

**Universidad de Sancti Spíritus**

**“José Martí Pérez”**



**Facultad de Ingeniería**

“Tutorial para el apoyo del proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado de la carrera Ingeniería Informática de la Uniss”.

**Trabajo para optar por el título de Ingeniería Informática**

**Autor**

Dunelvis Mesa Pérez.

**Tutor**

MSc. Roxana Martín Ramos.

Sancti Spíritus

2013

## **Dedicatoria**

*Dedicatoria*

*A familia por su apoyo durante todos estos años.*

*A todas las persona que de una forma u otra hicieron posible la realización de este trabajo.*

## Agradecimientos

### *Agradecimientos*

*A mi hermana por estar siempre a mi lado brindándome su amor y cariño.*

*A mis padres por la educación que me han brindado, por su amor y apoyo incondicional y hacer de mí la persona que soy hoy.*

*A mi novia quien ha estado a mi lado todo este tiempo de lucha y estrés brindándome su amor, apoyo y sobre todo mucha paciencia.*

*A mis compañeros de aula, en especial a mis amigos Alexander, Claudia e Ivonne.*

*A Carlos quien me ha brindado incondicionalmente su apoyo.*

*A mi tutora Roxana por su preocupación, tiempo y dedicación.*

### Resumen

La investigación que se presenta a continuación se titula “Tutorial para el apoyo del proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado de la carrera Ingeniería Informática de la Uniss” y surgió por la necesidad que se tenía en la asignatura mencionada de disponer de un software educativo que contribuyera a una mejor comprensión de este tema tan complejo para los estudiantes de la carrera.

Las etapas llevadas a cabo en la elaboración del sistema fueron las planteadas por el Proceso Unificado de Desarrollo (RUP), en tal sentido se utilizó como herramienta para el análisis y diseño: Rational Rose Enterprise Edition. El sistema se diseñó utilizando el patrón Modelo Vista Controlador, y se emplearon los mapas conceptuales hipermediales como forma de representar el conocimiento en aras de una mayor comprensión de los algoritmos genéticos. Para ello, los mapas fueron creados utilizando el CMapTools, herramienta gratuita de fácil utilización. Luego se empleó PHP como lenguaje de programación, lo que proporcionó un código eficiente; para la persistencia de los datos, se utilizó MySQL, por ser muy rápido en entornos web; y como servidor web, se escogió Apache, por la seguridad y estabilidad que presenta.

### **Abstract**

The research presented below is titled "Tutorial to support the teaching - learning of the subject of Genetic Algorithms in Automated Learning at Uniss Computer Engineering career" and arose from the need on the subject mentioned to have an educational software to contribute to a better understanding of this complex subject for students of the career.

The steps taken in the development of the system were raised by the Rational Unified Process (RUP), in this sense is used as a tool for analysis and design: Rational Rose Enterprise Edition. The system was designed using the Model View Controller pattern, and used hypermedia concept maps as a way of representing knowledge for the sake of a better understanding of genetic algorithms. To do this, the maps were created using the CMapTools, easy to use a free tool. Then used as the programming language PHP, which provided efficient code for data persistence, we used MySQL, being very fast in the Web, and as a web server, Apache was chosen for security and stability presents.

**Índice**

Introducción .....	1
Capítulo1: Fundamentación teórica y metodológica que sustenta la elaboración de un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.....	6
1.1 El proceso de Enseñanza-Aprendizaje. ....	6
1.1.1 Medios de enseñanza.....	7
1.1.2 Mapas conceptuales. ....	7
1.2 Las Tecnologías de la Información y las Comunicaciones (TIC). ....	9
1.2.1 Las TIC empleadas a la Educación. ....	10
1.2.2 La computadora como medio de enseñanza. ....	11
1.2.3 Sistemas Tutoriales.....	12
1.3 La inteligencia artificial.....	12
1.3.1 Los algoritmos genéticos.....	13
1.3.2 Ventajas de los AG.....	13
1.3.3 Aplicaciones de los AG.....	14
1.3.4 Enseñanza - aprendizaje de Algoritmos Genéticos. ....	14
1.4 Fundamentación de la metodología utilizada.....	15
1.4.1 La Metodología RUP.....	15
1.4.2 ¿Por qué la metodología RUP? .....	16
1.4.3 Lenguaje Unificado de Modelado (UML). ....	16
1.4.4 ¿Por qué UML?.....	19
1.4.5 Arquitectura en N Capas. ....	19
1.4.6 Arquitectura Cliente-Servidor.....	20
1.4.7 Modelo vista controlador (MVC).....	21
1.5 Tecnologías web.....	21

1.5.1 CSS.....	21
1.5.2 ¿Por qué CSS?.....	22
1.5.3 Java Script.....	22
1.5.4 ¿Por qué Java Script?.....	22
1.5.5 HTML.....	23
1.5.6 ASP.....	23
1.5.7 PHP.....	24
1.5.8 ¿Por qué PHP?.....	24
1.5.9 Servidor Web Apache.....	25
1.6 Sistemas Gestores de Bases de Datos.....	25
1.6.1 SQL.....	26
1.6.2 SQL Server 2000.....	26
1.6.3 PostgreSQL.....	26
1.6.4 MySQL.....	26
1.6.5 ¿Por qué MySQL?.....	27
1.7 Herramientas de desarrollo.....	27
1.7.1 Macromedia Dreamweaver.....	27
1.7.2 Rational Rose.....	27
1.7.3 CmapTools.....	28
1.7.4 Exelearning.....	28
1.7.5 Server2Go.....	28
Capítulo 2. Descripción de la solución propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.....	30
2.1 Identificación de los procesos del negocio.....	30
2.1.1 Reglas del Negocio:.....	30
2.1.2 Modelo de casos de uso del negocio.....	30

2.1.3 Actores del Negocio.....	31
2.1.4 Trabajadores del negocio.....	31
2.1.5 Diagrama de casos de uso del negocio.....	32
2.1.6 Descripción de los Casos de Uso del negocio. ....	32
2.1.7 Diagrama de actividades.....	33
2.1.8 Modelo de objetos del negocio.....	33
2.2 Especificación de los requisitos de software.....	34
2.2.1 Listado de los requerimientos funcionales. ....	34
2.2.2 Requisitos no funcionales. ....	35
2.3 Descripción del sistema propuesto. ....	37
2.3.1 Modelo de casos de uso del sistema. ....	37
2.3.2 Actores del sistema.....	38
2.3.3 Casos de uso del sistema. ....	38
2.3.4 Descripción de los casos de uso del sistema. ....	42
Capítulo 3: Implementación de la aplicación propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado: .....	50
Introducción.....	50
3.1 Diagrama de Clases Web. ....	50
3.2 Diseño de la base de datos.....	58
3.2.1 Diagrama de clases persistentes.....	58
3.2.2 Diagrama del Modelo Físico de Datos.....	59
3.3 Principios de diseño. ....	60
3.3.1 Estándares en la interfaz de la aplicación.....	60
3.3.2 Concepción general de la ayuda. ....	61
3.3.3 Tratamiento de errores.....	61

3.3.4 Seguridad.....	61
3.4 Modelo de implementación.....	62
3.4.1 Diagrama de despliegue.....	62
3.4.2 Diagrama de componentes.....	62
Conclusiones.....	64
Recomendaciones.....	65
Bibliografía.....	66
Anexos.....	69

## Índice de figuras

Figura 1: Elementos del proceso Enseñanza-Aprendizaje.....	6
Figura 2. Elementos fundamentales de UML.....	17
Figura 3. Diagrama de casos de uso del negocio.....	32
Figura 4. Diagrama de actividades. CU Consultar bibliografía.....	33
Figura 5. Diagrama del modelo de objetos.....	34
Figura 6. Diagrama de casos de uso del sistema por paquetes.....	40
Figura 7. Diagrama de casos de uso. Paquete seguridad.....	40
Figura 8. Diagrama de casos de uso. Paquete ejercicios.....	41
Figura 9. Diagrama de casos de uso. Paquete reportes.....	41
Figura 10. Diagramas de clases de diseño. Caso de uso Modificar perfil.....	51
Figura 11. Diagramas de clases de diseño. Caso de uso Crear cuenta.....	51
Figura 12. Diagramas de clases de diseño. Caso de uso Autenticar usuario.....	52
Figura 13. Diagramas de clases de diseño. Caso de uso Cambiar contraseña.....	52
Figura 14. Diagramas de clases de diseño. Caso de uso Gestionar usuario.....	53
Figura 15. Diagramas de clases de diseño. Caso de uso Gestionar ejercicios propuestos.....	53
Figura 16. Diagramas de clases de diseño. Caso de uso Gestionar ejercicios resueltos.....	54
Figura 17. Diagramas de clases de diseño. Caso de uso Realizar autoexamen.....	54
Figura 18. Diagramas de clases de diseño. Caso de uso Reportar usuarios del sistema.....	55
Figura 19. Diagramas de clases de diseño. Caso de uso Ver recursos visitados.....	55
Figura 20. Diagramas de clases de diseño. Caso de uso Reportar estudiantes.....	56
Figura 21. Diagramas de clases de diseño. Caso de uso Ver resultados de examen.....	56

## *Índice de figuras*

Figura 22. Diagramas de clases de diseño. Caso de uso Consultar ejercicios propuestos. ....	57
Figura 23. Diagramas de clases de diseño. Caso de uso Consultar ejercicios resueltos. ....	57
Figura 24. Diagramas de clases persistentes. ....	58
Figura 25. Modelo de datos. ....	59
Figura 26. Diagramas de despliegue. ....	62
Figura 27. Diagramas de componentes. ....	63

## Índice de tablas

Tabla 1. Descripción del actor del negocio. ....	31
Tabla 2. Descripción del trabajador del negocio. ....	31
Tabla 3. Descripción del caso de uso consultar bibliografía. ....	32
Tabla 4. Descripción de los actores del sistema. ....	38
Tabla 5. Descripción del caso de uso consultar bibliografía. ....	42
Tabla 6. Descripción del caso de uso Crear cuenta. ....	42
Tabla 7. Descripción del caso de uso Autenticar usuario. ....	42
Tabla 8. Descripción del caso de uso Cambiar contraseña. ....	43
Tabla 9. Descripción del caso de uso Gestionar usuario. ....	43
Tabla 10. Descripción del caso de uso Gestionar ejercicios propuestos. ....	44
Tabla 11. Descripción del caso de uso Gestionar ejercicios resueltos. ....	45
Tabla 12. Descripción del caso de uso Realizar autoexamen. ....	45
Tabla 13. Descripción del caso de uso Reportar usuarios del sistema. ....	46
Tabla 14. Descripción del caso de uso Ver recursos visitados. ....	46
Tabla 15. Descripción del caso de uso Reportar estudiantes. ....	47
Tabla 16. Descripción del caso de uso Ver resultados de examen. ....	47
Tabla 17. Descripción del caso de uso Consultar ejercicios propuestos. ....	48
Tabla 18. Descripción del caso de uso Consultar ejercicios resueltos. ....	48

## Introducción

Actualmente las Tecnologías de la Información y la Comunicación TICs están sufriendo un desarrollo vertiginoso, esto está afectando a prácticamente todos los campos de nuestra sociedad, y la educación no es una excepción. 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).

No se puede entender el mundo de hoy sin un mínimo de cultura informática. Es preciso entender cómo se genera, cómo se almacena, cómo se transforma, cómo se transmite y cómo se accede a la información en sus múltiples manifestaciones (textos, imágenes, sonidos) si no se quiere estar al margen de las corrientes culturales (Eugenia, 2005).

Estas tecnologías han llegado a ser uno de los pilares básicos de la sociedad y hoy es necesario proporcionar al ciudadano una educación que tenga en cuenta esta realidad. Las posibilidades educativas de las TIC han de ser consideradas en dos aspectos: su conocimiento y su uso. La relación entre las TIC y la educación tiene dos vertientes: por un lado, los ciudadanos se ven abocados a conocer y aprender sobre las TIC; por otro, las TIC pueden aplicarse al proceso educativo. Este doble aspecto se refleja en dos expectativas educativas distintas: por un lado, tenemos a los informáticos, interesados en aprender informática, y en el otro, a los profesores, interesados en el uso de la informática para la educación (Rosario, 2005).

La introducción de las TIC en el proceso de enseñanza-aprendizaje, es una realidad y una necesidad social impuesta por el desarrollo tecnológico de la sociedad, ya que la "Era de la Internet" exige cambios y los profesionales de la educación tienen múltiples razones para aprovechar las nuevas posibilidades que proporcionan las nuevas tecnologías para impulsar este cambio hacia un nuevo paradigma educativo más personalizado y centrado en la actividad de los estudiantes.

Precisamente para favorecer este proceso que se empieza a desarrollar desde los entornos educativos, la escuela debe integrar también la nueva cultura: alfabetización digital, fuente de información, instrumento de productividad para realizar trabajos, material didáctico, instrumento cognitivo. Obviamente la escuela debe acercar a los estudiantes la

cultura de hoy, no la cultura de ayer. Por ello es importante la presencia en clase del ordenador (y de la cámara de vídeo, y de la televisión), como un instrumento más, que se utilizará con finalidades diversas: lúdicas, informativas, comunicativas e instructivas. Por lo que nos queda claro que la educación no puede quedar al margen de estos cambios, pues del mismo modo que en su día sucedió con la introducción de la escritura o la aparición de la imprenta, el auge de las TIC ha transformada nuestro modo de pensar, de relacionarnos e incluso la manera de enseñar y de aprender (Graells, 2011).

La universidad cubana no ha quedado exenta de dicha revolución tecnológica, pues desde hace varias décadas ha sido una preocupación e interés del Ministerio de Educación de Cuba, llevar a todos sus Centros Universitarios, los nuevos adelantos de las técnicas más actualizadas de la información y la comunicación, al servicio de profesores y estudiantes, con el objetivo de lograr un egresado con mente abierta y gran creatividad capaz de enfrentar los retos que ofrece la sociedad de estos tiempos (Historia de la Informática en América Latina y el Caribe: Investigaciones y testimonios, 2010).

En julio de 1976 la Facultad de Tecnología de la UH se convirtió en el Instituto Superior Politécnico «José Antonio Echeverría» (ISPJAE); en 1977 inició la carrera de Ingeniería en Sistemas Automatizados de Dirección (SAD), con estudiantes del primer y segundo años, realizando la primera graduación en 1981. La Ingeniería en SAD cubana fue la génesis de la actual ingeniería informática. Su perfil terminal resultó muy similar al de la Ingeniería en Sistemas de Información de los estándares de la ACM y el IEEE, de manera que ese currículo se inició en Cuba algunos años antes; desde 1990 esta carrera se denominó Ingeniería Informática (Historia de la Informática en América Latina y el Caribe: Investigaciones y testimonios, 2010).

En septiembre del 2002 se creó la Universidad de las Ciencias Informáticas (UCI), comenzando su primer curso con 2000 estudiantes. El 19 de julio del 2007 se graduaron sus primeros 1334 alumnos, diplomados en Ciencias Informáticas. En su sede central hoy estudian más de diez mil jóvenes, en diez facultades, las que se diferencian por el perfil terminal de especialización del egresado, entre ellos: comunicaciones, aplicaciones médicas y gestión de la salud, informática industrial, software libre, entre otros, aunque reciben un

título único como ingenieros en ciencias informáticas (Historia de la Informática en América Latina y el Caribe: Investigaciones y testimonios, 2010).

La inteligencia artificial (IA) es considerada una rama de la computación y forma parte del plan de estudio de la carrera Ingeniería Informática. Como disciplina, la IA agrupa varias asignaturas como: Introducción a la IA, IA, Matemática Discreta, Minería de Datos, Metodología para la Elaboración de Sistemas Expertos, Aprendizaje Automatizado y Programación Descriptiva (PROGRAMA ANALÍTICO DE APRENDIZAJE AUTOMATIZADO, 2012).

En particular la asignatura Aprendizaje Automatizado, le permite al estudiante abordar problemas que dependan fuertemente de la experiencia, con elementos de aprendizaje, en particular de redes neuronales, algoritmos genéticos, entre otros (PROGRAMA ANALÍTICO DE APRENDIZAJE AUTOMATIZADO, 2012).

Los Algoritmos Genéticos (AG) son métodos de búsqueda de propósito general basados en los principios de la genética natural, es decir, son algoritmos de búsqueda basados en los mecanismos de la selección natural y la genética.

La Universidad José Martí Pérez de la provincia de Sancti-Spíritus (Uniss); cuenta con una facultad de ingeniería, en la cual se le imparte la asignatura Aprendizaje Automatizado a los estudiantes de la carrera de Informática. Profesores que imparten esta asignatura indican que a los estudiantes se les dificulta la comprensión de los Algoritmos Genéticos por ser un tema de alta complejidad. Las clases se imparten por los métodos tradicionales, con medios de enseñanzas tales como: pizarra, tiza, y presentaciones en power point. La bibliografía existente en estos momentos está actualizada, pero se encuentra mayormente en formato digital y podría enriquecerse aún más. Por lo general el estudiantado para ampliar sus conocimientos una vez concluidas las clases, sólo cuenta con el libro de texto donde aparece un tema que trata los algoritmos genéticos, y artículos en formato digital publicados en la plataforma interactiva Moodle; pero el hecho de no tener un texto específico que profundice en este tema hace que se dificulte la comprensión del mismo. Además no se cuenta con un software educativo que facilite el proceso de enseñanza - aprendizaje de este tema. Todo lo expresado anteriormente conforma la **situación problemática** de la presente investigación, de la cual se derivó el siguiente **problema de investigación**: ¿Cómo apoyar el proceso de enseñanza -

aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss?

Se propuso como **objetivo general** de esta investigación: Desarrollar un Tutorial que apoye el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss.

Para darle cumplimiento al objetivo surgieron las siguientes **preguntas de investigación**:

1. ¿Qué fundamentos teóricos y metodológicos sustentan la elaboración de una herramienta informática que apoye el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado?
2. ¿Cómo diseñar un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss?
3. ¿Cómo implementar un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss?

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

1. Determinación de los fundamentos teóricos y metodológicos que sustentan la elaboración de una herramienta informática que apoye el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.
2. Diseño de un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss.
3. Implementación de un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado para los estudiantes de Ingeniería Informática de la Uniss.

El trabajo está conformado por: introducción, tres capítulos, conclusiones, recomendaciones y bibliografía consultada. A continuación se presenta una breve descripción de cada capítulo:

**Capítulo 1. Fundamentación teórica y metodológica que sustenta la elaboración de un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.** En este capítulo se abordan la fundamentación teórica del tema y los conceptos asociados al negocio del problema. Se explica en detalles el problema a resolver y se describe el campo de acción donde se desarrolla, así como las tecnologías y metodologías utilizadas para su desarrollo teniendo en cuenta las tendencias actuales en el desarrollo de las TIC.

**Capítulo 2. Descripción de la solución propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.** Este capítulo se centra fundamentalmente en analizar el modelo del negocio, así como la descripción de dicho proceso utilizando los artefactos de UML. Además se muestran los diagramas y modelos de casos de uso utilizados en el sistema a construir, con su correspondiente descripción, así como los requisitos funcionales y no funcionales.

**Capítulo 3. Implementación de la aplicación propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.** En este capítulo se describe el diseño de la solución propuesta con sus correspondientes modelos y/o diagramas: modelo de clases del diseño, modelo de clases persistentes, modelo de despliegue y el modelo de datos. Los cuales ayudarán al entendimiento del funcionamiento del tutorial.

## Capítulo 1: Fundamentación teórica y metodológica que sustenta la elaboración de un Tutorial para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.

### 1.1 El proceso de Enseñanza-Aprendizaje.

Enseñanza y aprendizaje forman parte de un único proceso que tiene como fin la formación del estudiante. Enseñar es señalar algo a alguien, es mostrar lo que se desconoce.

Esto implica que hay un sujeto que conoce (el que puede enseñar), y otro que desconoce (el que puede aprender). El que puede enseñar, quiere y sabe enseñar (el profesor); El que puede aprender quiere y sabe aprender (el alumno). Ha de existir una disposición por parte de alumno y profesor. Aparte de estos agentes, están los contenidos, esto es, lo que se quiere enseñar o aprender (elementos curriculares) y los procedimientos o instrumentos para enseñarlos o aprenderlos (medios). Cuando se enseña algo es para conseguir alguna meta (objetivos). Por otro lado, el acto de enseñar y aprender acontece en un marco determinado por ciertas condiciones físicas, sociales y culturales (contexto) (Castellano, 2002).

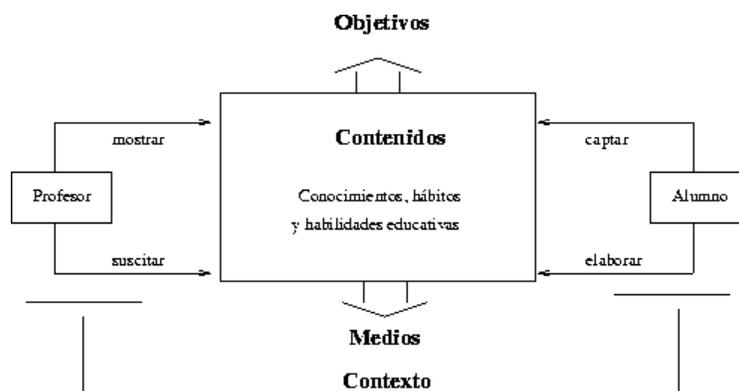


Figura 1: Elementos del proceso Enseñanza-Aprendizaje.

El proceso de enseñar es el acto mediante el cual el profesor muestra o suscita contenidos educativos (conocimientos, hábitos, habilidades) a un alumno, a través de unos medios, en función de unos objetivos y dentro de un contexto. El proceso de aprender es el proceso complementario de enseñar. Aprender es el acto por el cual un alumno intenta captar y elaborar los contenidos expuestos por el profesor, o por cualquier

otra fuente de información. Él lo alcanza a través de unos medios (técnicas de estudio o de trabajo intelectual). Este proceso de aprendizaje es realizado en función de unos objetivos, que pueden o no identificarse con los del profesor y se lleva a cabo dentro de un determinado contexto (Hernández, 1989).

El proceso de Enseñanza-Aprendizaje debe considerarse como un sistema estrechamente vinculado con la actividad práctica del hombre, que en última instancia, condiciona sus posibilidades de conocer, comprender y transformar la realidad objetiva.

### **1.1.1 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 (Campos, 2009).

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 (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 (Campos, 2009).

### **1.1.2 Mapas conceptuales.**

Son una técnica usada para la representación gráfica del conocimiento. Un mapa conceptual es una red de conceptos. En la red, los nodos representan los conceptos, y los enlaces las relaciones entre los conceptos. Los mapas conceptuales fueron desarrollados en la década de los 1960 por Joseph D. Novak, profesor emérito en la Universidad de Cornell, basándose en las teorías de David Ausubel del aprendizaje significativo. Según Ausubel "el factor más importante en el aprendizaje es lo que el sujeto ya conoce". Por lo

tanto, el aprendizaje significativo ocurre cuando una persona consciente y explícitamente vincula esos nuevos conceptos a otros que ya posee. Cuando se produce ese aprendizaje significativo, se produce una serie de cambios en nuestra estructura cognitiva, modificando los conceptos existentes, y formando nuevos enlaces entre ellos. Esto es porque dicho aprendizaje dura más y es mejor que la simple memorización: los nuevos conceptos tardan más tiempo en olvidarse, y se aplican más fácilmente en la resolución de problemas.

Los nuevos conceptos son adquiridos por descubrimiento, que es la forma en que los niños adquieren sus primeros conceptos y lenguaje, o por aprendizaje receptivo, que es la forma en que aprenden los niños en la escuela y los adultos. El problema de la mayor parte del aprendizaje receptivo en las escuelas, es que los estudiantes memorizan definiciones de conceptos, o algoritmos para resolver sus problemas, pero fallan en adquirir el significado de los conceptos en las definiciones o fórmulas (Novak & Roca, 2002).

Cuando se realiza un mapa conceptual, se obliga al estudiante a relacionarse, a jugar con los conceptos, a que se empape con el contenido. No es una simple memorización; se debe prestar atención a la relación entre los conceptos. Es un proceso activo (Novak & Roca, 2002).

### 1.1.3 Los mapas conceptuales en la enseñanza.

Los Mapas conceptuales resultan muy útiles en las diversas etapas del proceso educativo (Dávila & Martínez, 2000):

- En la planeación, como recurso para organizar y visualizar el plan de trabajo.
- En el desarrollo, como una herramienta que ayuda a los estudiantes a captar el significado de los materiales que pretenden aprender.
- En la evaluación, como recurso para la evaluación formativa. Permite “visualizar el pensamiento del alumno”, para así, corregir a tiempo posibles errores en la relación de los conceptos principales.

Entre las principales ventajas de la utilización de los mapas conceptuales en las actividades docentes se pueden citar las siguientes:

- Son fáciles de usar, la mayoría de los usuarios se convierten en expertos en una o dos horas.
- Pueden ser evaluados para revelar una eventual desinformación o falta de comprensión.
- Ofrecen una representación espacial de los contenidos, lo que ayuda a la memorización.
- Mejoran la sedimentación de los contenidos objeto de estudio.
- Muestran la interconexión de las ideas desde diversos puntos de vista, de la misma manera en la que diversas personas pueden concebirlas.

Está demostrado que cuando un estudiante sabe de antemano que su comprensión de la lección será evidenciada en un mapa, este hecho lo estimula a venir preparado a clase.

La presencia del mapa estimula la percepción visual y hace más interesante el aprendizaje.

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

Se puede denominar como Tecnologías de la Información y la Comunicación (TIC) el conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información (Romero, 2004).

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

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

Las definiciones ofrecidas de las nuevas tecnologías de la información y comunicación son diversas, variadas y algunas veces incluso contradictorias, entre

otros motivos por el simple hecho de que el propio término "nuevo" ya implica caducidad. Aunque suele existir un acuerdo en considerarlas como aquellos instrumentos técnicos que giran en torno a la información y a la comunicación, y a los nuevos descubrimientos que sobre las mismas se van originando (Borges, 2007).

### **1.2.1 Las TIC empleadas a la Educación.**

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 (Martínez, 2009).

Las TIC han incidido en todos los campos relacionados con la educación, facilitando la transformación y la optimización de la mayoría de los procesos administrativos, el desarrollo de metodologías innovadoras en los procesos de enseñanza-aprendizaje, el acceso a la formación superior de nuevos grupos de personas, así como una incipiente transformación del sistema organizativo de la sociedad (Conociendo algo de software educativo, 2009).

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

### **1.2.2 La computadora como medio de enseñanza.**

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 (Campos, 2009).

Una computadora con un buen software puede producir un aprendizaje con calidad con respecto a ciertos procedimientos de trabajo, pero también un mal uso de este puede traer aparejado dificultades de aprendizaje muy serias para el estudiante. En ocasiones hemos encontrado personas que se preguntan: ¿el uso de los medios y en particular de las computadoras originará automáticamente un aprendizaje efectivo?. Sencillamente la respuesta sin dudar es que NO. En este sentido, la forma en que se utilice, la motivación a proporcionar, el ajuste de las necesidades educativas, entre otras cuestiones, es lo que propiciará un mejor aprendizaje (Conociendo algo de software educativo, 2009).

### **1.2.3 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).

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 (Catoní, 2009).

### **1.2.3 Sistemas Tutoriales.**

García D. plantea que “constituye un programa especializado en la enseñanza de un dominio específico del conocimiento, apoyándose para ello en el diálogo con el estudiante, en la consolidación de un conjunto de aspectos esenciales que por su complejidad requieren de un nivel de abstracción que permita la representación adecuada del conocimiento” (Suares & Delgado, 2007).

Esta definición es retomada por Rodríguez, R. para puntualizar que: “el tutorial es un programa especializado en un área del conocimiento, que establece una estrategia basada en el diálogo, está de acuerdo a las características del estudiante y además, existe una estrategia pedagógica para guiar a este estudiante”. Los tutoriales pueden contener ejercicios y sesiones de preguntas y respuestas para presentar un tema y verificar su comprensión por parte del estudiante, permitiéndole también estudiar a su propio ritmo. En determinados momentos del programa se introducen elementos de evaluación cuyo resultado orienta al alumno a pasar a una lección siguiente o a repasar una anterior, proponiéndole, incluso, ejercicios de refuerzo (Rodríguez, 1991).

### **1.3 La inteligencia artificial.**

La inteligencia artificial es considerada una rama de la computación y relaciona un fenómeno natural con una analogía artificial a través de programas de computador. La inteligencia artificial puede ser tomada como ciencia si se enfoca hacia la elaboración de programas basados en comparaciones con la eficiencia del hombre, contribuyendo a un mayor entendimiento del conocimiento humano. Si por otro lado es tomada como ingeniería, basada en una relación deseable de entrada-salida para sintetizar un programa de computador. "El resultado es un programa de alta eficiencia que funciona como una poderosa herramienta para quien la utiliza." A través de la inteligencia artificial se han desarrollado los sistemas expertos que pueden imitar la capacidad mental del hombre y relacionan reglas de sintaxis del lenguaje hablado y escrito sobre la base de la experiencia, para luego hacer juicios acerca de un problema, cuya solución se logra con mejores juicios y más rápidamente que el ser humano. En la medicina tiene gran utilidad al acertar el 85 % de los casos de diagnóstico (Loaiza, 2009).

### 1.3.1 Los algoritmos genéticos.

Una de las herramientas de la Inteligencia Artificial aplicada para la resolución de distintos problemas son los Algoritmos Genéticos, que presentan marcados beneficios si se los compara con otras técnicas más tradicionales.

Un Algoritmo Genético es una herramienta basada en los procesos naturales que permiten la evolución de las especies en las poblaciones. Su principal fundamento es la Teoría de la Evolución de Darwin, complementado con otros conceptos y teorías más recientes de la Genética. Se pueden definir como métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a éstos los mismos métodos de la evolución biológica: selección basada en la población, reproducción sexual y mutación (Guervós, 2011).

### 1.3.2 Ventajas de los AG.

- Una clara ventaja es que los algoritmos genéticos son intrínsecamente paralelos, es decir, operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales. Esto significa que mientras técnicas tradicionales sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta subóptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo. Sin embargo, los algoritmos genéticos simplemente desechan esta solución subóptima y siguen por otros caminos (Peña & Truyol, 2008).
- Otra ventaja es su habilidad para manipular muchos parámetros simultáneamente. Resulta interesante en caso de tener varios objetivos a resolver.
- No necesitan conocimientos específicos sobre el problema que intentan resolver. Realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora o no.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

### 1.3.3 Aplicaciones de los AG.

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes. Sin embargo, no todos los problemas pudieran ser apropiados para esta técnica. Se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla (Peña & Truyol, 2008):

- Su espacio de búsqueda debe estar delimitado dentro de un cierto rango.
- Debe poderse definir una función de aptitud que nos indique qué tan buena o mala es una cierta respuesta.
- Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

Dentro de los distintos problemas de optimización podemos encontrar unas áreas de aplicación:

- Diseño por computadora de nuevos materiales que cumplan múltiples objetivos.
- Optimización de la carga de containers.
- Asignación de procesos en topologías de redes con procesamiento distribuido.
- Ubicación de archivos en sistemas de almacenamiento distribuido.
- Diseño de circuitos integrados.
- Optimización de la infraestructura de telefonía celular.
- Ingeniería Aeroespacial.
- Juegos.
- Robótica.

### 1.3.4 Enseñanza - aprendizaje de Algoritmos Genéticos.

En Escobar et al. (2008) se detalla el diseño y desarrollo de un artefacto educativo destinado a apoyar la comprensión de los contenidos fundamentales de los algoritmos genéticos (AG) (Nunes de Miranda, 2000; Nilsson, 2001 y Russell y Norvig, 2004). Se

dispone de materiales didácticos en múltiples formatos. Los simuladores ilustran el desempeño de los principales operadores de los AG, logrando así una mayor comprensión de los conceptos relacionados a esta técnica de la IA. En futuros trabajos se expondrá el empleo de simuladores de otros operadores, así como también, innovadoras alternativas de tratamiento y presentación de los contenidos tendientes a mejorar la comprensión de los mismos (Mariño, 2009).

### **1.4 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.

#### **1.4.1 La Metodología RUP.**

El Proceso Unificado de Desarrollo (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto al UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (Jacobson, Booch, & Rumbaugh, 2006).

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

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos (Jacobson, Booch, & Rumbaugh, 2006):

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Surgen nuevos requisitos a ser analizados.
- **Mantenimiento:** una vez instalado el producto, el usuario realiza requerimientos de ajuste, esto se hace de acuerdo a solicitudes generadas como consecuencia del interactuar con el producto.

### 1.4.2 ¿Por qué la metodología RUP?

El RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, entre otros) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso) (Jacobson, Booch, & Rumbaugh, 2006).

### 1.4.3 Lenguaje Unificado de Modelado (UML).

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje estándar para escribir planos de software, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de softwares (Jacobson, Booch, & Rumbaugh, 2006).

UML es apropiado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la Web, e incluso para sistemas empotrados de tiempo real muy exigentes. Es un lenguaje muy expresivo, que cubre todas las vistas

necesarias para desarrollar y luego desplegar tales sistemas. Detrás de cada símbolo en la notación UML hay una semántica bien definida. De esta manera, un desarrollador puede escribir un modelo en UML y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad (Jacobson, Booch, & Rumbaugh, 2006).

Los tres elementos que forman el modelo conceptual de UML son: los bloques básicos de construcción del lenguaje, las reglas que se aplican sobre esos bloques y los mecanismos comunes de UML (Laguna, 2004).

## Bloques de construcción

Existen tres tipos de bloques de construcción:

**Elementos:** Son los modelos UML.

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

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

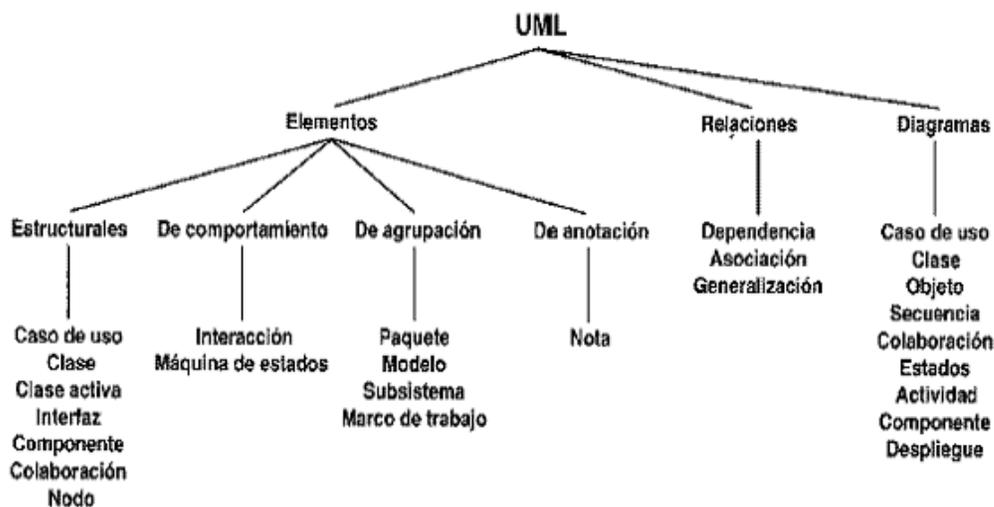


Figura 2. Elementos fundamentales de UML.

## Reglas.

Los bloques de construcción se deben combinar siguiendo las normas que establece UML. UML establece una serie de normas sobre como nombrar a los elementos, relaciones y diagramas; la visibilidad y alcance de dichos nombres y sobre su integridad

(como diseñar relaciones consistentes). Podremos decir que un modelo está bien formado cuando cumpla estas reglas.

### **Mecanismos comunes.**

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

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

**Adornos:** Todos los elementos gráficos en UML poseen una notación gráfica que representa los aspectos más importantes de éstos; aún así existe una notación que permite representar los aspectos más específicos a través de adornos, consiguiendo así una especificación aún más detallada.

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

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

- **Estereotipo:** Permite crear nuevos bloques de construcción partiendo de los ya existentes.
- **Valor etiquetado:** Permite añadir nuevas propiedades a un bloque de construcción.
- **Restricción:** Permite añadir o modificar las reglas referidas a los bloques de construcción.

### 1.4.4 ¿Por qué UML?

La decisión de utilizar como notación para el desarrollo del software se debe a que se ha convertido en un estándar que tiene las siguientes características (Jacobson, Booch, & Rumbaugh, 2006):

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

### 1.4.5 Arquitectura en N Capas.

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada (Pelaez, 2009).

Cuando se habla de la arquitectura en capas casi siempre se utilizan 3 capas: capa de datos, capa de negocio y capa de interfaz de usuario. Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario. El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular y gracias a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo (Carrasco, 2006).

### **1.4.6 Arquitectura Cliente-Servidor.**

Es un modelo de hardware y software adecuado para el proceso distribuido, en el que la comunicación se establece de uno a varios. Un proceso es un programa en ejecución. Proceso cliente es el que solicita un servicio. Proceso servidor es el capaz de proporcionar un servicio (Cáceres, 2005).

Un proceso cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos pueden ejecutarse en la misma máquina o en distintas máquinas comunicadas a través de una red. Por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, a la que se denomina Terminal o cliente, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en una máquina remota, denominada de forma abreviada, simplemente, servidor (Arquitectura Cliente/Servidor, 2007). □

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo

### **Los cambios en el servidor implican pocos o ningún cambio en el cliente.**

La arquitectura Cliente/Servidor es una infraestructura versátil, modular y basada en mensajes que mejora la portabilidad, la interoperabilidad y la escalabilidad de los sistemas (Cáceres, 2005):

Una aplicación cliente/servidor típica es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. Los sistemas distribuidos pueden consistir en diversos servidores que alojen datos, de forma que el cliente no tiene por qué conocer exactamente dónde se encuentran, simplemente hace una petición de servicio, y es el sistema servidor el encargado de localizarlos y proporcionar el resultado de la consulta al usuario que hizo la petición (Arquitectura Cliente/Servidor, 2007).

### 1.4.7 Modelo vista controlador (MVC).

En el diseño de aplicaciones con sofisticadas interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador (MVC). Pues la lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica del negocio. Por lo que si se realiza un diseño que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando se necesite cambiar la interfaz, se tendrán que modificar los componentes de negocio, siendo esto trabajoso y engorroso, además de provocar posibles errores en el sistema. Por lo que se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio (Lago, 2007).

Elementos del patrón:

- **Modelo:** datos y reglas de negocio
- **Vista:** muestra la información del modelo al usuario
- **Controlador:** gestiona las entradas del usuario

## 1.5 Tecnologías web.

### 1.5.1 CSS.

CSS es una tecnología que permite controlar la presentación de los documentos en la Web. Las Hojas de Estilo en Cascada o CSS constituyen un lenguaje sencillo que complementa el de HTML, suponiendo un apoyo fundamental a la hora de diseñar páginas Web, porque permiten una mayor precisión en el ajuste de los elementos de diseño. Esta técnica consiste en separar el diseño del contenido, de manera que las indicaciones para conformar el diseño se agrupan en una hoja de estilo o archivo que reside fuera del contenido del documento de la pagina HTML (Cáceres, 2005).

Lo que hace fundamentalmente el código de las hojas de estilos es transformar las etiquetas del lenguaje HTML y conformarlas a las características que se quiera darle; pero también, y esto es lo importante, con este código se pueden crear etiquetas nuevas, que se introducen dentro del documento. Una de las ventajas de las hojas de estilos es que se puede modificar algunas características de todos los documentos de un sitio Web desde un archivo, sin tener que modificarlas en cada uno de los documentos (Cáceres, 2005).

### 1.5.2 ¿Por qué CSS?

Las Hojas de Estilo en Cascada o CSS (Cascade Style Sheets), proporcionan una manera muy eficiente de aplicar estilos y formato a las páginas web y a prácticamente cualquiera de sus elementos. Además, permiten definir nuevas clases de estilos y subclases de etiquetas HTML o de otras clases, Es la mejor forma de separar los contenidos y su presentación, reduce la complejidad de su mantenimiento, permite visualizar el mismo documento en infinidad de dispositivos diferentes, un cambio en un fichero genera cambios en todos los ficheros y permite especificar la apariencia de todo el sitio web en un único lugar, los estilos para fuentes, tamaños de la fuente, espaciado de letras, colores, y márgenes. Estas opciones la convierten en una herramienta muy poderosa y muy flexible (Ruiz, 2006).

### 1.5.3 Java Script.

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Java Script. El código Java Script puede ser integrado dentro de las páginas web. Los script tienen capacidades limitadas, por razones de seguridad, el código visible por cualquier usuario y el código deben descargarse completamente (Álvarez, 2005).

### 1.5.4 ¿Por qué Java Script?

Es un lenguaje de tipo script compacto, basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Ofrece la posibilidad de agregar un poco de dinamismo y de interactividad en un sitio Web (Cáceres, 2005).

Características del lenguaje:

- Fue desarrollado para incrementar las funcionalidades del lenguaje HTML.
- Es un lenguaje interpretado puro (no requiere compilación, ni generación de intermedios codificados de ningún tipo). El navegador del cliente es el encargado de interpretar las instrucciones Java Script y ejecutarlas.
- Es un lenguaje de programación sencillo y pensado para hacer las tareas con rapidez. Incluso personas que no tengan una experiencia previa en la

programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con solo un poco de práctica.

- Se utiliza embebido en el código documentos HTML. Fuera de ellos no tiene vigencia, o sea, no crea aplicaciones autónomas.
- Puede ejecutarse en el cliente evitando la necesidad de realizar peticiones continuamente al servidor, por lo que descarga el trabajo de este y hace que la navegación sea más rápida.

### **1.5.5 HTML.**

HTML significa Hiper Text Markup Language, cuya traducción sería Lenguaje de etiquetas e hipertexto, en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. Este lenguaje está compuesto por etiquetas o marcas, y gracias a ellas es posible dar forma a todos los componentes de una página. HTML no es propiamente un lenguaje de programación sino un sistema de etiquetas, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. El entorno para trabajar HTML es simplemente un procesador de texto. Estos documentos pueden ser mostrados por los visores o navegadores de páginas Web en Internet, como Microsoft Internet Explorer (Ruiz, 2006).

### **1.5.6 ASP.**

ASP es una tecnología del lado del servidor creada por Microsoft, ofrecida como componente gratuito de Internet Information Services, con la que se pueden desarrollar sofisticadas aplicaciones Web que los usuarios pueden correr desde cualquier navegador. La utilidad más practicada de las páginas ASP es la facilidad para conectar con una base de datos y extraer datos de la misma dinámicamente visualizándolos en el navegador. Puede conectarse a gestores de Base de datos SQL Server, Microsoft Access, Oracle, Sybase, DB2, FoxPro, dBase, Informix o cualquier otro motor que disponga de driver ODBC (Cáceres, 2005).

Para procesar una página ASP no existe restricción especial en el lado del cliente, por lo que es indiferente la utilización del navegador, Internet Explorer o Netscape Communicator. Sin embargo, en el lado del servidor, es necesario un servidor Web de Microsoft, debido a que ASP es una tecnología propietaria de Microsoft. Se utiliza el archivo ASP.DLL

para interpretar el código, siendo el servidor más extendido Internet Information Server (más conocido como IIS). ASP también puede correr en un servidor con Linux, siempre que tenga instalado el correspondiente programa que lo haga funcionar, algunos de los más conocidos son Instant ASP y Chilisoft (Cáceres, 2005).

### **1.5.7 PHP.**

PHP nació como un lenguaje sencillo para páginas personales (de ahí sus siglas en inglés, Personal Home Page) y en muy poco tiempo se convirtió en una de las tecnologías más utilizadas en la Web. PHP trae una gran cantidad de funciones predefinidas para trabajar con las tecnologías más comunes como Apache, MySQL, Oracle,... etc. Todo esto hace que programar cualquier tipo de aplicación sea muy sencillo (Ruiz, 2006).

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML (Munz, 2006).

PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML. Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente. El cliente no ve el código PHP sino los resultados que produce. No es necesario que su navegador lo soporte, este es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP (Valdés, 2007).

### **1.5.8 ¿Por qué PHP?**

Luego de hacer el estudio de las diferentes tecnologías, en especial un análisis entre PHP y ASP, se decide utilizar el PHP ya que: Está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que ASP por ser propiedad de Microsoft no es multiplataforma. El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sinnúmero de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas estas se pueden adquirir gratis en Internet. En el caso de ASP forma parte del Internet Information Server que viene integrado en Windows NT

2000 Server con su elevado costo de adquisición. PHP y ASP son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma (Valdés, 2007).

### **1.5.9 Servidor Web Apache.**

Apache tal vez no sea uno de los servidores web más fáciles de configurar, pero una vez instalado, Apache se convertirá en el servidor más potente y estable que podamos imaginar. Es código abierto (open source, en inglés), lo que significa que su código es libre y gratuito. Además, es el servidor más utilizado en Internet. Servidor web que por defecto lo traen instalado en todas las distribuciones Linux. También existe para otras plataformas incluso Windows. Su funcionamiento básico es ejecutando un proceso padre y tantos procesos hijos como peticiones reciba para atender a cada cliente. Apache es usado para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web (Ruiz, 2006).

#### **Ventajas:**

- Multi-plataforma
- El software es gratuito.
- Popular (fácil conseguir ayuda/suporte).
- Estándar del sector para la mayoría de servidores web.
- Permite la administración remota.
- Tiene interfaz con todos los sistemas de autenticación.
- Tiene integración estándar del protocolo de seguridad SSL.
- Provee interfaz a todas las bases de datos.

#### **Desventajas:**

- Instalación en modo consola.
- Apache no se actualiza regularmente.
- Requiere un conocimiento más técnico para instalar y configurar.

### **1.6 Sistemas Gestores de Bases de Datos.**

Los Sistemas de Gestión de Base de Datos (en inglés Data Base Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de

definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta (Valdés, 2007).

La función fundamental de un SGBD es proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos.

### **1.6.1 SQL.**

SQL (Structured Query Language, Lenguaje de Consultas Estructurado) nos permite "hablar" con las bases de datos para pedirle que nos entregue un grupo de registros que cumplan requisitos determinados y en un orden dado. Además, también podemos crear nuevas bases de datos, nuevas tablas, nuevos campos, nuevos registros y realizar toda clase de modificaciones y eliminaciones (Ruiz, 2006).

### **1.6.2 SQL Server 2000.**

SQL Server 2000 es un sistema de gestión de bases de datos relacionales (SGDBR o RDBMS: Relational Database Management System) diseñado para trabajar con grandes cantidades de información y con la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios Web. Ofrece el soporte de información para las tradicionales aplicaciones Cliente/Servidor, las cuales están conformadas por una interfaz a través de la cual los clientes acceden a los datos por medio de una LAN. La hoy emergente plataforma .NET exige un gran porcentaje de distribución de recursos, desconexión a los servidores de datos y un entorno descentralizado, para ello sus clientes deben ser livianos, tales como los navegadores de Internet, los cuales accederán a los datos por medio de servicios como el Internet Information Services (Enríquez, 2007).

### **1.6.3 PostgreSQL.**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto (Espinoza, 2005).

### **1.6.4 MySQL.**

MySQL Server es la base de datos de código fuente abierto más usada del mundo desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo

negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. Su origen se debió a la búsqueda por parte de los fundadores de crear un manejador de bases de datos que fuera "rápido", todavía más rápido. Así surgió MySQL, primero como un producto de la empresa y después como software de dominio público. MySQL ofrece un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet. MySQL es una solución para empresas que buscan una vía rápida de despliegue de bases de datos de código libre, dando garantías en la calidad del soporte, el nuevo MySQL está diseñado para servicios informativos de gran envergadura con un bajo costo (Castellanos, 2005).

### **1.6.5 ¿Por qué MySQL?**

MySQL es un servidor de bases de datos multiusuario, concretamente, el más rápido en entornos web. El principal objetivo de MySQL es velocidad y robustez, es multiproceso, es decir puede usar varias CPU si éstas están disponibles. Además puede trabajar en distintas plataformas y Sistemas Operativos distintos (Castellanos, 2005).

## **1.7 Herramientas de desarrollo.**

### **1.7.1 Macromedia Dreamweaver.**

Dreamweaver es la herramienta de diseño de páginas Web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar: hojas de estilo y capas, Javascript para crear efectos e interactividades e inserción de archivos multimedia (Álvarez, 2005).

### **1.7.2 Rational Rose.**

Ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose desde el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto. Esta herramienta CASE propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite especificar, analizar y diseñar el sistema antes de codificarlo. Mantiene la

consistencia de los modelos del sistema con el software. Posibilita la generación de documentación automáticamente. Permite la generación de código a partir de los modelos o ingeniería inversa (crear modelo a partir código).

### **1.7.3 CmapTools.**

CmapTools se diseñó con el objetivo de apoyar la construcción de modelos de conocimiento representados en forma de “Mapas Conceptuales” aunque también pueden elaborarse con él “Telarañas”, “Mapas de Ideas” y “Diagramas Causa-Efecto”, todos dentro de un entorno de trabajo intuitivo, amigable y fácil de utilizar. CmapTools reúne todas las características que debe tener un software para cumplir con los objetivos de aprendizaje a través de las tecnologías de la Información y las Comunicaciones. Proporciona algunas libertades que permiten incluir programación al mapa creado. El hecho de que CmapTools no sea un software de código abierto, imposibilita incluir dentro de él la programación mencionada. Existen otras herramientas con este mismo fin que son libres y por tanto se tiene acceso al código, pero tienen otras limitaciones para hacer sistemas de enseñanza amigables y poderosos. No obstante el señalamiento anterior, CmapTools es gratuito y posee una licencia que ofrece facilidades para aplicaciones con fines docentes (Cañas & Ford, 2008).

### **1.7.4 Exelearning.**

Exelearning es un programa tipo open source (recursos libres o gratuitos); una creación de aplicación que permite a profesores y académicos la publicación de contenidos didácticos en soportes informáticos (CD, memorias USB, en la web), sin necesidad de ser ni convertirse en expertos en HTML o XML.

Los recursos creados en EXelearning pueden exportarse en formatos de paquete de contenido de IMS, SCORM 1.2 o IMS Common Cartridge o como simples páginas web independientes.

### **1.7.5 Server2Go.**

Es una aplicación que instala un servidor web. Puede ser instalado y utilizado desde dispositivos tales como tarjetas de memoria, pendrives, CD, DVD o desde un disco duro. Pero su principal punto fuerte es el de poder ser llevado a cualquier sitio y ejecutado desde allí. Además de su facilidad de instalación y ser gratuito, Server2Go, soporta PHP 5, MySQL 5, SQLite y Pearl 5.8. Su sistema está basado en Apache y está disponible para Windows (98, XP, Vista y 7) y Mac OS. Soporta las versiones posteriores de Internet

## Capítulo I

Explorer 6, Firefox, Safari, Camino y algunos otros navegadores. Detecta automáticamente el nombre del host y el puerto cuando iniciamos el navegador, se apaga automáticamente cuando se cierra el navegador y además soporta PHP 5 con múltiples extensiones instaladas (Sitio oficial de Server2Go, 2007).

### **Capítulo 2. Descripción de la solución propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado.**

#### **Introducción**

En este capítulo se exponen las reglas del negocio a tener en cuenta para el diseño del sistema, una caracterización de los actores y trabajadores del mismo, sus casos de uso, conformándose el diagrama de casos de uso del negocio, la descripción de estos en formato expandido, y se muestran los diagramas de actividades y del modelo objetos. Además se describe de modo general el funcionamiento del sistema diseñado. Se definen los requerimientos funcionales y no funcionales, se realiza la descripción del modelo de casos de uso del sistema, basándose en los actores y los casos de uso.

#### **2.1 Identificación de los procesos del negocio.**

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y forma, que emplean los recursos de la organización para dar resultados que apoyen sus objetivos. El modelo del negocio está soportado por dos tipos de modelos de UML: modelo de casos de uso y modelo de objetos (Jacobson, Booch, & Rumbaugh, 2006).

A partir del planteamiento anterior se realizó un estudio de la organización en cuestión y el proceso identificado para ser modelado fue el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado en los estudiantes de Ingeniería Informática de la Uniss.

##### **2.1.1 Reglas del Negocio:**

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio (Jacobson, Booch, & Rumbaugh, 2006).

- El profesor debe orientarle ejercicios y ejemplos al estudiante para el desarrollo de sus conocimientos.
- El estudiante debe consultar la bibliografía para la realización realizar el estudio.
- El profesor debe evaluar los conocimientos del estudiante.

##### **2.1.2 Modelo de casos de uso del negocio.**

El modelo de casos de uso del negocio describe los procesos del negocio de una empresa en términos de casos de uso y actores del negocio que se corresponden con los procesos y clientes respectivamente. El modelo de casos de uso del negocio presenta un

sistema (en este caso, el negocio) desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. (Jacobson, Booch, & Rumbaugh, 2006).

El modelo de Casos de Uso del Negocio es definido a través de tres artefactos:

1. El diagrama de casos de uso del negocio,
2. La descripción de los casos de uso del negocio
3. El diagrama de actividades de cada caso de uso del negocio.

### 2.1.3 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Jacobson, Booch, & Rumbaugh, 2006).

**Tabla 1. Descripción del actor del negocio.**

Actor	Descripción
Estudiante	Para el estudio, el estudiante debe consultar la bibliografía, que consiste en libros de texto en formato duro, textos electrónicos, y materiales complementarios que tiene a su disposición en la plataforma interactiva Moodle.

### 2.1.4 Trabajadores del negocio.

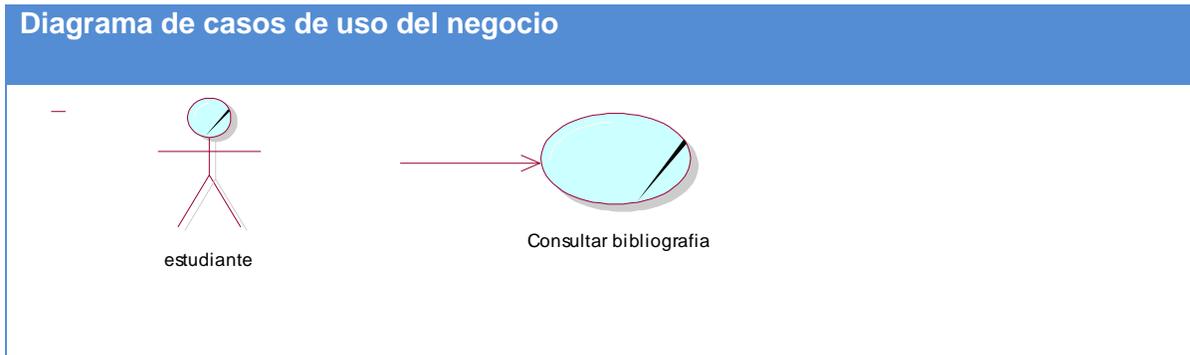
Un trabajador es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades (Jacobson, Booch, & Rumbaugh, 2006).

**Tabla 2. Descripción del trabajador del negocio.**

Trabajador	Descripción
Profesor	Es el encargado de facilitarle la bibliografía al estudiante.

**2.1.5 Diagrama de casos de uso del negocio.**

Los diagramas de casos de uso se utilizan para especificar las funcionalidades y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O sea es un diagrama que muestra la relación entre los actores y los casos de uso en un sistema (Jacobson, Booch, & Rumbaugh, 2006).



**Figura 3. Diagrama de casos de uso del negocio.**

**2.1.6 Descripción de los Casos de Uso del negocio.**

Luego de identificar el proceso del negocio es necesario explicarlo detalladamente. A continuación se muestra la descripción del mismo.

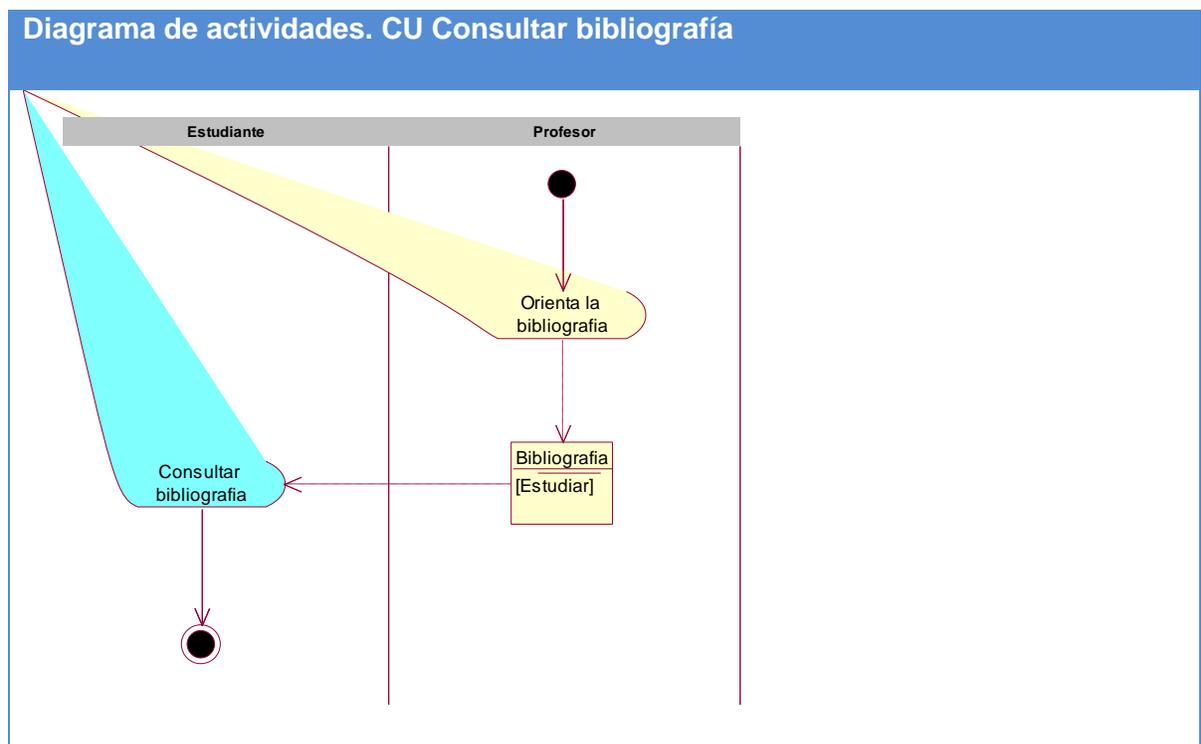
**Tabla 3. Descripción del caso de uso consultar bibliografía.**

Caso de Uso del Negocio		Consultar bibliografía
<b>Actor</b>	Estudiante	
<b>Propósito</b>	Consultar la bibliografía orientada para realizar el estudio.	
<b>Resumen</b>		
El caso de uso se inicia cuando el Profesor orienta la bibliografía a consultar y sitúa los materiales en formato electrónico en el Moodle. El Estudiante revisa su texto en formato duro y busca en la plataforma interactiva la información publicada por el profesor.		
<b>Curso Normal de los eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del negocio</b>	
	1. El Profesor orienta la bibliografía a consultar y sitúa los materiales en	

	formato electrónico en el Moodle.
2. El Estudiante revisa su texto en formato duro y busca en la plataforma interactiva la información publicada por el profesor y realiza el estudio.	
<b>Prioridad</b>	alta

**2.1.7 Diagrama de actividades.**

El diagrama de actividad es un grafo que contiene los estados en que puede hallarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio. (Jacobson, Booch, & Rumbaugh, 2006).



**Figura 4. Diagrama de actividades. CU Consultar bibliografía.**

**2.1.8 Modelo de objetos del negocio.**

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores

que utilizan un conjunto de entidades del negocio y unidades de trabajo. Una entidad del negocio representa algo, que los trabajadores toman, inspeccionan, manipulan, producen o utilizan en un caso de uso del negocio (Jacobson, Booch, & Rumbaugh, 2006).

### Diagrama del modelo de objetos.

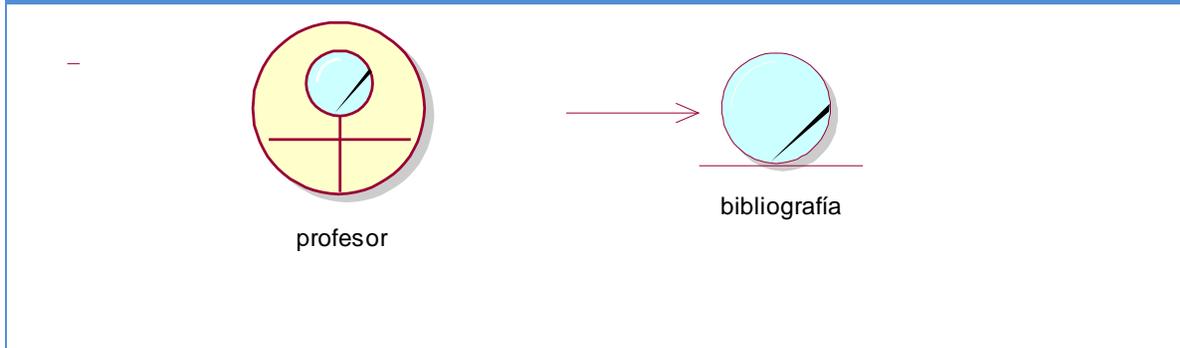


Figura 5. Diagrama del modelo de objetos.

## 2.2 Especificación de los requisitos de software.

### 2.2.1 Listado de los requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo (Jacobson, Booch, & Rumbaugh, 2006).

Teniendo en cuenta el criterio de los autores, el sistema propone los siguientes requerimientos funcionales:

RF -1. Gestionar usuario

RF-1.1 Adicionar usuario

RF-1.2 Modificar usuario

RF-1.3 Eliminar usuario

RF-1.4 Cambiar contraseña

RF-2. Entrar al sistema

RF-2.1 Iniciar sesión

RF-2.2 Crear cuenta

RF-2.3 Cerrar sesión

RF-3. Gestionar ejercicio resuelto

RF-3.1 Insertar ejercicio resuelto

RF-3.2 Modificar ejercicio resuelto

- RF-3.3 Eliminar ejercicio resuelto
- RF-4. Mostrar ejercicio resuelto
- RF-5. Gestionar ejercicios propuestos
  - RF-5.1 Insertar ejercicio propuesto
  - RF-5.2 Modificar ejercicio propuesto
  - RF-5.3 Eliminar ejercicio propuesto
- RF-6. Mostrar ejercicio propuesto
- RF-7. Resolver autoexamen
- RF-8. Reportar usuarios del sistema.
- RF-9. Reportar estudiantes del sistema.
- RF-10. Mostrar resultado de examen.
- RF-11. Mostrar mapas conceptuales.
- RF-12. Reportar visitas a recursos.

### **2.2.2 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 (Jacobson, Booch, & Rumbaugh, 2006).

### **Requerimientos de Apariencia o Interfaz Externa**

La interfaz estará diseñada de modo 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.

### **Usabilidad**

El producto está orientado para ser usado por diferentes tipos de usuarios, entre los que se encuentran aquellas personas que tengan poca experiencia en el uso de la computadora y que 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 y cuadros de texto).

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

### **Rendimiento**

El sistema deberá funcionar de manera estable evitando errores que conciernan directamente a su programación. Los tiempos de respuestas deben ser cortos ante las solicitudes de los usuarios y en el procesamiento de la información. No se requiere de una capacidad de procesamiento alta, pues el Tutorial no ejecutará algoritmos complejos.

### **Soporte**

Se requiere un servidor de bases de datos para el soporte de información, con velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes. Se documentará la aplicación para garantizar su soporte. El administrador tendrá la responsabilidad de mantener actualizada la aplicación. Las pruebas del sistema se realizarán en la UNISS. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto. El sistema deberá permitir futuras mejoras y nuevas opciones que se le quieran incorporar.

### **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 requisitos del momento.

### **Seguridad**

La información manejada por el sistema garantiza la protección contra acciones no autorizadas o que puedan afectar la integridad de los datos, pues solo la puede modificar el administrador (es).

No se permitirán accesos sin autorización al sistema, para ello se definirá una política de usuarios con roles y privilegios diferentes para que la información pueda ser consultada por las personas autorizadas. Para ello el sistema contará con dos tipos de usuarios, el administrador y los usuarios, que según el rol que tengan asignado en el sistema será el acceso a la parte de la información que le corresponde.

### **Políticos-culturales**

El nivel cultural, social o técnico; no determinará una prioridad o limitante a la hora de brindar los servicios que ofrece el producto.

### **Legales**

Este sistema tiene entera legalidad y aprobación por el personal del Instituto puesto que no constituye ninguna violación a leyes y la información es correctamente trabajada. Es para uso interno, por lo que no tiene problemas con las licencias de los software utilizados en su confección que nuestro país no puede adquirir producto del bloqueo.

### **Confiability**

Es importante que es sistema presente un mecanismo de respuesta rápida ante fallos y que en caso de ocurrencia se minimicen las pérdidas de información.

### **Ayuda y documentación en línea.**

Tendrá una ayuda en línea. Dispondrá de una ayuda bien detallada sobre las principales opciones del sistema.

### **Software**

El sistema se desarrollará con tecnología PHP y gestor de base de datos MySQL para servidores Web Apache específicamente XAMPP, permitiendo su instalación en plataformas Windows y Linux. Los clientes podrán acceder desde diferentes navegadores Web, en diferentes sistemas operativos.

### **Hardware**

Los usuarios que deseen acceder al sistema lo podrán hacer desde estaciones de trabajo clientes con los siguientes requerimientos mínimos: Microprocesador, Pentium 3 a 1,86GHz 783 MHz de velocidad o similares. Memoria física, 256MB de memoria RAM y 20 Gbytes de Capacidad de disco duro.

## **2.3 Descripción del sistema propuesto.**

### **2.3.1 Modelo de casos de uso del sistema.**

El modelado de Casos de Uso es la técnica más simple y que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. El modelo de casos de uso consiste en actores y casos de uso. Los actores

representan usuarios y otros sistemas que interactúan con el sistema y los casos de uso presentan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor (Jacobson, Booch, & Rumbaugh, 2006).

### 2.3.2 Actores del sistema.

Un actor no es más que un rol que juega un usuario de Caso de Uso cuando interactúan con estos casos de uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que se han identificado los actores del sistema, se ha identificado el entorno externo del sistema (Jacobson, Booch, & Rumbaugh, 2006).

Teniendo en cuenta lo planteado anteriormente se definieron los siguientes actores del sistema:

**Tabla 4. Descripción de los actores del sistema.**

Actor	Descripción
Estudiante	Es la persona que utiliza el sistema para buscar información y realizar el estudio. Este tiene acceso limitado al sistema.
Profesor	Es la persona que inserta la información en el sistema. Aunque tiene acceso limitado al sistema posee mayores privilegios que el estudiante.
Administrador	Es la persona que puede interactuar con todas las funcionalidades del sistema y posee mayores privilegios que el estudiante y el profesor.

### 2.3.3 Casos de uso del sistema.

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Jacobson, Booch, & Rumbaugh, 2006).

Se decide realizar el diagrama de casos de uso definiendo paquetes para subdividir éste, debido a que el diagrama es extenso y se hace difícil su comprensión. Se muestra un diagrama por cada paquete.

Sobre los paquetes del sistema Pressman expresa: Subdividir los casos de uso en paquetes resulta de mucha ayuda en la modelación de cualquier sistema informático. Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí. (Pressman, 2007).

Se realizaron 3 paquetes: Seguridad, Ejercicios y Reportes. El paquete de Reportes y el de Ejercicios dependen del paquete de Seguridad y a su vez el de Reportes depende del paquete de Ejercicios.

A continuación se muestran los paquetes y sus respectivos casos de usos que se identificaron para el desarrollo de la aplicación.

### 1. **Paquete Seguridad.**

- 1.1 Modificar perfil.
- 1.2 Crear cuenta.
- 1.3 Autenticar usuario.
- 1.4 Cambiar contraseña.
- 1.5 Gestionar usuario.

### 2. **Paquete Ejercicios.**

- 2.1 Gestionar ejercicios propuestos.
- 2.2 Gestionar ejercicios resueltos.
- 2.3 Realizar autoexamen.

### 3. **Paquete Reportes.**

- 3.1 Reportar usuarios del sistema.
- 3.2 Ver recursos visitados.
- 3.3 Reportar estudiantes.
- 3.4 Ver resultados de examen.
- 3.5 Consultar ejercicios propuestos.
- 3.6 Consultar ejercicios resueltos.

Diagramas de casos de uso del sistema por paquetes.

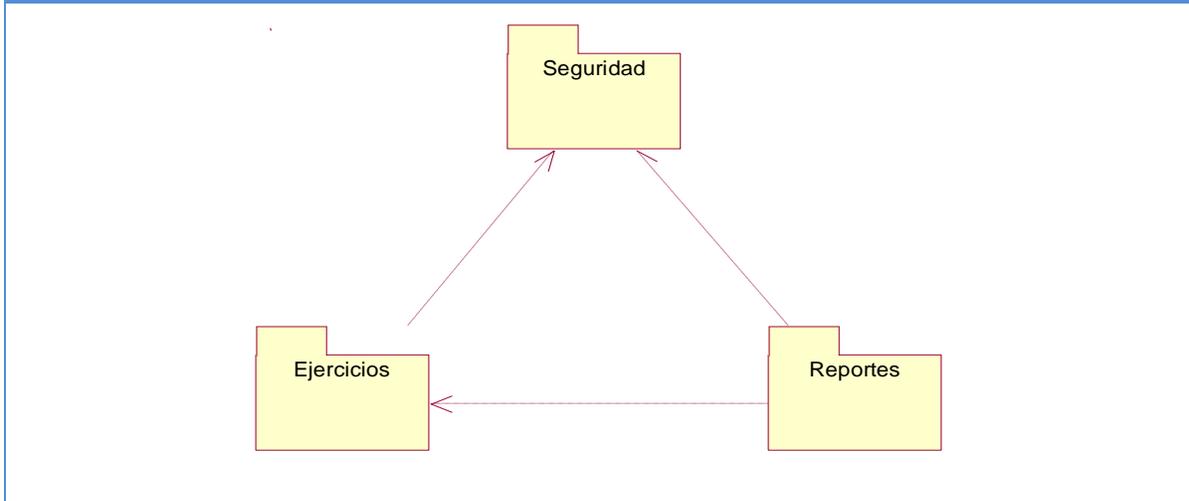


Figura 6. Diagrama de casos de uso del sistema por paquetes.

Diagramas de casos de uso. Paquete seguridad.

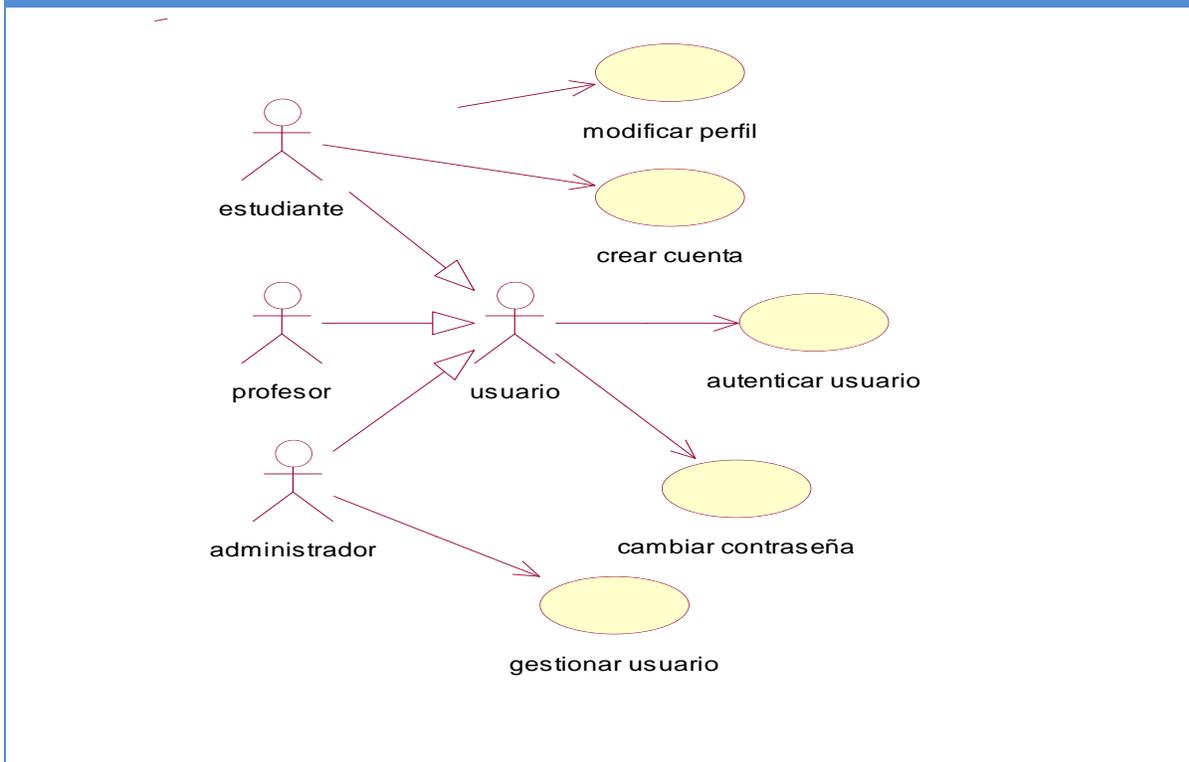


Figura 7. Diagrama de casos de uso. Paquete seguridad.

Diagramas de casos de uso. Paquete ejercicios.

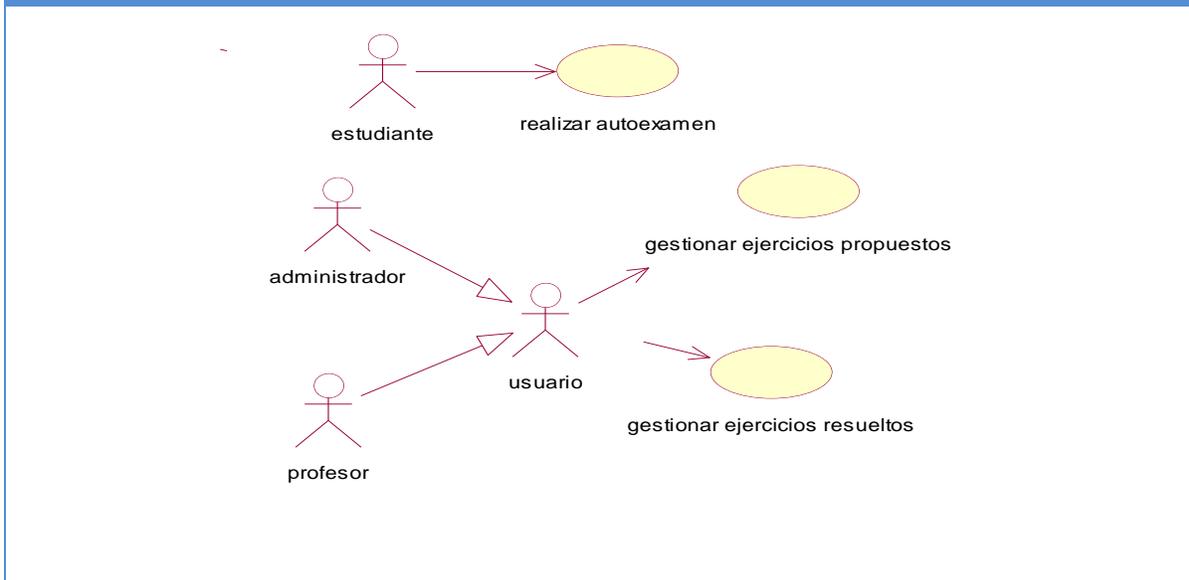


Figura 8. Diagrama de casos de uso. Paquete ejercicios.

Diagramas de casos de uso. Paquete reportes.

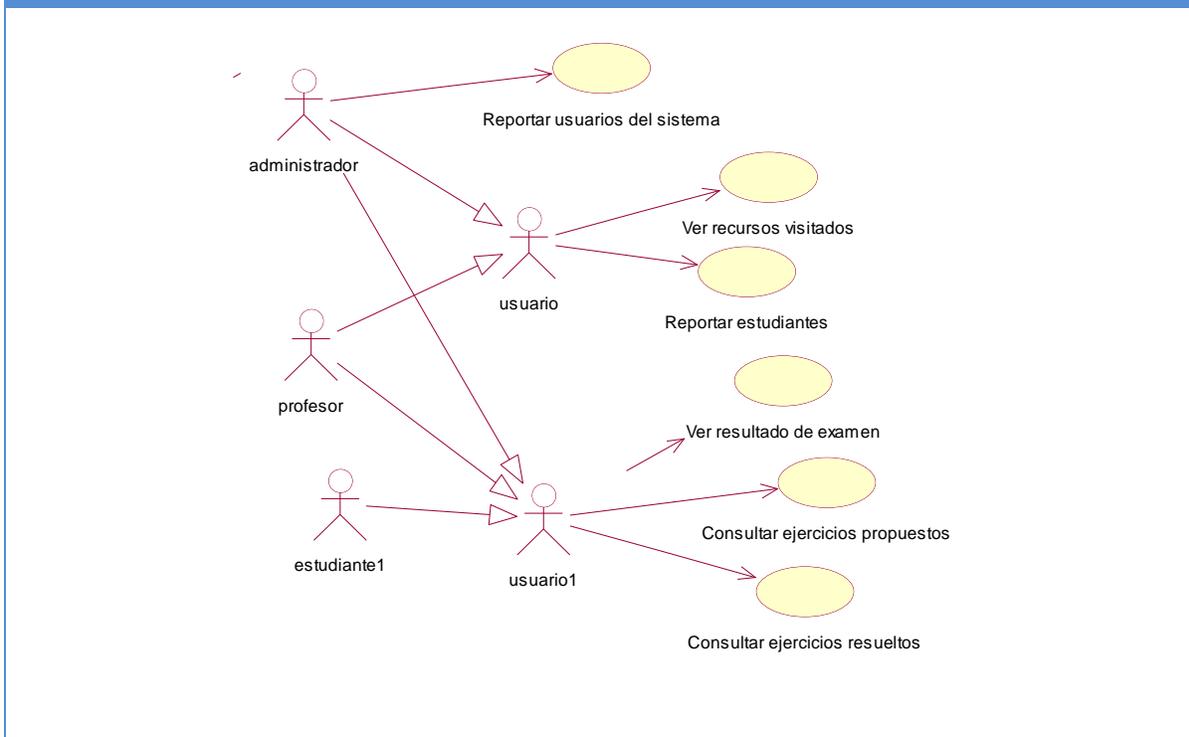


Figura 9. Diagrama de casos de uso. Paquete reportes.

2.3.4 Descripción de los casos de uso del sistema.

Tabla 5. Descripción del caso de uso consultar bibliografía.

<b>Caso de uso</b>	Modificar perfil.
<b>Actores</b>	Estudiante.
<b>Propósito:</b>	Modificar los datos referentes al estudiante.
<b>Resumen:</b> El caso de uso inicia cuando el estudiante desea modificar sus datos. Se muestra un formulario con todos sus datos, selecciona el que desea modificar y realiza los cambios.	
<b>Referencias</b>	RF-1.2
<b>Prototipo</b>	Anexo 1

Tabla 6. Descripción del caso de uso Crear cuenta.

<b>Caso de uso</b>	Crear cuenta.
<b>Actores</b>	Estudiante.
<b>Propósito:</b>	Crear un usuario en el sistema.
<b>Resumen:</b> El caso de uso inicia cuando el estudiante desea crear una nueva cuenta para entrar al sistema. Se muestra el formulario donde introduce sus datos. El caso de uso culmina con la actualización de los datos en el sistema.	
<b>Referencias</b>	RF-1.1
<b>Prototipo</b>	Anexo 2

Tabla 7. Descripción del caso de uso Autenticar usuario.

<b>Caso de uso</b>	Autenticar usuario.
--------------------	---------------------

<b>Actores</b>	Estudiante, Profesor, Administrador.
<b>Propósito:</b>	Entrar al sistema.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea entrar al sistema. Se muestra el formulario donde introduce su nombre y su contraseña. El caso de uso culmina con la entrada de los usuarios al sistema.	
<b>Referencias</b>	RF-2, RF-2.1, RF-2.2, RF-2.3
<b>Prototipo</b>	Anexo 3

**Tabla 8. Descripción del caso de uso Cambiar contraseña.**

<b>Caso de uso</b>	Cambiar contraseña.
<b>Actores</b>	Estudiante, Profesor, Administrador.
<b>Propósito:</b>	Cambiar la contraseña.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea cambiar su contraseña. Se muestra un formulario donde introduce su actual y nueva contraseña. El caso de uso culmina con la actualización de los datos en el sistema.	
<b>Referencias</b>	RF-1.4
<b>Prototipo</b>	Anexo 4

**Tabla 9. Descripción del caso de uso Gestionar usuario.**

<b>Caso de uso</b>	Gestionar usuario.
<b>Actores</b>	Administrador.
<b>Propósito:</b>	Adicionar, modificar o eliminar un usuario al sistema.

**Resumen:** El caso de uso inicia por tres vías diferentes:

- ✓ Cuando el administrador desea insertar un usuario al sistema. Se muestra el formulario donde este inserta el usuario y sus datos.
- ✓ Cuando el administrador desea modificar un usuario existente. Se muestran los usuarios registrados en el sistema, selecciona el que desea modificar y realiza los cambios.
- ✓ Cuando el administrador desea eliminar un usuario del sistema. Se muestran los usuarios registrados en el sistema, selecciona el que desea eliminar y realiza los cambios.

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

<b>Referencias</b>	RF-1, RF-1.1, RF-1.2, RF-1.3
<b>Prototipo</b>	Anexo 5

**Tabla 10. Descripción del caso de uso Gestionar ejercicios propuestos.**

<b>Caso de uso</b>	Gestionar ejercicios propuestos.
<b>Actores</b>	Profesor, Administrador.
<b>Propósito:</b>	Adicionar, modificar o eliminar un ejercicio propuesto en el sistema.

**Resumen:** El caso de uso inicia por tres vías diferentes:

- ✓ Cuando el usuario desea insertar un ejercicio propuesto al sistema. Se muestra el área de texto donde este inserta la orden del ejercicio.
- ✓ Cuando el usuario desea insertar un ejercicio propuesto existente. Se muestra un listado de los ejercicios propuestos, selecciona el que desea modificar y realiza los cambios.
- ✓ Cuando el usuario desea insertar un ejercicio propuesto del sistema. Se muestra un listado de los ejercicios propuestos, selecciona el que desea eliminar y realiza los cambios.

El caso de uso culmina con la actualización de los datos en el sistema.	
<b>Referencias</b>	RF-5, RF-5.1, RF-5.2, RF-5.3
<b>Prototipo</b>	Anexo 6

**Tabla 11. Descripción del caso de uso Gestionar ejercicios resueltos.**

<b>Caso de uso</b>	Gestionar ejercicios resueltos.
<b>Actores</b>	Profesor, Administrador.
<b>Propósito:</b>	Adicionar, modificar o eliminar un ejercicio resuelto en el sistema.
<p><b>Resumen:</b> El caso de uso inicia por tres vías diferentes:</p> <ul style="list-style-type: none"> <li>✓ Cuando el usuario desea insertar un ejercicio resuelto al sistema. Se muestran dos áreas de texto donde este inserta la orden del ejercicio y su respuesta.</li> <li>✓ Cuando el usuario desea insertar un ejercicio resuelto existente. Se muestra un listado de los ejercicios propuestos, selecciona el que desea modificar y realiza los cambios.</li> <li>✓ Cuando el usuario desea insertar un ejercicio resuelto del sistema. Se muestra un listado de los ejercicios propuestos, selecciona el que desea eliminar y realiza los cambios.</li> </ul> <p>El caso de uso culmina con la actualización de los datos en el sistema.</p>	
<b>Referencias</b>	RF-3, RF-3.1, RF-3.2, RF-3.3
<b>Prototipo</b>	Anexo 7

**Tabla 12. Descripción del caso de uso Realizar autoexamen.**

<b>Caso de uso</b>	Realizar autoexamen.
<b>Actores</b>	Estudiante

<b>Propósito:</b>	Realizar autoexamen.
<b>Resumen:</b> El caso de uso inicia cuando el estudiante desea Realizar el autoexamen para medir sus conocimientos. Se muestra e examen donde el estudiante va a responder cada pregunta. El caso de uso culmina cuando el estudiante recibe la calificación del examen.	
<b>Referencias</b>	RF-7, RF-10
<b>Prototipo</b>	Anexo 8

**Tabla 13. Descripción del caso de uso Reportar usuarios del sistema.**

<b>Caso de uso</b>	Reportar usuarios del sistema.
<b>Actores</b>	Administrador.
<b>Propósito:</b>	Ver los usuarios registrados en el sistema.
<b>Resumen:</b> El caso de uso inicia cuando el administrador desea ver los usuarios registrados en el sistema. El caso de uso culmina cuando se muestra el listado de usuarios.	
<b>Referencias</b>	RF-8
<b>Prototipo</b>	Anexo 9

**Tabla 14. Descripción del caso de uso Ver recursos visitados.**

<b>Caso de uso</b>	Ver recursos visitados.
<b>Actores</b>	Profesor, Administrador.
<b>Propósito:</b>	Ver los recursos que han sido visitados por los estudiantes.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea ver los recursos que han sido visitados por los estudiantes. El caso de uso culmina cuando se muestra el listado de las	

visitas.	
<b>Referencias</b>	RF-12
<b>Prototipo</b>	Anexo 10

**Tabla 15. Descripción del caso de uso Reportar estudiantes.**

<b>Caso de uso</b>	Reportar estudiantes.
<b>Actores</b>	Profesor, Administrador.
<b>Propósito:</b>	Ver los estudiantes registrados en el sistema.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea ver los estudiantes registrados en el sistema. El caso de uso culmina cuando se muestra el listado de estudiantes.	
<b>Referencias</b>	RF-9
<b>Prototipo</b>	Anexo 11

**Tabla 16. Descripción del caso de uso Ver resultados de examen.**

<b>Caso de uso</b>	Ver resultado de examen.
<b>Actores</b>	Profesor, Administrador.
<b>Propósito:</b>	Ver la calificación del examen de cada estudiante.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea ver la calificación del examen de cada estudiante. El caso de uso culmina cuando se muestra el listado de las calificaciones.	
<b>Referencias</b>	RF-10
<b>Prototipo</b>	Anexo 12

Tabla 17. Descripción del caso de uso Consultar ejercicios propuestos.

<b>Caso de uso</b>	Consultar ejercicios propuestos.
<b>Actores</b>	Estudiante, Profesor, Administrador.
<b>Propósito:</b>	Consultar ejercicios propuestos.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea Consultar un ejercicio propuesto. Se muestra un listado con todos los ejercicios propuestos. El caso de uso culmina cuando el usuario selecciona el ejercicio que desea consultar.	
<b>Referencias</b>	RF-6
<b>Prototipo</b>	Anexo 13

Tabla 18. Descripción del caso de uso Consultar ejercicios resueltos.

<b>Caso de uso</b>	Consultar ejercicios resueltos.
<b>Actores</b>	Estudiante, Profesor, Administrador.
<b>Propósito:</b>	Consultar ejercicios resueltos.
<b>Resumen:</b> El caso de uso inicia cuando el usuario desea Consultar un ejercicio resuelto. Se muestra un listado con todos los ejercicios resueltos. El caso de uso culmina cuando el usuario selecciona el ejercicio que desea consultar.	
<b>Referencias</b>	RF-4
<b>Prototipo</b>	Anexo 14

### **Conclusiones**

En este capítulo se realizó un análisis para conocer cómo funciona el negocio en el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado en los estudiantes de Ingeniería Informática de la Uniss, se estudiaron los procesos que lo conforman, lo que permitió modelarlo mediante reglas, casos de uso, actores y trabajadores que en él participan. Se definieron los principales requerimientos funcionales del sistema, así como los actores y su relación con estos, se definieron los paquetes de casos de uso que permiten agruparlos según su funcionalidad y por último se realiza una descripción de cada caso de uso.

### **Capítulo 3: Implementación de la aplicación propuesta para apoyar el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos en la asignatura Aprendizaje Automatizado:**

#### **Introducción**

El presente capítulo abarca todos los aspectos referentes al diseño de la solución propuesta. Se presentará el diagrama de componente y de despliegue, que resultaron del diseño realizado de cada uno de los casos de uso del sistema planteados. Se planteará también el modelo de datos a partir del diagrama de clases persistentes. Por último, se hará referencia a los estándares de diseño y de programación que se tienen en cuenta.

#### **3.1 Diagrama de Clases Web.**

El diagrama de clases es el principal diagrama de análisis y diseño, en él se representan las clases, con sus atributos, operaciones, interfaces y relaciones que existen entre ellas; se puede dividir en capas, aplicación y datos, las cuales muestran por separado las clases que intervienen en la lógica de la aplicación, la interfaz de usuario y el almacenamiento de datos. El diagrama de clases Web, representa a las páginas lógicas como clase, y de igual manera se representan sus relaciones, sobre el tema Jacobson, Booch & Rumbaugh expresan: “Un diagrama de clases Web representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML” (Jacobson, Booch, & Rumbaugh, 2006).

Teniendo en cuenta lo anterior y haciendo uso de las extensiones de UML para Web se elaboraron los siguientes diagramas de clases Web:

Diagramas de clases de diseño. Caso de uso Modificar perfil.

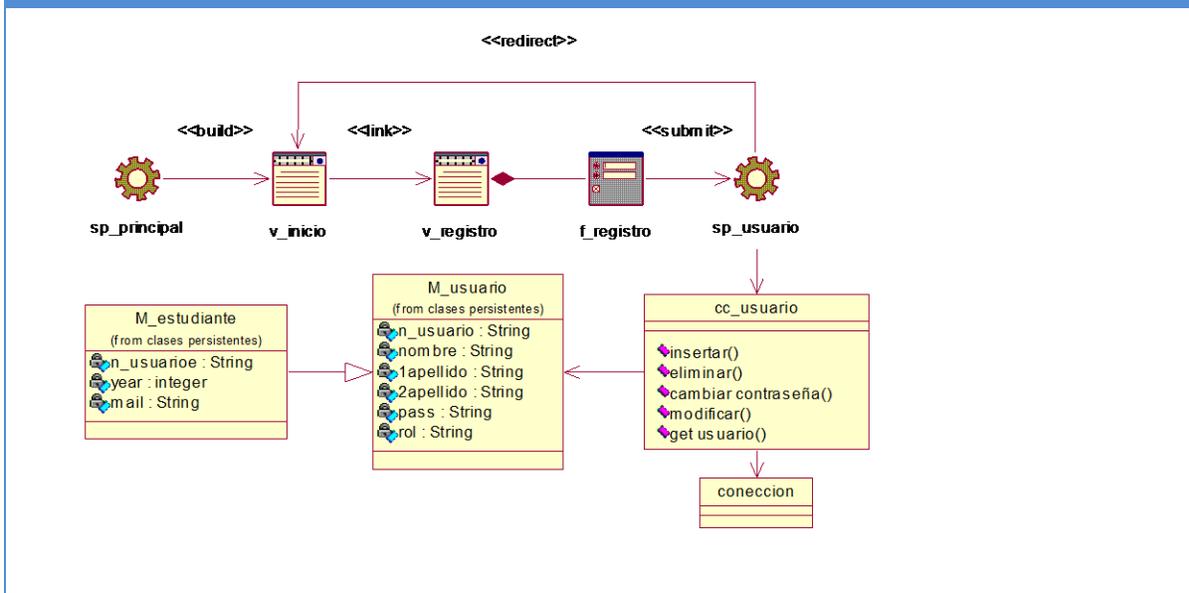


Figura 10. Diagramas de clases de diseño. Caso de uso Modificar perfil.

Diagramas de clases de diseño. Caso de uso Crear cuenta.

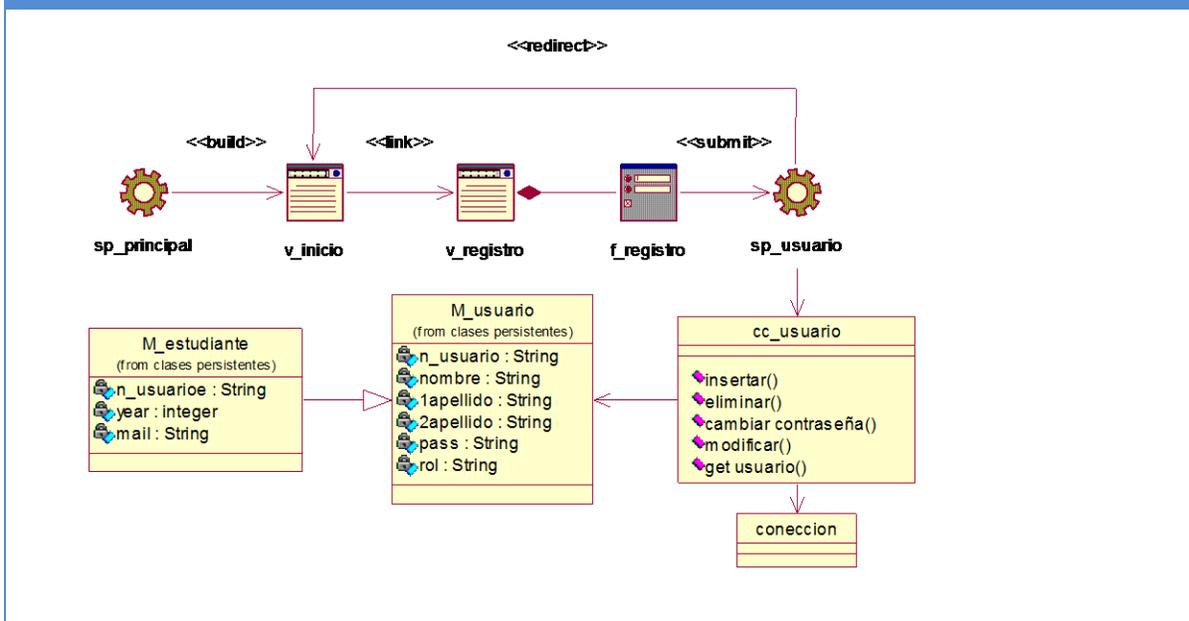


Figura 11. Diagramas de clases de diseño. Caso de uso Crear cuenta.

Diagramas de clases de diseño. Caso de uso Autenticar usuario.

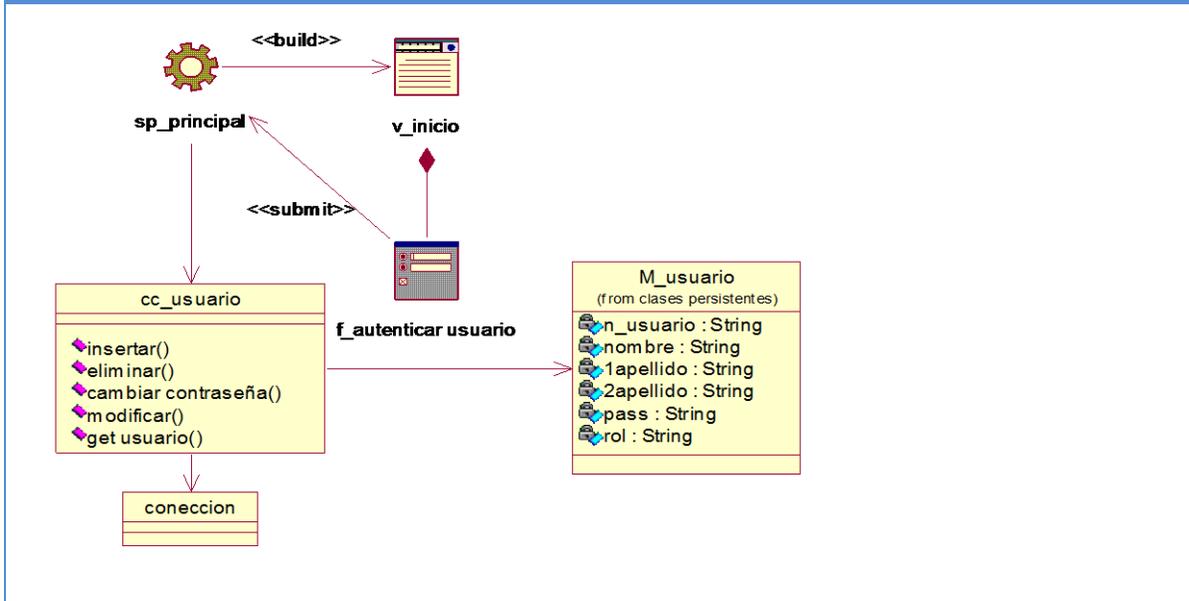


Figura 12. Diagramas de clases de diseño. Caso de uso Autenticar usuario.

Diagramas de clases de diseño. Caso de uso Cambiar contraseña.

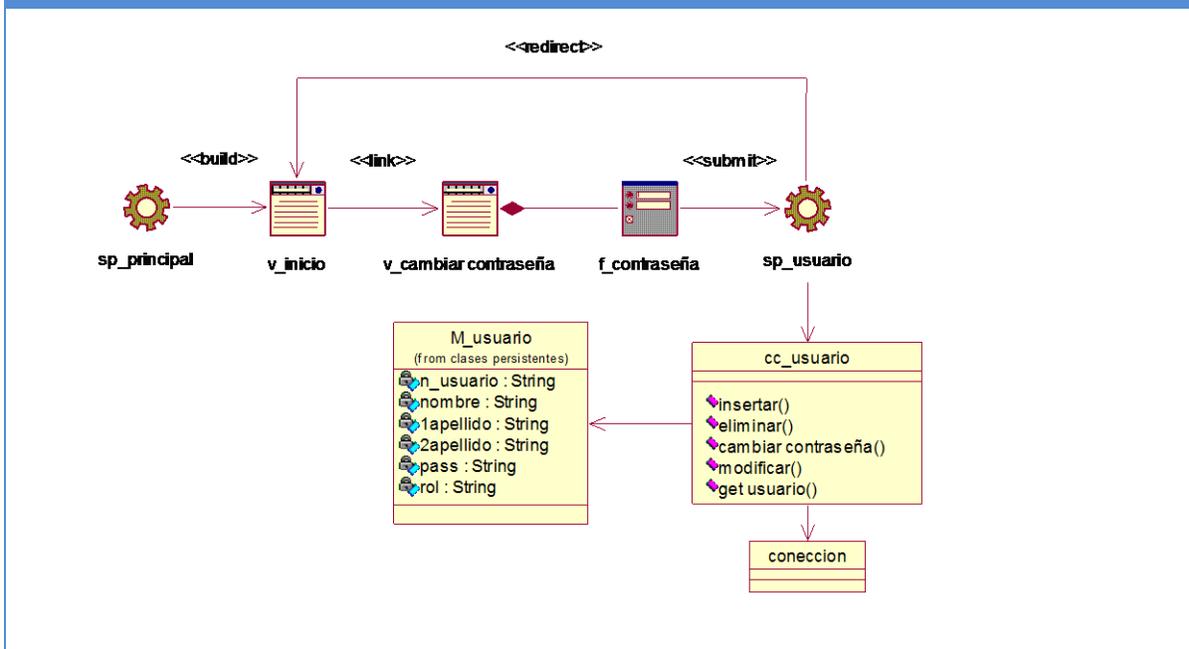


Figura 13. Diagramas de clases de diseño. Caso de uso Cambiar contraseña.

Diagramas de clases de diseño. Caso de uso Gestionar usuario.

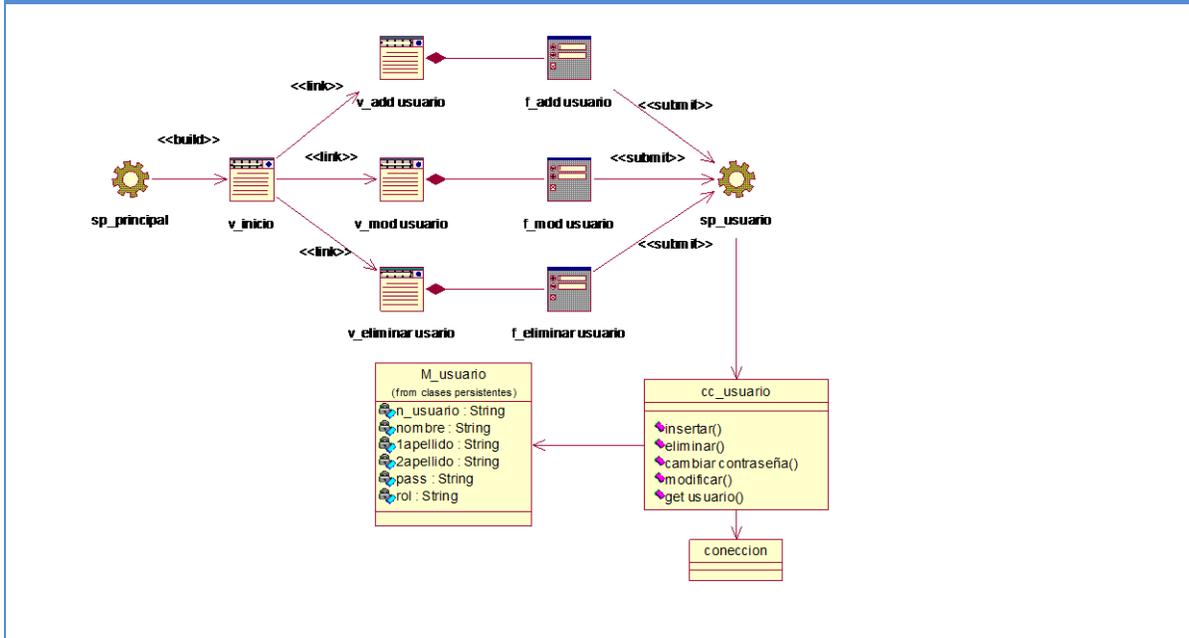


Figura 14. Diagramas de clases de diseño. Caso de uso Gestionar usuario.

Diagramas de clases de diseño. Caso de uso Gestionar ejercicios propuestos.

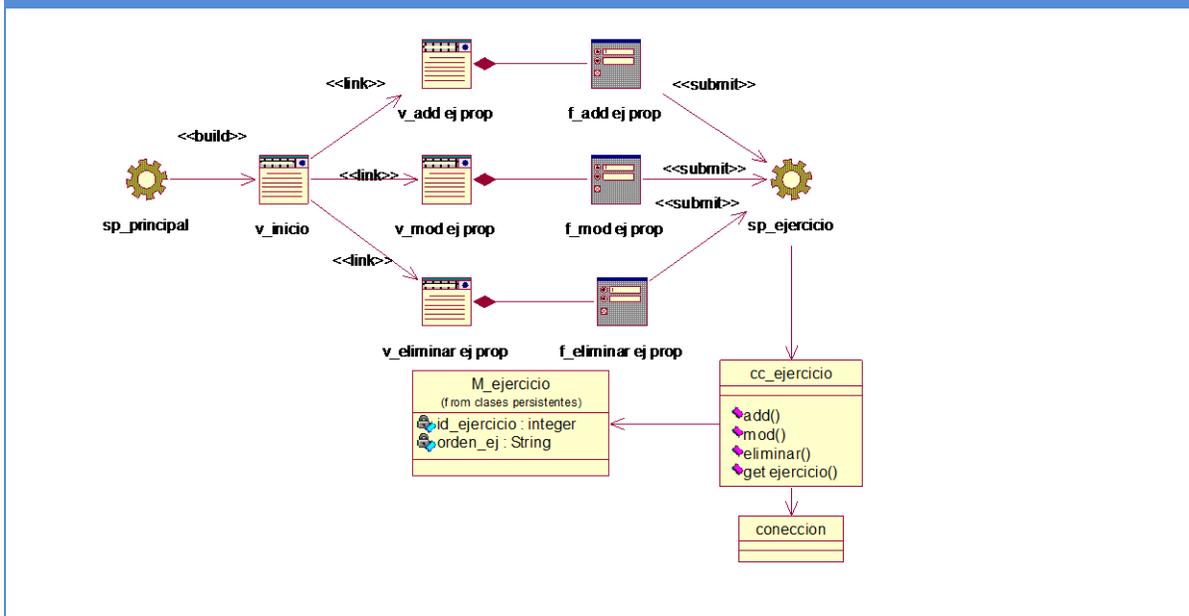


Figura 15. Diagramas de clases de diseño. Caso de uso Gestionar ejercicios propuestos.

Diagramas de clases de diseño. Caso de uso Gestionar ejercicios resueltos.

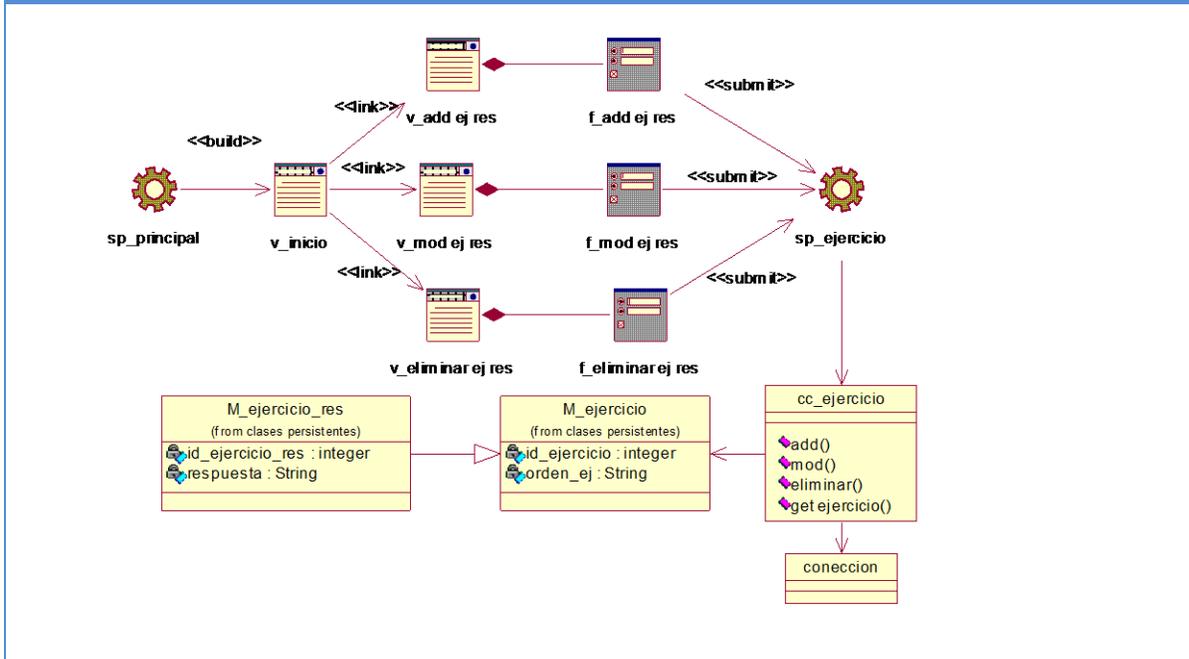


Figura 16. Diagramas de clases de diseño. Caso de uso Gestionar ejercicios resueltos.

Diagramas de clases de diseño. Caso de uso Realizar autoexamen.

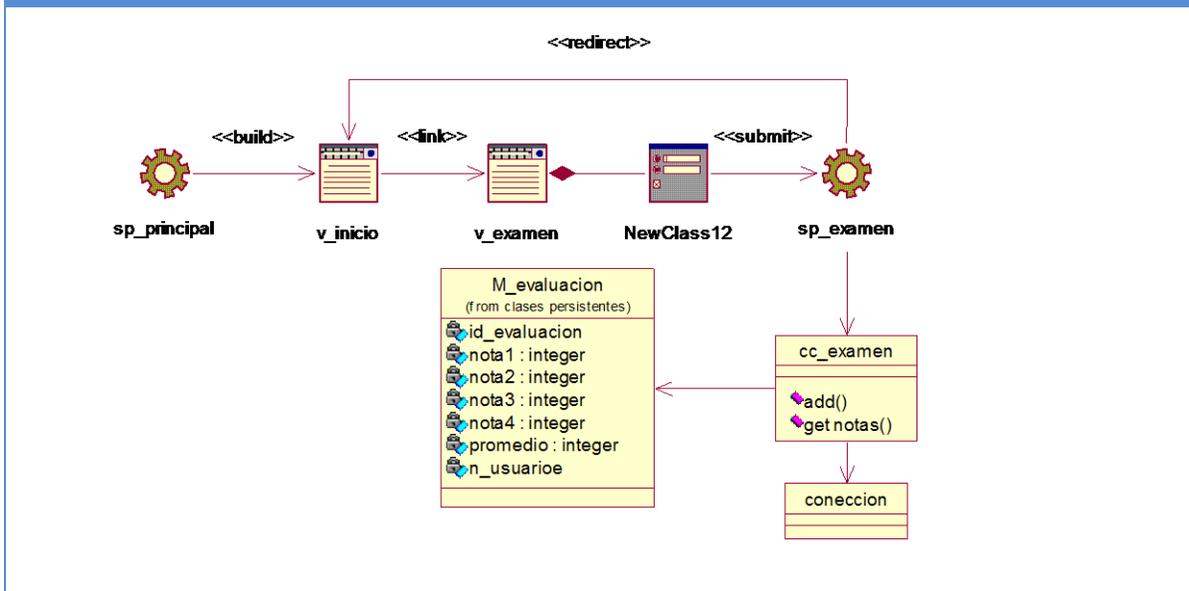


Figura 17. Diagramas de clases de diseño. Caso de uso Realizar autoexamen.

Diagramas de clases de diseño. Caso de uso Reportar usuarios del sistema.

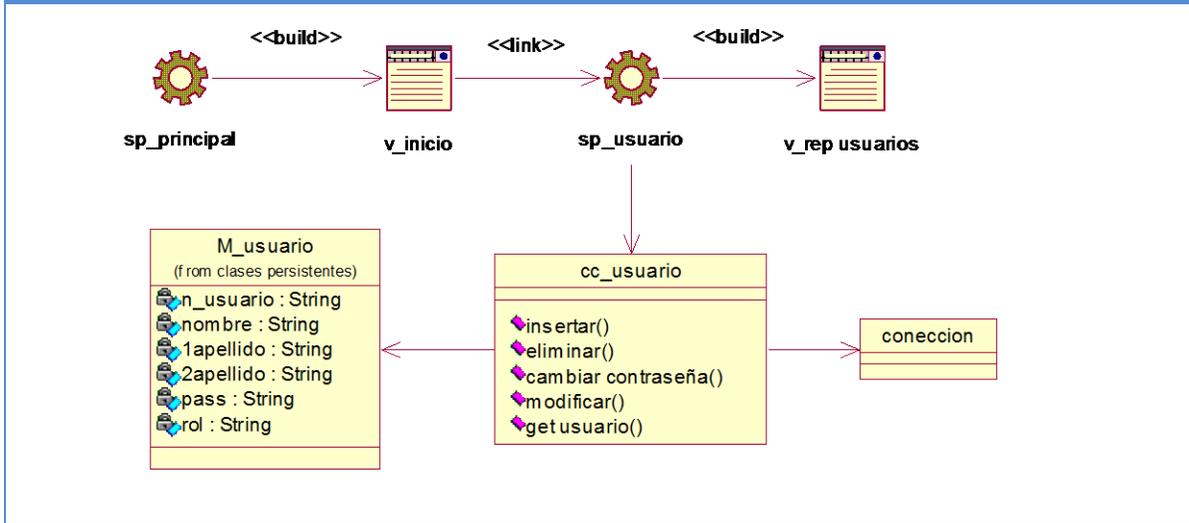


Figura 18. Diagramas de clases de diseño. Caso de uso Reportar usuarios del sistema.

Diagramas de clases de diseño. Caso de uso Ver recursos visitados.

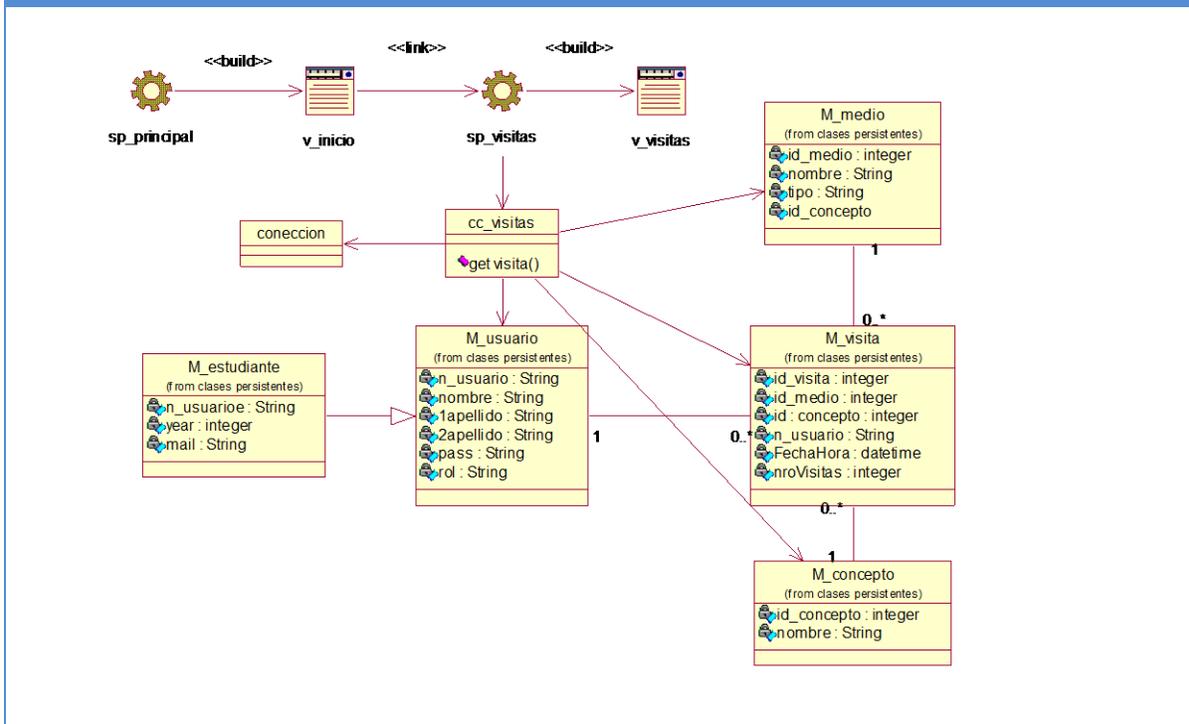


Figura 19. Diagramas de clases de diseño. Caso de uso Ver recursos visitados.

Diagramas de clases de diseño. Caso de uso Reportar estudiantes.

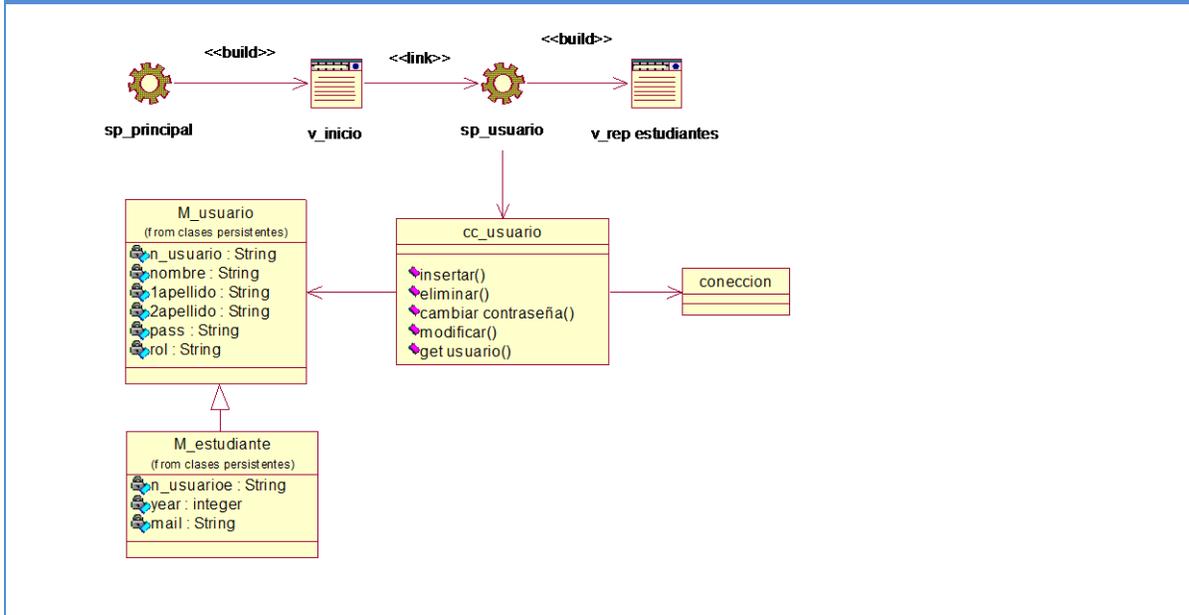


Figura 20. Diagramas de clases de diseño. Caso de uso Reportar estudiantes.

Diagramas de clases de diseño. Caso de uso Ver resultados de examen.

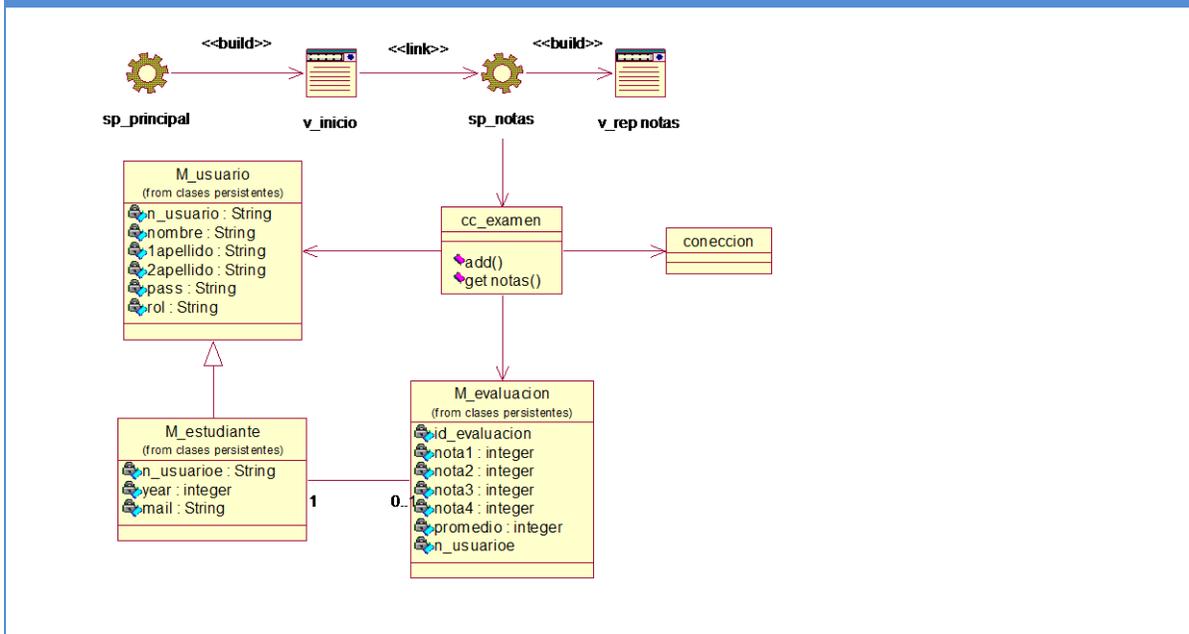


Figura 21. Diagramas de clases de diseño. Caso de uso Ver resultados de examen.

Diagramas de clases de diseño. Caso de uso Consultar ejercicios propuestos.

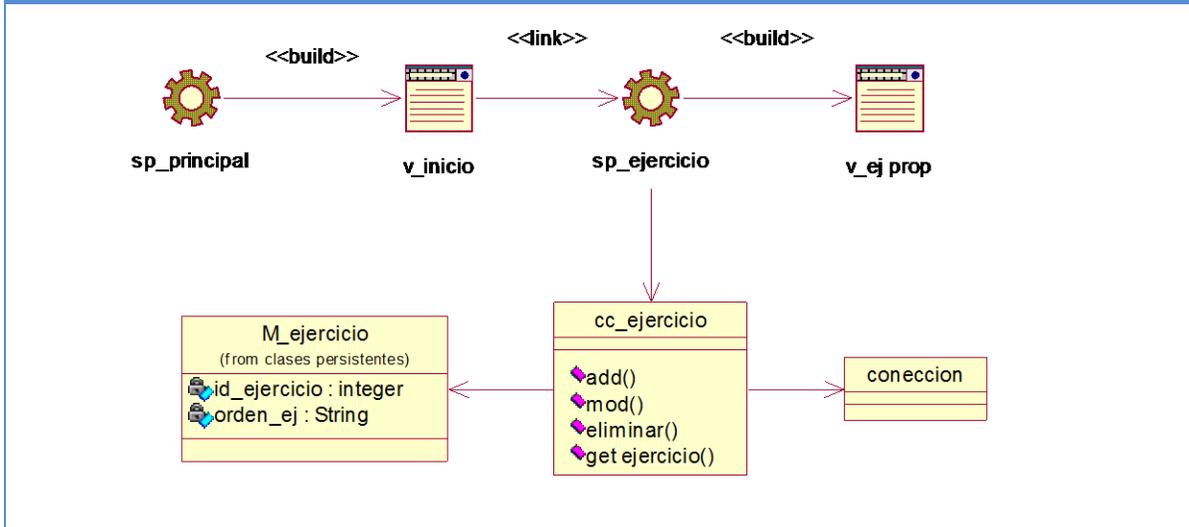


Figura 22. Diagramas de clases de diseño. Caso de uso Consultar ejercicios propuestos.

Diagramas de clases de diseño. Caso de uso Consultar ejercicios resueltos.

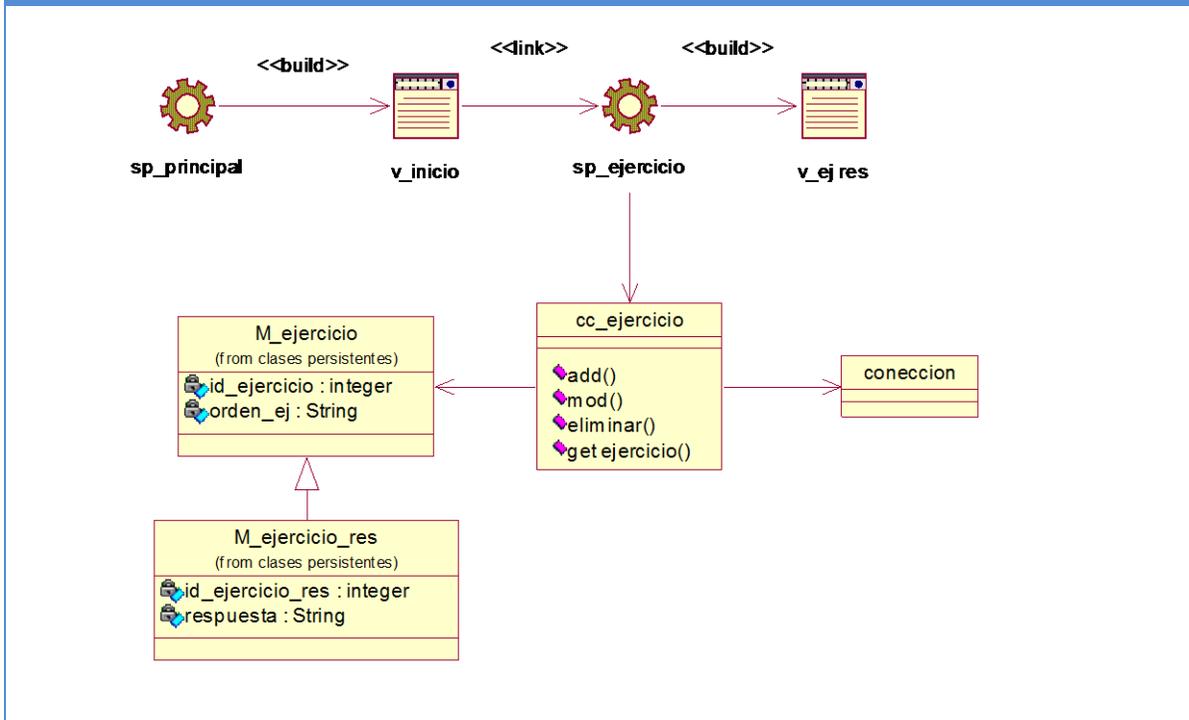


Figura 23. Diagramas de clases de diseño. Caso de uso Consultar ejercicios resueltos.

### 3.2 Diseño de la base de datos.

Cuando se va a realizar el diseño de la base de datos para un sistema dado, es necesario determinar los datos que se deben tomar en cuenta y las dependencias funcionales existentes entre ellos. Esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en ésta (García, 2006).

#### 3.2.1 Diagrama de clases persistentes.

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo (Jacobson, Booch, & Rumbaugh, 2006).

A partir de este planteamiento se definieron las clases que participan en el modelo lógico de datos.

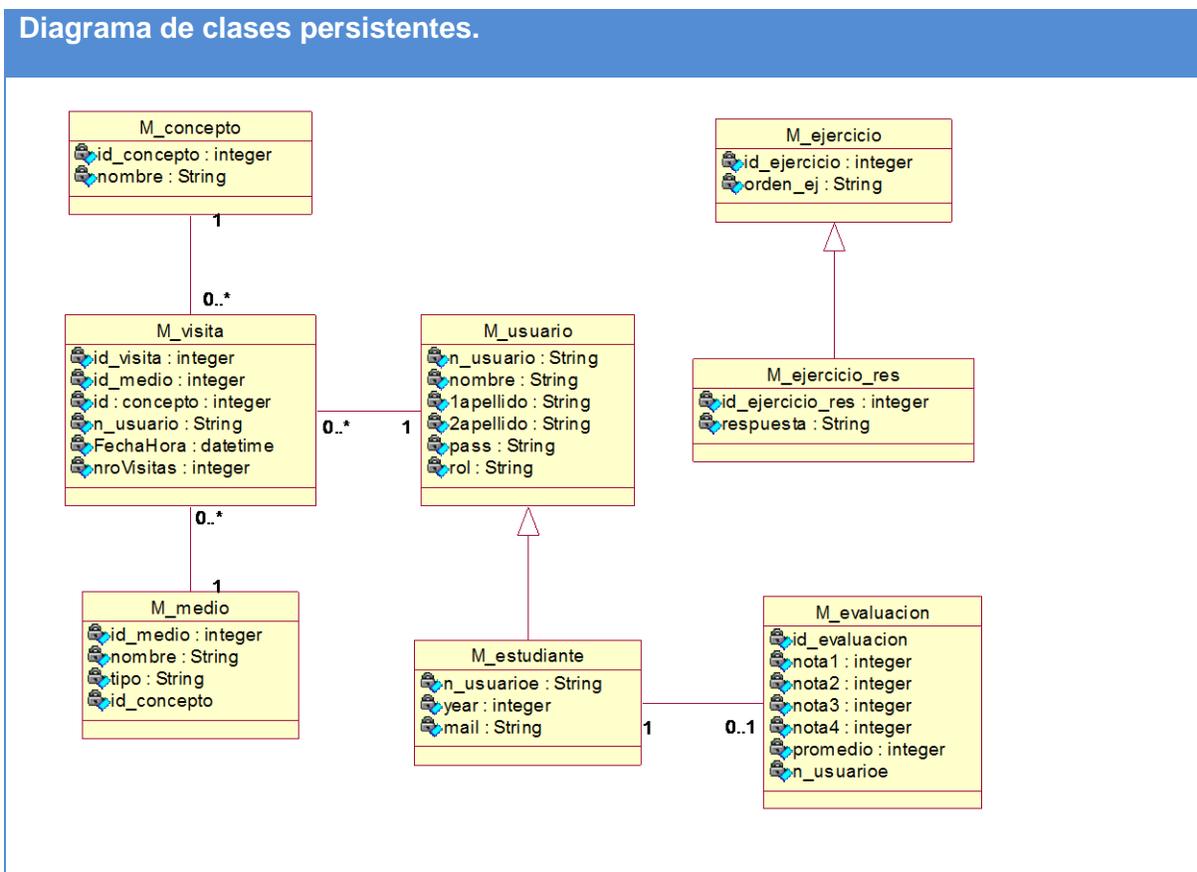


Figura 24. Diagramas de clases persistentes.

### 3.2.2 Diagrama del Modelo Físico de Datos.

El modelo físico de datos o modelo de datos, representa la estructura o descripción física de las tablas de la base de datos y se obtiene a partir del modelo lógico de datos (Jacobson, Booch, & Rumbaugh, 2006).

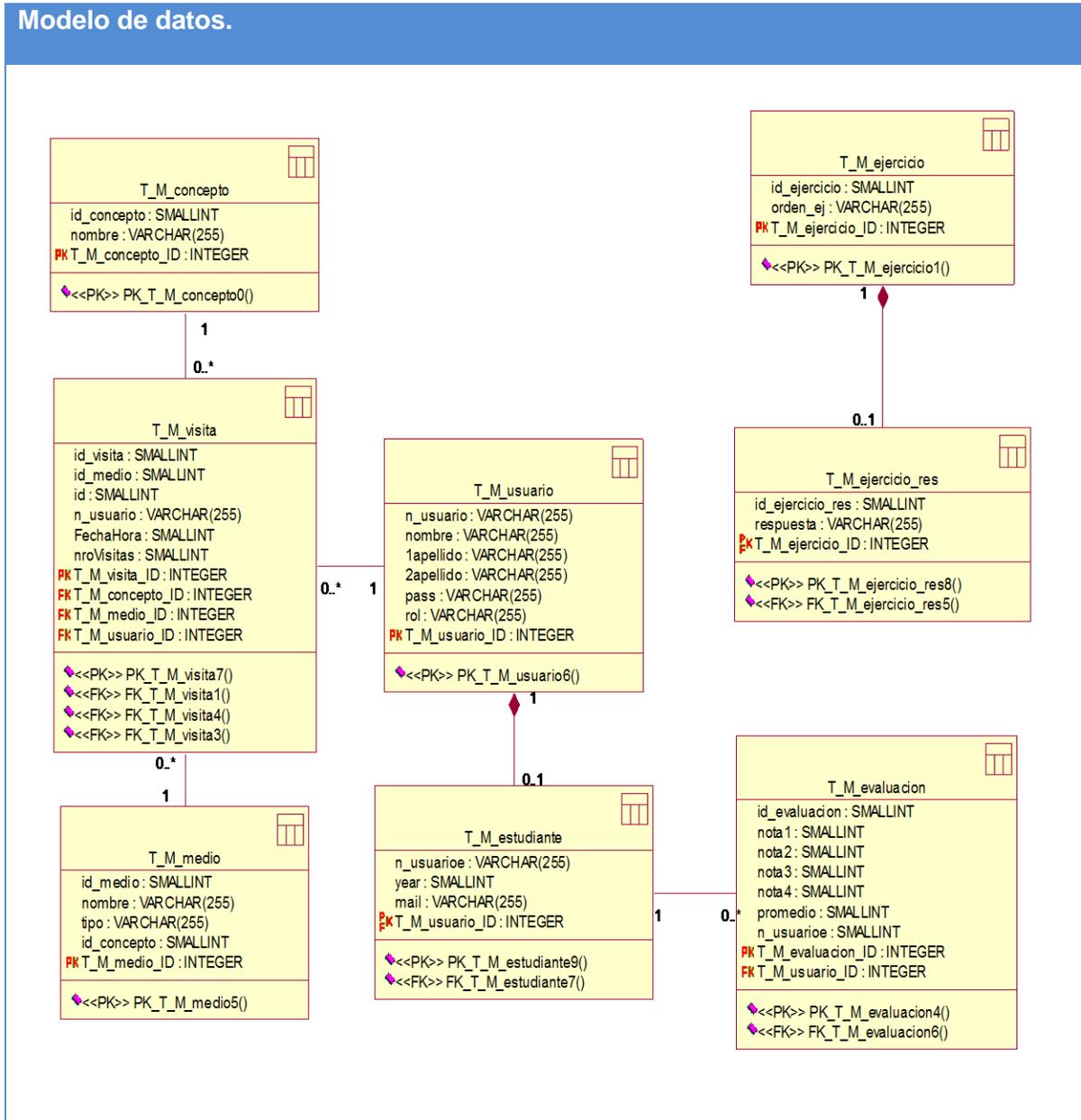


Figura 25. Modelo de datos.

### 3.3 Principios de diseño.

El diseño de la interfaz de un Sitio Web, el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones tiene gran influencia en el éxito o fracaso del mismo. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

#### 3.3.1 Estándares en la interfaz de la aplicación.

Cuando un usuario visita por primera vez un Sitio Web, la impresión que dicho usuario se lleva está dada por el atractivo de su diseño. Es por ello que, para lograr una apariencia adecuada y que el usuario se sienta cómodo, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, y navegación.

La interfaz diseñada para el sistema presenta las siguientes características:

- Se utilizarán las tonalidades de azul por ser suaves y refrescantes en concordancia con los colores del logo de la organización.
- El vocabulario no utiliza términos técnicos, acercándose al utilizado por los usuarios.
- Información legible.
- La navegación y el uso del sistema es fácil.
- La entrada de información por parte de los usuarios se realiza a través de formularios.
- Las interacciones con el sistema se basan fundamentalmente en selecciones de menú, en sus distintas formas y en acciones sobre elementos de código visual como botones, y mensajes.
- Los procedimientos que se ejecutan para acceder a la información almacenada en la base de datos son rápidos e incrementales obteniéndose resultados inmediatos.

### **3.3.2 Concepción general de la ayuda.**

Dentro del mundo de los Sitios Web en general, la ayuda constituye una parte importante del sistema. Las tendencias actuales apuntan a que estas no deben ser muy detallistas o extensas, sino simplemente explicaciones sencillas y aclaraciones del producto y de las operaciones que puede realizar el usuario sobre el mismo. Se concibió una ayuda amigable y práctica para el módulo del sistema relacionado con la entrada y actualización de los datos. La ayuda de la aplicación quedó formada por un menú interno que contiene las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultar la funcionalidad que realiza, descrita de un modo fácil de entender.

### **3.3.3 Tratamiento de errores.**

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que se registra y muestra en el sistema. Las posibilidades de introducir información errónea en el sistema por parte del usuario son mínimas, para lograr esto se realizaron dos tipos de validaciones la del cliente y la del servidor. Las validaciones del lado del cliente se realizaron en Javascript, evitando hacer peticiones innecesarias al servidor, en el caso de que los script no puedan ser validados en el cliente, entonces la validación se realiza en el servidor. En cualquiera de los dos casos los mensajes de error que emite el sistema se muestran en un lenguaje claro, preciso y de fácil comprensión para los usuarios.

### **3.3.4 Seguridad.**

Para lograr la seguridad y protección de la información almacenada, el sistema exige a los usuarios un nombre de usuario y una contraseña para poder acceder al módulo de entrada y actualización de los datos, controlándose de esta forma los niveles de acceso a la información. Solo podrán actualizar información presente en la base de datos del sistema, los usuarios que cuenten con dichos privilegios. La consistencia de los datos es otro aspecto que se toma en consideración, y para ello el sistema cuenta con formularios validados, con funciones del lenguaje PHP que garantizan que la información que se registre en la base de datos sea totalmente consistente e íntegra.

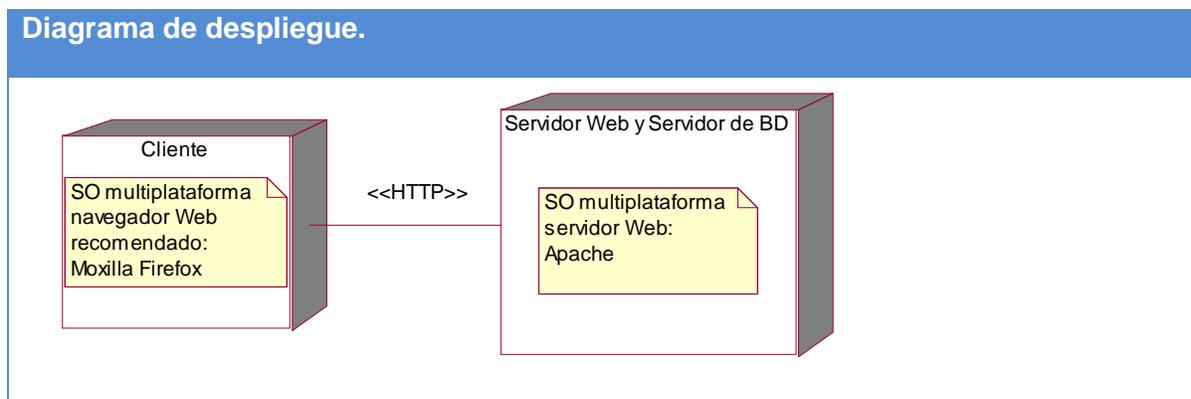
### 3.4 Modelo de implementación.

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración, y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (Cáceres, 2005).

#### 3.4.1 Diagrama de despliegue.

El diagrama de despliegue muestra la forma en que los componentes se desplegarán en el sistema, en este se detalla las especificaciones de la red, del servidor y los requisitos de hardware. Según Jacobson, Booch, & Rumbaugh un diagrama de despliegue es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos (Jacobson, Booch, & Rumbaugh, 2006).

A continuación se muestra el Diagrama de Despliegue de la aplicación:



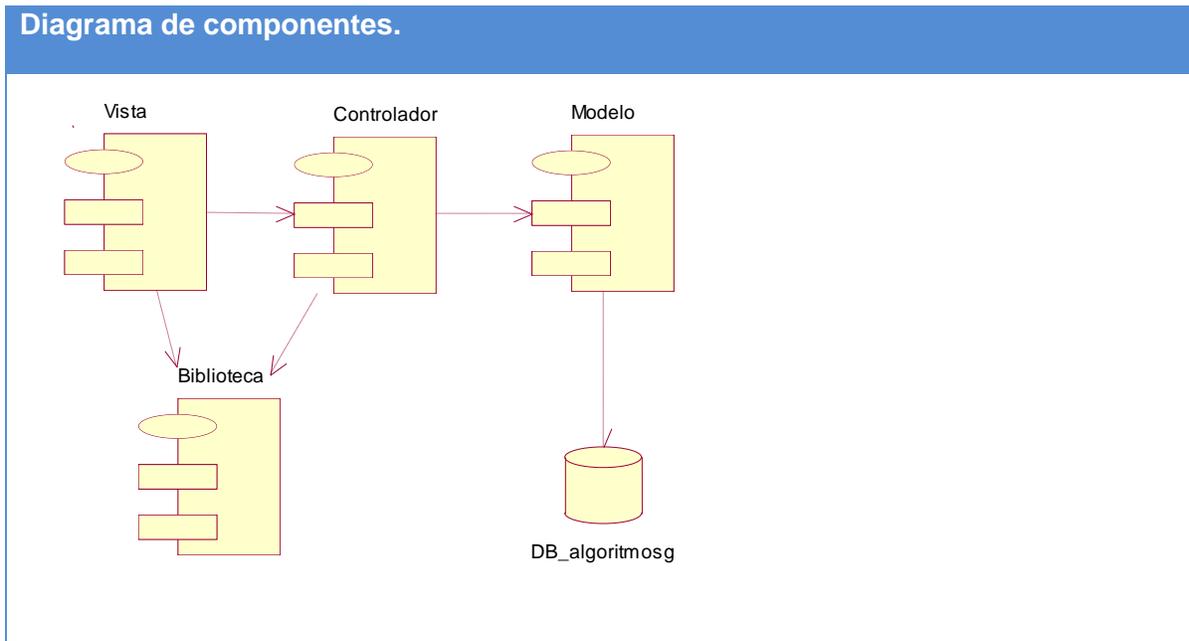
**Figura 26. Diagramas de despliegue.**

#### 3.4.2 Diagrama de componentes.

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que

un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones (Jacobson, Booch, & Rumbaugh, 2006).

La siguiente figura muestra el diagrama de componentes:



**Figura 27. Diagramas de componentes.**

### Conclusiones.

En el presente capítulo se mostraron los resultados de la etapa de implementación del sistema. Se desarrollaron los diagramas de clases, el diseño de la base de datos, el diagrama de despliegue y el de componentes. Además, se describieron los principios de diseño seguidos, específicamente, estándares de la interfaz de usuario, el formato de reportes, el tratamiento de excepciones, la concepción de la ayuda y los estándares de codificación.

### Conclusiones

- Los fundamentos teóricos y metodológicos determinados sustentaron la utilidad de elaborar un tutorial que apoyara el proceso de enseñanza - aprendizaje de los Algoritmos Genéticos utilizando los mapas conceptuales para representar el conocimiento y las tecnologías web para la implementación.
- Se diseñó una propuesta de solución, partiendo de la descripción de los procesos del negocio y de la modelación de los casos de uso utilizando para ello la metodología RUP y el lenguaje de modelado UML.
- Se implementó un tutorial para el apoyo de la enseñanza-aprendizaje de los algoritmos genéticos que empleó los mapas conceptuales como forma de representar el conocimiento, utilizando el lenguaje PHP para la lógica del negocio y los lenguajes Java Script y HTML para la interfaz del usuario. Se utilizó para ello la herramienta Dreamweaver 8.

### **Recomendaciones**

- Poner a prueba el sistema durante un período de tiempo significativo para comprobar su desempeño.
- Ampliar el marco de la aplicación y no dejarlo solamente en el proceso de enseñanza-aprendizaje de los algoritmos genéticos.

## Bibliografía

Álvarez, M. A. (2005). *Evaluando Zend Studio*. Obtenido de Evaluando Zend Studio:  
<http://www.maestrosdelweb.com/editorial/zendstudio/>

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

*Arquitectura Cliente/Servidor*. (2007). Obtenido de Arquitectura Cliente/Servidor:  
[www.desarrolloweb.com/.../arquitectura-cliente-servidor.html](http://www.desarrolloweb.com/.../arquitectura-cliente-servidor.html)

Borges, T. R. (2007). *LA EVOLUCIÓN DE LAS TELECOMUNICACIONES EN SANCTI SPÍRITUS*. SANCTI SPÍRITUS.

Cáceres, M. (2005). *Generador de Patrones de Navegación Aplicando Web Log Mining en Cliente/Servidor*. Obtenido de Generador de Patrones de Navegación Aplicando Web Log Mining en Cliente/Servidor: [www.tamps.cinvestav.mx/~vjsosa/thesis.html](http://www.tamps.cinvestav.mx/~vjsosa/thesis.html)

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

Cañas, A. J., & Ford, K. (2008). *Herramientas para construir y compartir Modelos de Conocimiento Basados en Mapas Conceptuales*. Obtenido de Herramientas para construir y compartir Modelos de Conocimiento Basados en Mapas Conceptuales:  
<http://www.ihmc.us/users/acanas/Publications/RevistaInformaticaEducativa/HerramientasConsonRIE.htm>

Carrasco, R. C. (2006). *Introducción al diseño y a la programación orientada a objetos*. Obtenido de Introducción al diseño y a la programación orientada a objetos:  
<http://www.nielsoft.com/Seminario/3capas/introduccion.ppt>

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

Castellanos, A. G. (2005). *Comparación entre sistemas de gestión de bases de datos (SGBD)*. Bogotá.

Castillo, F. S. (2008). *La investigación sobre medios de enseñanza: Pasado y Presente*.

Catoní, M. L. (25 de enero de 2009). *Software y Educación*. Obtenido de Software y Educación:  
<http://blogs.rimed.cu/infoedu/category/software-educativo/>

*Conociendo algo de software educativo*. (2009). Obtenido de Conociendo algo de software educativo: <http://www.enlaces.udec.cl>

- Dávila, & Martínez, G. (2000). *Mapas Conceptuales, en busca del aprendizaje significativo*.
- Enríquez. (2007). *SQL Server 2000*. Obtenido de SQL Server 2000:  
<http://exa.unne.edu.ar/depar/areas/informática/.../SQL.pdf>
- Espinoza. (2005). Obtenido de [www.dataprix.com/251-funciones-agregacion](http://www.dataprix.com/251-funciones-agregacion)
- Eugenia, M. (26 de junio de 2005). *Las TICS en la Educación*. Recuperado el 2013, de Las TICS en la Educación: <http://educatics.blogspot.com/>
- García, R. M. (2006). *Sistema de Base de Datos*. La Habana: Félix Varela.
- Graells, P. M. (7 de agosto de 2011). *IMPACTO DE LAS TIC EN EDUCACIÓN: FUNCIONES Y LIMITACIONES*. Recuperado el 30 de marzo de 2013, de IMPACTO DE LAS TIC EN EDUCACIÓN: FUNCIONES Y LIMITACIONES: <http://www.peremarques.net>
- Guervós, J. J. (2011). *Informática evolutiva: Algoritmos genéticos*. Obtenido de Informática evolutiva: Algoritmos genéticos: <http://geneura.ugr.es/~jmerelo>
- Hernández. (1989). Recuperado el 05 de febrero de 2013, de <http://www.infor.uva.es/~descuder/docencia/pd/Node24.html>
- Historia de la Informática en América Latina y el Caribe: Investigaciones y testimonios*. (2010). Obtenido de Historia de la Informática en América Latina y el Caribe: Investigaciones y testimonios: <http://dc.exa.unrc.edu.ar/historia/node/31>
- Jacobson, I., Booch, G., & Rumbaugh, J. (2006). *El Proceso Unificado de Desarrollo de Software*. Felix varela.
- Lago, R. (abril de 2007). *Patrones de diseño software*. Obtenido de Patrones de diseño software: <http://www.proactiva-calidad.com/java/patrones/mvc.html>
- Laguna, M. A. (2004). *Ingeniería del Software II*. Recuperado el 12 de abril de 2013, de Ingeniería del Software II: <http://www.infor.uva.es/mlaguna/is2/1-Introd.pdf>
- Loaiza, R. (2009). *De la información a la informática*. Recuperado el 30 de enero de 2013, de De la información a la informática: [http://www.bvs.sld.cu/revistas/san/vol2\\_2\\_98/san14298.htm](http://www.bvs.sld.cu/revistas/san/vol2_2_98/san14298.htm)
- Mariño, S. I. (2009). *Diseño de un entorno virtual de enseñanza-aprendizaje para la asignatura Inteligencia Artificial*. Obtenido de Diseño de un entorno virtual de enseñanza-aprendizaje para la asignatura Inteligencia Artificial:  
[http://www.quadernsdigitals.net/datos\\_web/hemeroteca/r\\_1/nr\\_796/a\\_10731/10731.pdf](http://www.quadernsdigitals.net/datos_web/hemeroteca/r_1/nr_796/a_10731/10731.pdf)
- Marquez, P. (2010). *El software educativo*. Barcelona: Universidad Autónoma de Barcelona.

- Martínez, I. G. (2009). *Medios de transmisión de las información*.
- Munz, S. (31 de enero de 2006). *Tecnologías Web/ PHP*. Obtenido de Tecnologías Web/ PHP: <http://es.selfhtml.org/introduccion/tecnologias/php.htm>
- Novak, J. D., & Roca, D. L. (2002). *Aprendiendo a aprender*.
- Peláez, J. C. (25 de mayo de 2009). *Blog de Juan Peláez en Geeks.ms*. Obtenido de Blog de Juan Peláez en Geeks.ms: <http://www.juanpelaez.com/>
- Peña, J. A., & Truyol, A. P. (2008). *ALGORITMOS GENÉTICOS*. Universidad Carlos III .
- Pressman, R. S. (2007). *Ingeniería del Software Un enfoque práctico*. La Habana: Félix Varela.
- PROGRAMA ANALÍTICO DE APRENDIZAJE AUTOMATIZADO*. (2012).
- Rodríguez, R. (1991). *Búsqueda y estudio de los sistemas tutoriales para la importación de la computación*.
- Romero, S. B. (2004). *El Mapa Conceptual como estrategia de enseñanza y aprendizaje en la resolución de problemas*. La Habana.
- Rosario, J. (2005). *La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual*. Recuperado el 29 de enero de 2013, de La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual: <http://www.cibersociedad.net/archivo/articulo.php?art=218>
- Ruiz, M. H. (2006). *Programación WEB avanzada Soluciones rápidas y efectivas para desarrolladores de sitios*. La Habana: Felix Varela.
- Sitio oficial de Server2Go*. (2007). Obtenido de Sitio oficial de Server2Go: <http://www.server2go-web.de/>
- Suares, L. G., & Delgado, L. L. (2007). *Medios para aprender a aprender*.
- Valdés, D. P. (julio de 2007). *Los Frameworks de PHP agilizan tu trabajo*. Recuperado el marzo de 2013, de Los Frameworks de PHP agilizan tu trabajo: <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-traba>

Anexos

Anexo 1 Prototipo CU Modificar perfil.

Anexo 2 Prototipo CU Crear cuenta.

## Iniciar sesión

Nombre de usuario:

Contraseña:

Aceptar

[Registrarse](#)

Anexo 3 Prototipo CU Autenticar usuario.

**Ing** Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

Es necesario aclarar que todos los campos son obligatorios.

Usuario:

Contraseña:

Nueva contraseña:

Confirmar Contraseña:

Aceptar

Bienvenido dmesa

- Ver ejercicios resueltos
- Ver ejercicios propuestos
- Ver mapas
- Realizar examen
- Cambiar contraseña
- Modificar mi perfil
- Cerrar sesión

Anexo 4 Prototipo CU Cambiar contraseña.

Es necesario aclarar que todos los campos son obligatorios.

Nombre

1 Apellido

2 Apellido

Usuario

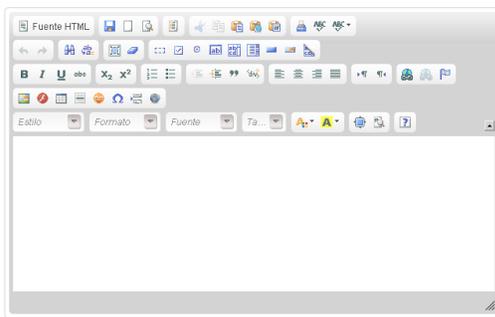
Contraseña

Repetir Contraseña

Rol

		Nombre de usuario	Nombre	Rol
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	dmesa	dunelvis mesa perez	estudiante
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	diane	dianelis mesa perez	estudiante
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	administrador		administrador
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	rrc	rangel rodriguez cabrera	profesor

Anexo 5 Prototipo CU Gestionar usuario.



[Ejercicio 1](#)

[Ejercicio 2](#)

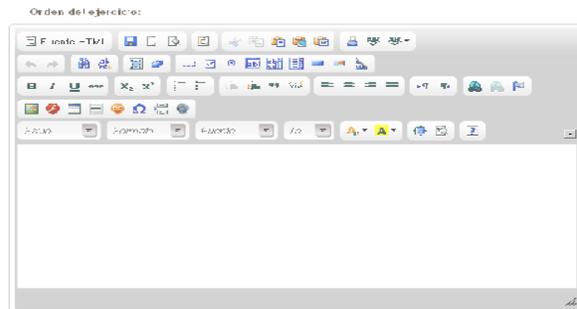
[Ejercicio 3](#)

[Ejercicio 4](#)

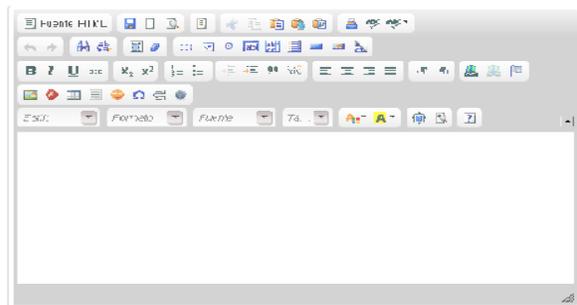
[Ejercicio 5](#)



Anexo 6 Prototipo CU Gestionar ejercicios propuestos.



Respuesta:



[Ejercicio 1](#)

[Ejercicio 2](#)

[Ejercicio 3](#)

[Ejercicio 4](#)

[Ejercicio 5](#)



Anexo 7 Prototipo CU Gestionar ejercicios resueltos.



**Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos**

0000111110000  
AABBAAAABBBBAAAAA  
0101010000111110  
ACCCBBBDD

🏠 ?

**Bienvenido dmesa**

- Ver ejercicios resueltos
- Ver ejercicios propuestos
- Ver mapas
- Realizar examen
- Cambiar contraseña
- Modificar mi perfil
- Cerrar sesión

**1-Responda verdadero o falso según corresponda.**

a) Un alelo es el valor asociado a un gen.  
Verdadero  Falso

b) La única vía para generar la población inicial es el sembrado de individuos.  
Verdadero  Falso

c) El genotipo es la posición de un gen en el cromosoma.  
Verdadero  Falso

Anexo 8 Prototipo CU Realizar autoexamen.



**Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos**

0000111110000  
AABBAAAABBBBAAAAA  
0101010000111110  
ACCCBBBDD

🏠 ?

**Bienvenido administrador**

- Añadir ejercicios resueltos
- Ver ejercicios resueltos
- Añadir ejercicios propuestos
- Ver ejercicios propuestos
- Ver mapas

**Listado de usuarios registrados**

		Nombre de usuario	Nombre	Rot
		dmesa	dunelvis mesa perez	estudiante
		diane	dianelis mesa perez	estudiante
		administrador		administrador
		rrc	rangel rodríguez cabrera	profesor

Anexo 9 Prototipo CU Reportar usuarios del sistema.

# Ing Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

Medios visitados por los estudiantes

Usuario	Concepto	Medio	Visitas	Última Visita
dmesa	Generacion aleatoria de la Poblacion Inicial	Generacion aleatoria de la poblacion inicial.pdf	1	2013-05-10 07:51:35
dmesa	Evaluacion de las cadenas de acuerdo a la Funcion	Evaluacion de las cadenas de acuerdo a la Funcion.pdf	1	2013-05-10 07:53:28
dmesa	Seleccion	Seleccion.pdf	1	2013-05-10 07:41:37
dmesa	Mutacion	Mutacion.pdf	1	2013-05-10 07:48:00
administrador	Locus	Locus.pdf	2	2013-05-27 15:24:51
administrador	Analisis y planteamiento del problema	Analisis y planteamiento del problema.pdf	1	2013-05-10 02:58:30

Bienvenido administrador

- Añadir ejercicios resueltos
- Ver ejercicios resueltos
- Añadir ejercicios propuestos
- Ver ejercicios propuestos
- Ver mapas
- Cambiar contraseña
- Insertar usuarios
- Reporte de usuarios
- Reporte de estudiantes

Anexo 10 Prototipo CU Ver recursos visitados.

# Ing Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

Listado de estudiantes registrados

Nombre de usuario	Nombre	Año	Correo
dmesa	dunelvis mesa perez	1	dmesa@suss.co.cu
diane	dianelis mesa perez	1	dmesa@fch.suss.co.cu

Bienvenido administrador

- Añadir ejercicios resueltos
- Ver ejercicios resueltos
- Añadir ejercicios propuestos

Anexo 11 Prototipo CU Reportar estudiantes.

**Ing** Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

10000111110000  
AABBAAAABBBBAAAAA  
0101010000111110  
ACCCBBBDD

🏠 🧠

**Listado de notas de los estudiantes**

Nombre	Pregunta1	Pregunta2	Pregunta2	Pregunta4	Nota final
dunelvis mesa perez	3	4	3	2	3
dianelis mesa perez	0	0	0	0	0

**Bienvenido administrador**

- Añadir ejercicios resueltos
- Ver ejercicios resueltos
- Añadir ejercicios propuestos
- Ver ejercicios propuestos

Anexo 12 Prototipo CU Ver resultados de examen.

**Ing** Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

10000111110000  
AABBAAAABBBBAAAAA  
0101010000111110  
ACCCBBBDD

🏠 🧠

[Ejercicio 1](#)  
[Ejercicio 2](#)  
[Ejercicio 3](#)  
[Ejercicio 4](#)  
[Ejercicio 5](#)

Antes de ponerlo en funcionamiento en el sistema, un programa Linux se debe transformar en comandos de lenguaje de máquina de bajo nivel conocidos como código binario. Por ejemplo, cuando se está programando se crea un conjunto de instrucciones utilizando palabras ordinarias que después deben convertirse en códigos binarios que se ajustan como una aplicación que se puede ejecutar. Un intérprete realiza en tiempo de ejecución esta conversión o se puede generar explícitamente de una vez con un cotnpilador. Los lenguajes de programación GAWK, Per1,Tcl y Tk de los shells BASH y TCSH usan sus propios intérpretes. Los programas desarrollados en estos lenguajes se pueden poner a funcionar directamente aunque sean archivos de la aplicación. Un compilador creará un archivo de aplicación por separado con el código binario que entonces se puede ejecutar esto.

**Bienvenido dmesa**

- Ver ejercicios resueltos
- Ver ejercicios propuestos
- Ver mapas
- Realizar examen
- Cambiar contraseña
- Modificar mi perfil
- Cerrar sesión

Anexo 13 Prototipo CU Consultar ejercicios propuestos.

# Ing Tutorial para el apoyo del proceso de enseñanza-aprendizaje de los Algoritmos Genéticos

[Ejercicio 1](#)

[Ejercicio 2](#)

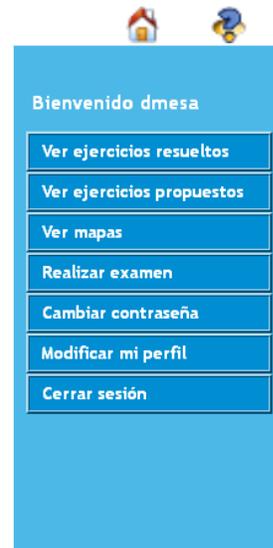
[Ejercicio 3](#)

[Ejercicio 4](#)

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración, y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros

## Respuesta

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



Bienvenido dmesa

- Ver ejercicios resueltos
- Ver ejercicios propuestos
- Ver mapas
- Realizar examen
- Cambiar contraseña
- Modificar mi perfil
- Cerrar sesión

Anexo 14 Prototipo CU Consultar ejercicios resueltos.