujetos a ciertas limitaciones. (a)

- Construcción del modelo de programación lineal

El procedimiento para la construcción del modelo puede resumirse de la forma siguiente :

Paso 1- Definición de las variables de decisión.
Paso 2- Construcción del sistema de restricciones.
Paso 3- Construcción de la función objetivo. (a)

- Definición de las variables de decisión

La definición de las variables de decisión es el primer paso en la construcción del modelo de programación lineal. Cada variable de decisión se identifica con cada una de las actividades en que se descompone el problema que se estudia.

El nombre de variables de decisión que se da a las actividades responde al hecho de que son variables sobre las cuales se puede tomar una decisión directamente para resolver el problema que nos ocupa, en otras palabras, son los elementos a través de los cuales se logra el objetivo que se persigue con la solución del problema. Por ejemplo si el objetivo consiste en minimizar el costo total de producción, el mismo se puede lograr a través del establecimiento de un surtido de productos, el cual conformaría en este caso el conjunto de las variables de decisión. (a)

- 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. (a)

- Construcción de la función objetivo

La función objetivo es una función lineal que incluye a todas las variables del problema (aunque haya algunas que posean coeficientes nulo) y expresa el propósito central que se persigue con la solución del problema. (a)

Activo

Guardar Cancelar