



Universidad de las Ciencias Informáticas

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Título: *Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial.*

Autores:

Adriana Gandol Magaña
José Dalién Roque Pérez

Tutores:

Msc. Darian Horacio Grass Boada
Ing. Dany Cuellar Rodríguez

“Año 55 de la Revolución”
La Habana, Cuba, Enero, 2013

Agradecimientos

De Adriana

A mi novio, por su paciencia, cariño, dedicación, amor, por haber llegado en el momento justo para ser de gran apoyo y sustento en mi vida. Te quiero mucho.

A mi suegra, por ser como una madre, por el gran amor que ha sabido brindarme desde que la conocí.

A Darian y Dani, nuestros tutores, por la confianza depositada, asesoramiento y ayuda, especialmente a Darian por haberse portado tan bien conmigo durante esta etapa tan difícil.

A mi compañero de tesis Dalién, por todos los momentos y apuros que le he hecho pasar, por su comprensión y su apoyo incondicional.

A mis compañeras de cuarto, especialmente a mis amigas Yudelís, Yanelis, Lianet, Rosali y Yenni por haber soportado mi mal carácter y pesadez, por ayudarme tanto, por sus consejos y apoyo, por ser mi familia desde que las conocí, las quiero muchísimo.

A mi amiga Arianna, por ser una hermana más, por sus consejos, por toda la ayuda que me ha brindado en la tesis y en todo momento. Espero siempre tenerla a mi lado.

A todos los que de una forma u otra siempre estuvieron conmigo en los buenos y malos momentos, Ramiro, Leonel, Carlos, Jenny, Wendy, Rogelio y Diannet muchas gracias.

A Darvis, Maidelís y Yunia, por su apoyo y comprensión.

De Dalién

A mis abuelos y mis padres, por confiar en mí y por apoyarme siempre.

A mi novia, por su comprensión y apoyo en estos cinco años.

A Darian y Dani, nuestros tutores, por la confianza depositada, asesoramiento y ayuda.

A mi compañera de tesis Adriana, por su apoyo.

A mis amigos Michel, Denis, Luis, Alejandro, Javier, Yunier, Pedro, Tan, Dani y Jova, por compartir estos años conmigo.

A mis compañeros de tesis del laboratorio 302, por compartir sus criterios.

Agradecimientos

A todos los que me dieron su apoyo, muchas gracias.

Dedicatoria

De Adriana

A mi mamá, mi papá, mis hermanos y mi bebita que viene en camino.

De Dalíen

A mi familia y mis amigos.

Declaración de autoría

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a que haga uso del mismo.

Para que así conste, firmamos la presente a los ____ días del mes de _____ del año _____.

Adriana Gandol Magaña

Firma del autor

José Dalién Roque Pérez

Firma del autor

Msc. Darían Horacio Grass Boada

Firma del tutor

Ing. Dany Cuellar Rodríguez

Firma del co-tutor

Resumen

En el contexto de los profundos cambios realizados en la Educación Superior encaminados a satisfacer las necesidades de una enseñanza productiva e individualizada al estudiante, que le faciliten una autogestión de su aprendizaje a través de la utilización de las Tecnologías de la Información y las Comunicaciones, surge el Trabajo de Diploma: “Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial”.

El propósito de su desarrollo consiste en crear una aplicación web que apoye el Proceso de Enseñanza-Aprendizaje de la asignatura Inteligencia Artificial I en la Universidad de las Ciencias Informáticas. Para ello se definieron Objetos de Aprendizaje capaces de transmitir conocimientos y habilidades a los estudiantes, en el contexto de los Métodos de Solución de Problemas, tema que presenta mayores dificultades por su abstracción y poca ejemplificación. Los Objetos de Aprendizaje creados permitirán la simulación de una serie de problemas computacionales clásicos que puedan ser solucionados mediante los Métodos de Solución de Problemas impartidos en la asignatura, de manera que se muestre una solución más atractiva. La aplicación ofrecerá al estudiante de forma interactiva, modificar parámetros y experimentar diferentes soluciones de un mismo problema, con el objetivo de mejorar la comprensión del mismo y su funcionamiento.

Índice

INTRODUCCIÓN	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	16
1.1. EL E-LEARNING COMO MODALIDAD EDUCATIVA	16
1.1.1. ¿Qué es el e-learning?	16
1.2. HERRAMIENTAS PARA EL ACCESO AL E-LEARNING	16
1.2.1. <i>Sistemas de Gestión de Contenido</i>	17
1.2.2. <i>Sistemas de Gestión de Aprendizaje</i>	17
1.2.3. <i>Sistemas de Gestión de Contenido de Aprendizaje</i>	17
1.3. OBJETOS DE APRENDIZAJE	19
1.3.1. <i>Definición de Objeto de Aprendizaje</i>	19
1.3.2. <i>Características de los Objetos de Aprendizaje</i>	20
1.3.3. <i>Ventajas de los Objetos de Aprendizaje</i>	20
1.3.4. <i>Clasificación de los Objetos de Aprendizaje</i>	21
1.3.5. <i>Problemas computacionales clásicos</i>	22
1.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	23
1.4.1. <i>Metodologías Tradicionales</i>	24
1.4.2. <i>Metodologías Ágiles</i>	24
1.5. CONCLUSIONES PARCIALES.....	28
CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO.....	29
2.1. TECNOLOGÍAS Y HERRAMIENTAS PARA EL DESARROLLO DE LA SOLUCIÓN	29
2.1.1. <i>Sistema de Gestión de Contenidos</i>	29
2.1.2. <i>Lenguajes de programación utilizados</i>	31
2.1.3. <i>Técnica de programación AJAX</i>	33
2.1.4. <i>Bibliotecas</i>	34
2.1.5. <i>Lenguaje Unificado de Modelado</i>	34
2.1.6. <i>Herramienta de modelado: Visual Paradigm</i>	35
2.1.7. <i>Servicios Web de Java 7</i>	35
2.1.8. <i>Servidor Web Apache 2.2</i>	36
2.1.9. <i>Entorno de Desarrollo Integrado</i>	36
2.1.10. <i>Sistema Gestor de Base de Datos</i>	37
2.1.11. <i>Herramienta utilizada para las pruebas unitarias: JUnit 4.10</i>	37

2.2.	HISTORIAS DE USUARIO.....	38
2.3.	REQUISITOS NO FUNCIONALES	40
2.4.	PERSONAS RELACIONADAS CON EL SISTEMA	41
2.5.	ESTIMACIÓN DE ESFUERZO POR HISTORIA DE USUARIO.....	42
2.6.	PLAN DE ENTREGAS	42
2.7.	PROPUESTA DEL SISTEMA	42
2.8.	PATRÓN DE ARQUITECTURA	43
2.9.	PATRONES DE DISEÑO	45
2.10.	TARJETAS CLASE-RESPONSABILIDADES-COLABORADORES (CRC).....	47
2.11.	CONCLUSIONES PARCIALES	50
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....		51
3.1.	PLANEACIÓN.....	51
3.1.1.	<i>Plan de lanzamiento</i>	51
3.2.	ITERACIONES.....	51
3.2.1.	<i>Primera iteración</i>	51
3.2.2.	<i>Segunda iteración</i>	56
3.2.3.	<i>Tercera iteración</i>	58
3.3.	RESUMEN DEL RESULTADO DE LAS ITERACIONES	62
3.4.	PRUEBAS AL SISTEMA	63
3.5.	CASO DE ESTUDIO DEL OBJETO DE APRENDIZAJE ANÁLISIS DE LAS SOLUCIONES DEL PROBLEMA VIAJANTE MEDIANTE ALGORITMOS GENÉTICOS.	65
3.6.	CONCLUSIONES PARCIALES	66
CONCLUSIONES GENERALES		67
RECOMENDACIONES		68
REFERENCIAS BIBLIOGRÁFICAS.....		69
ANEXOS.....		72
ANEXO 1 VENTAJAS DE LOS OBJETOS DE APRENDIZAJE		72

ÍNDICE DE TABLAS

<i>Tabla 1 HU-1 Representación gráfica para el problema “El viajante”: Elaboración propia.</i>	39
<i>Tabla 2 HU-2: Representación gráfica para el problema “Mochila 01”: Elaboración propia.</i>	39
<i>Tabla 3 Personas relacionadas con el sistema: Elaboración propia.</i>	41
<i>Tabla 4 Estimación de esfuerzo por Historia de Usuario: Elaboración propia.</i>	42
<i>Tabla 5 Plan de entrega: Elaboración propia.</i>	42
<i>Tabla 6 CRC 1.</i>	47
<i>Tabla 7 CRC 2.</i>	47
<i>Tabla 8 CRC 3.</i>	47
<i>Tabla 9 CRC 4.</i>	47
<i>Tabla 10 CRC 5.</i>	47
<i>Tabla 11 CRC 6.</i>	48
<i>Tabla 12 CRC 7.</i>	48
<i>Tabla 13 CRC 8.</i>	48
<i>Tabla 14 CRC 9.</i>	48
<i>Tabla 15 Tarea de Ingeniería No1. Implementación de la interfaz principal: Elaboración propia.</i>	51
<i>Tabla 16 HU-1: Tarea de Ingeniería No2. Implementación del problema “El viajante” por Algoritmos Genéticos: Elaboración propia.</i>	52
<i>Tabla 17 Tarea de Ingeniería No3. Vinculación de los elementos gráficos a la solución del problema.</i>	52
<i>Tabla 18 Ejecutar algoritmo genético para el problema El viajante: Elaboración propia.</i>	54
<i>Tabla 19 Mostrar los resultados del algoritmo genético para el problema “El viajante”: Elaboración propia.</i>	54
<i>Tabla 20 Mostrar paginado de los resultados del algoritmo genético para el problema “El viajante”: Elaboración propia.</i>	55
<i>Tabla 21 Mostrar los mejores resultados del algoritmo genético para el problema “El viajante”: Elaboración propia.</i>	55
<i>Tabla 22 HU-2: Tarea de Ingeniería No1. Implementación del problema “Mochila 01” por Algoritmos Genéticos: Elaboración propia.</i>	58
<i>Tabla 23 HU-2: Tarea de Ingeniería No2. Vinculación de los elementos gráficos a la solución del problema “Mochila 01” por Algoritmos Genéticos: Elaboración propia.</i>	58
<i>Tabla 24 Ejecutar algoritmo genético para el problema “Mochila 01”: Elaboración propia.</i>	61

<i>Tabla 25 Mostrar los resultados del algoritmo genético para el problema “Mochila 01”: Elaboración propia.....</i>	<i>61</i>
<i>Tabla 26 Mostrar paginado de los resultados del algoritmo genético para el problema “Mochila 01”: Elaboración propia.....</i>	<i>62</i>
<i>Tabla 27 Ventajas de los Objetos de Aprendizaje.....</i>	<i>72</i>

ÍNDICE DE FIGURAS

FIGURA 1 CICLO DE VIDA XP.	27
FIGURA 2 PROPUESTA DEL SISTEMA: ELABORACIÓN PROPIA.	43
FIGURA 4 MODELO ARQUITECTÓNICO PROPUESTO: ELABORACIÓN PROPIA.	45
FIGURA 5 DIAGRAMA DE CLASES PARA LA REPRESENTACIÓN DE LOS PATRONES: ELABORACIÓN PROPIA.	46
FIGURA 6 DIAGRAMA DE CLASES: ELABORACIÓN PROPIA.	49
FIGURA 7 PANTALLA DE INGRESO AL SISTEMA: ELABORACIÓN PROPIA.	53
FIGURA 8 FORMULARIO DEL PROBLEMA 'EL VIAJANTE': ELABORACIÓN PROPIA.	57
FIGURA 9 VISUALIZACIÓN AL PROBLEMA 'EL VIAJANTE': ELABORACIÓN PROPIA.	57
FIGURA 10 FORMULARIO DEL PROBLEMA 'MOCHILA 01: ELABORACIÓN PROPIA'.....	60
FIGURA 11 VISUALIZACIÓN DEL PROBLEMA 'MOCHILA 01': ELABORACIÓN PROPIA.	60
FIGURA 12 RESUMEN DE LOS RESULTADOS DE LAS PRUEBAS REALIZADAS: ELABORACIÓN PROPIA.	63
FIGURA 13 EJEMPLO DE PRUEBA UNITARIA CON ERROR: ELABORACIÓN PROPIA.	64
FIGURA 14 EJEMPLO DE PRUEBA UNITARIA SIN ERROR: ELABORACIÓN PROPIA.	64

Introducción

Las Tecnologías de la Información y las Comunicaciones (TICs), se han constituido como uno de los recursos más importantes de la sociedad, trayendo como consecuencia una explosión en la transmisión e intercambio de datos, información y conocimientos, a los cuales se puede acceder sin tener en cuenta las barreras geográficas. Es por ello que en los últimos años casi todos los países del mundo han establecido e implementado proyectos, políticas y estrategias para promover su uso y aprovechar los beneficios y aportes que ofrecen. (1)

Cuba, no se ha quedado atrás en cuanto a la implementación y desarrollo de las TICs. A pesar de las limitaciones económicas en los años 90 se comenzaron a dar los primeros pasos, tal fue así que en el año 2000 se crea el Ministerio de la Informática y las Comunicaciones como estrategia vital para el desarrollo de las TICs en Cuba. También la Educación Superior ha tenido un papel trascendental en la aplicación de estas tecnologías. En el 2002 surge la Universidad de las Ciencias Informáticas (UCI) como proyecto para la masificación de las Tecnologías de la Información y las Comunicaciones y en el 2007 se crean tres facultades regionales de la Universidad de las Ciencias Informáticas para incrementar los graduados en la carrera de Ingeniería en Ciencias Informáticas. (2)

Las TICs se implementan en todos los niveles de enseñanza y cobran especial fuerza en la universalización de la Educación Superior, cuyo impacto fundamental se centra en su uso como fuente de información, canal de comunicación, instrumento cognitivo y de procesamiento de la información. Esto impone nuevos roles para los protagonistas del proceso formativo e implican nuevos retos para el profesional del futuro y las instituciones formadoras, las que han de lidiar con aspectos técnicos, formación especializada, seguridad informática y otros elementos que determinan la expansión de las TICs. Según Salinas (3): *"La utilización de las Tecnologías de la Información y las Comunicaciones en la educación exige un aumento de la autonomía del alumno. A la superación de las barreras de la distancia y el tiempo para acceder al aprendizaje, se añade mayor interacción y la oportunidad de controlar las actividades de aprendizaje y compartirlas mediante la intercomunicación en un marco de apoyo y colaboración"*.

En la UCI se utilizan las TICs como apoyo a Proceso de Enseñanza-Aprendizaje (PEA), mediante la implementación de herramientas que apoyan la calidad de la educación; entre estas se puede mencionar el Entorno Virtual de Aprendizaje (EVA) donde se pueden encontrar conferencias, clases prácticas, bibliografía y un sistema de evaluación que permite ejercitar los conocimientos aprendidos. Como parte de los cursos que contiene dicha plataforma de aprendizaje se encuentra la disciplina Inteligencia Artificial (IA). La misma cuenta con la asignatura Inteligencia Artificial I estructurada en varios temas, entre los que se encuentra

Métodos de Solución de Problemas (MSP), en el cual los estudiantes presentan mayores dificultades por su abstracción y poca ejemplificación, específicamente en el contenido relacionado con Algoritmos Genéticos.

Debido a que los resultados académicos no han sido satisfactorios y al hecho de que existe una deficiencia de materiales didácticos para enriquecer el proceso docente y apoyar los resultados en la enseñanza se deriva la siguiente **situación problemática**:

En la Universidad de las Ciencias Informáticas no se cuenta con suficientes recursos didácticos para apoyar el tema Métodos de Solución de Problemas de la asignatura Inteligencia Artificial I, específicamente el contenido relacionado con los Algoritmos Genéticos, que les permita a los estudiantes una mayor comprensión del contenido y contribuyan a su autosuperación, a la vez que brindan un complemento a las clases del profesor.

En base a lo anterior se plantea como **problema a resolver**: ¿Cómo contribuir al Proceso de Enseñanza-Aprendizaje de la disciplina Inteligencia Artificial a través del desarrollo de Objetos de Aprendizaje?

Se define como **objeto de estudio**, el Proceso de Enseñanza-Aprendizaje de la disciplina Inteligencia Artificial. Enmarcado en el **campo de acción**, Objetos de Aprendizaje de apoyo a la asignatura Inteligencia Artificial I en la Universidad de las Ciencias Informáticas.

Para resolver el problema planteado, se propone como **objetivo general**: Desarrollar una aplicación web con soporte para Objetos de Aprendizaje que apoyen el Proceso de Enseñanza-Aprendizaje de la asignatura Inteligencia Artificial I en el tema Métodos de Solución de Problemas.

A partir de esto, se proponen como **objetivos específicos**:

- Definir los Objetos de Aprendizaje más adecuados para apoyar el Proceso de Enseñanza-Aprendizaje en la asignatura Inteligencia Artificial I, en el tema Métodos de Solución de Problemas.
- Definir problemas clásicos de la Inteligencia Artificial que puedan ser resueltos con los Métodos de Solución de Problemas impartidos en la asignatura Inteligencia Artificial I.
- Diseñar una aplicación web que soporte los diferentes Objetos de Aprendizaje definidos.
- Implementar los Objetos de Aprendizaje definidos con anterioridad.
- Validar la solución propuesta.

Para la realización de estos objetivos se proponen las siguientes **tareas de la investigación**:

- Elaborar el marco teórico de la investigación.

- Caracterizar los tipos de Objetos de Aprendizaje que apoyen el proceso de Enseñanza-Aprendizaje en la disciplina de Inteligencia Artificial.
- Caracterizar los problemas computacionales clásicos que pueden ser resueltos con los Métodos de Solución de Problemas impartidos en la asignatura.
- Caracterizar las herramientas de programación web que sirvan de soporte para la implementación de la aplicación.
- Elaborar un caso de estudio que permita ilustrar las ventajas de la aplicación de los Objetos de Aprendizaje como apoyo al Proceso de Enseñanza-Aprendizaje en la disciplina Inteligencia Artificial.

Para dar cumplimiento a las tareas se utilizaron los siguientes métodos teóricos: **analítico-sintético** el cual se empleó para analizar la bibliografía permitiendo recopilar, extraer y sintetizar los elementos más significativos relacionados con los Objetos de Aprendizaje, tecnologías web, herramientas de desarrollo, metodologías y lenguajes de programación, así como el **histórico-lógico** para relacionar de manera coherente los contenidos del tema Métodos de Solución de Problemas en la asignatura Inteligencia Artificial I.

El documento estará estructurado de la siguiente manera.

Capítulo 1 Fundamentación Teórica: se hace referencia a algunos conceptos necesarios para el desarrollo de aplicaciones para e-learning (aprendizaje en línea). Se tratan los distintos tipos de e-learning que existen en la actualidad, los Objetos de Aprendizaje, Sistemas de Gestión de Contenido (CMS, por sus siglas en inglés), Sistemas de Gestión de Aprendizaje (LMS, por sus siglas en inglés), Sistemas de Gestión de Contenidos de Aprendizaje (LCMS, por sus siglas en inglés), en qué consisten y para qué se utilizan. Por último se da una explicación acerca del uso de algunas metodologías para el desarrollo de sistemas informáticos, haciendo énfasis en la que se propone para el desarrollo de este trabajo.

Capítulo 2 Planificación y Diseño: se presentan los aspectos relacionados a las actividades que propone la metodología XP para el desarrollo de la aplicación “Objetos de Aprendizajes para apoyar la disciplina Inteligencia Artificial”, durante las fases de Exploración y Planificación, dichas actividades están relacionadas con: la selección de las tecnologías y herramientas a utilizar, descripción de los requisitos funcionales a través de historias de usuario y la propuesta del sistemas a desarrollar.

Capítulo 3 Implementación y Prueba: muestra las actividades del marco de trabajo propuestas por la metodología XP, Implementación y Pruebas. Por cada iteración se realiza una planificación para darle cumplimiento a la historia de usuario seleccionada, así como el conjunto

de tareas que se ejecutan para su terminación. Finalmente cada iteración concluye con un conjunto de pruebas de aceptación que permiten validar el cumplimiento de los requisitos.

Capítulo 1: Fundamentación Teórica

Introducción

En el capítulo se hace referencia a algunos conceptos necesarios para el desarrollo de aplicaciones para e-learning (aprendizaje en línea). Se tratan los distintos tipos de e-learning que existen en la actualidad, los Objetos de Aprendizaje, Sistemas de Gestión de Contenido (CMS, por sus siglas en inglés), Sistemas de Gestión de Aprendizaje (LMS, por sus siglas en inglés), Sistemas de Gestión de Contenidos de Aprendizaje (LCMS, por sus siglas en inglés), en qué consisten y para qué se utilizan. Por último se da una explicación acerca del uso de algunas metodologías para el desarrollo de sistemas informáticos, haciendo énfasis en la que se propone para el desarrollo de este trabajo.

1.1. El e-learning como modalidad educativa

1.1.1. ¿Qué es el e-learning?

Desde el surgimiento de Internet, las posibilidades de acceso a la formación se han ido incrementando en la medida en que la red nos va permitiendo acceder a más personas y ofrecer ambientes de aprendizaje más complejos y elaborados. El término e-learning se refiere a la utilización de las nuevas Tecnologías de la Información y las Comunicaciones con un propósito de aprendizaje (4).

Una de las principales ventajas del e-learning es la facilidad de acceso. La formación puede llegar a más personas, puesto que desaparecen las barreras espacio-temporales. De esta forma, personas que antes tenían dificultades para estar en contacto continuo con los procesos de formación debido a: problemas de desplazamiento al centro donde se imparten los cursos, escasez de tiempo, incapacidad física para asistir a clase, tienen ahora todo un abanico de posibilidades puestas a su disposición para una continua formación.

E-Learning es una modalidad formativa que permite una educación a distancia donde los alumnos acceden a los contenidos, actividades y tareas del curso a través de las plataformas tecnológicas, no compartiendo necesariamente con los profesores un mismo espacio físico. Permite además una formación semi-presencial, asistiendo los alumnos solo a determinadas sesiones presenciales orientadas a resolver dudas, observar y practicar.

1.2. Herramientas para el acceso al e-learning

El repaso de los progresos de las soluciones e-learning en los últimos años revela la aparición y desarrollo de plataformas como los CMS, LSM, LCMS. A continuación se explican cuáles son las principales características de estos sistemas.

1.2.1. Sistemas de Gestión de Contenido

Se conoce como CMS a las aplicaciones que en la industria de las publicaciones en línea permiten la creación y administración de contenido. El objetivo de estos programas informáticos es la creación y la gestión de información en línea, dicha información está compuesta por: textos (artículos, informes), gráficos, videos y sonido. Para la creación de la información a servir se utiliza la estrategia de separar el contenido de la presentación. De hecho, las personas que se encargan de cada aspecto suelen ser distintas, además la información se construye ensamblando porciones de contenidos que se denominan “componentes”, esos pedazos de información pueden considerarse independientes entre sí y su característica más importante es que son reutilizables.

Además, la utilización de componentes reutilizables permite personalizar la información a la medida del usuario que la consume. Estas ideas justifican el éxito de los CMS, y son perfectamente trasladables al mundo del e-learning. Desafortunadamente estos sistemas no tienen la capacidad de gestionar el proceso de aprendizaje, lo cual es esencial en todo proyecto de e-learning. (5)

1.2.2. Sistemas de Gestión de Aprendizaje

Surgieron como respuesta a la necesidad de gestionar el proceso de aprendizaje. Permiten planificar el aprendizaje de acuerdo a las necesidades de los usuarios, sean estos estudiantes, trabajadores o empresas, así como ayudan a mejorar las competencias de los usuarios de los cursos y su intercomunicación. Es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional. Admiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación en general.

La implementación de una plataforma LMS no garantiza, sin embargo, los medios para la creación y generación de los cursos necesarios para la organización. Desde la perspectiva de los materiales docentes actúa como plataforma de distribución donde se remarca la idea de que en un sistema LMS la mínima unidad de instrucción es el curso en sí mismo. Los LMS permiten la gestión de los usuarios, pero demuestran carencias en la gestión de los contenidos. (5)

1.2.3. Sistemas de Gestión de Contenido de Aprendizaje

Los LCMS representan la integración de dos vías tradicionalmente separadas: los CMS y los LMS. Estos dos mundos se han desarrollado de espaldas entre sí, ajenos a una realidad incuestionable. El aprendizaje a través de Internet necesariamente requiere de recursos que permitan tanto la creación como la distribución de contenidos integrados en una misma plataforma. Esto permite a expertos en cualquier área del saber, no necesariamente a expertos

Capítulo 1: Fundamentación Teórica

en el manejo del software específico, generar materiales, diseñar, crear, distribuir y controlar el proceso de aprendizaje de una forma sencilla, rápida y eficiente.

Aparentemente los LCMS podrían contribuir a resolver muchos de los problemas anteriormente mencionados. Por lo tanto, una plataforma LCMS además de garantizar el control del proceso de aprendizaje, debe facilitar la creación, almacenamiento y reparto de los contenidos, con las siguientes características (5):

- Herramientas sencillas que facilitan la creación de contenidos, en forma de herramientas de autor, incluyendo editores con el fin de eliminar la necesidad de manejar editores de Lenguajes de Marcas de Hipertexto (HTML, por sus siglas en inglés).
- Sistemas flexibles de diseño y distribución de los cursos que permitan adaptarse a las necesidades de la organización y a los diferentes sistemas y ritmos de aprendizaje de los usuarios.
- Posibilidad de rehusar los Objetos de Aprendizaje.
- Conectividad con otros LCMS y en general adecuación a los estándares actuales.
- Herramientas para la comunicación y el aprendizaje colaborativo. Incluye recursos tanto síncronos como asíncronos que faciliten la comunicación sencilla entre iguales y con los profesores.
- Sencillez en la migración de contenidos para facilitar la adaptación a las diferentes necesidades y escenarios de formación que se puedan presentar.

Selección de la herramienta para el acceso al e-learning

Después de hacer un estudio de las herramientas utilizadas para el acceso a e-learning, se llega a la conclusión que para el desarrollo de la aplicación se seleccionará un CMS, dicha selección se basa en los beneficios que aporta, entre los que se encuentran (6):

- **Uso sencillo:** no es necesario tener conocimientos de programación para publicar y gestionar contenido dinámico o estático.
- **Escalabilidad e implementación de nuevas funcionalidades:** ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos que agregarán servicios de valor a la aplicación.
- **Organización de la aplicación:** organiza los contenidos en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla.
- **Reutilización de objetos o componentes:** permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.

- **Gestión independiente de contenido y diseño de la aplicación:** permite cambiar la apariencia de la aplicación sin perder los contenidos o sin necesidad de darles formato nuevamente.
- **Seguridad:** actualizaciones de seguridad frecuentes, protocolos de encriptación de información sensible y buen entendimiento con las opciones de seguridad del propio servidor.
- **Control de acceso:** gestiona los permisos de administración, creación, modificación de contenidos y configuración de módulos aplicados a grupos o individuos.

1.3. Objetos de aprendizaje

La web puede ser considerada como una “gran colección de objetos digitales”. Se pueden encontrar recursos de diferentes formatos y para aplicaciones de todo tipo y sobre cualquier tema. Sin embargo, el problema es que esta colección no tiene un orden, la búsqueda es compleja y la permanencia de la información es impredecible. Para que los recursos de información disponibles en la web sean verdaderamente explotables se han dado iniciativas y tecnologías para la organización de los recursos. (7)

Los OA, también llamados objetos didácticos (8), surgen con el interés de compartir recursos y para su reutilización en el ámbito educativo, originados bajo el paradigma de la orientación a objetos. Es aplicado a *“materiales digitales creados como pequeñas piezas de contenido o de información, con la finalidad de maximizar el número de situaciones educativas en que las que el recurso pueda ser utilizado.”* (7)

1.3.1. Definición de Objeto de Aprendizaje

De acuerdo a los diferentes conceptos encontrados, un objeto de aprendizaje es:

- *“Una entidad digital, autocontenible y reutilizable, con un claro propósito educativo, constituido por al menos tres componentes editables: contenidos, actividades de aprendizaje y elementos de contextualización. A manera de complemento, los objetos de aprendizaje han de tener una estructura (externa) de información que facilite su identificación, almacenamiento y recuperación: los metadatos”.* (9)
- *“Una estructura (distribución, organización) autónoma que contiene un objetivo general, objetivos específicos, una actividad de aprendizaje, un metadato (estructura de información externa) y por ende, mecanismos de evaluación y ponderación, el cual puede ser desarrollado con elementos multimedia con el fin de posibilitar su reutilización, interoperabilidad, accesibilidad y duración en el tiempo. Un Objeto de Aprendizaje puede estar constituido al menos con los siguientes componentes: Contenido (s), actividad (es) de aprendizaje y un contexto. Un Objeto de Aprendizaje puede ser incorporado,*

Capítulo 1: Fundamentación Teórica

instalado, configurado en una plataforma de Gestión de Aprendizaje o LMS (Learning Management System)." (10)

Se concluye que los Objetos de Aprendizaje son: cualquier recurso digital que se pueda reutilizar para apoyar el aprendizaje y debe cumplir con las siguientes características: reusabilidad, interoperabilidad, accesibilidad y duración en el tiempo.

1.3.2. Características de los Objetos de Aprendizaje

Formato digital: capacidad de actualización y/o modificación constante; es decir, es utilizable desde Internet y accesible a muchas personas simultáneamente y desde distintos lugares.

Propósito pedagógico: el objetivo es asegurar un proceso de aprendizaje satisfactorio. Por tanto, el Objeto de Aprendizaje incluye no sólo los contenidos sino que también guía el propio proceso de aprendizaje del estudiante.

Contenido interactivo: implica la participación activa de cada individuo (profesor-alumno/s) en el intercambio de información. Para ello es necesario que el objeto incluya actividades (ejercicios, simulaciones, diagramas, gráficos, exámenes, experimentos.) que permitan facilitar el proceso de asimilación y el seguimiento del progreso de cada alumno.

Es **indivisible** e **independiente** de otros objetos de aprendizaje, por lo que:

- Debe tener sentido en sí mismo.
- No puede descomponerse en partes más pequeñas.

Es **reutilizable** en contextos educativos distintos. Esta característica es la que determina que un objeto tenga valor, siendo uno de los principios que fundamentan el concepto de objeto de aprendizaje.

Para que un objeto de aprendizaje pueda ser reutilizable es necesario que:

- Los contenidos no estén contextualizados (no hacer referencia a su ubicación ni en la asignatura, ni en la titulación, ni en el tiempo).
- Se determinen algunos de los posibles contextos de uso, facilitando el proceso posterior de rediseño e implementación.
- Se le otorguen previamente una serie de características identificativas o atributos (metadatos) que permitan distinguirlos de otros objetos para ser buscado y encontrado fácilmente. Junto con otros objetos, se pueden alcanzar objetivos de aprendizaje más amplios, llevando a la construcción de los llamados: módulos de aprendizaje.

1.3.3. Ventajas de los Objetos de Aprendizaje

Ventajas para los educadores:

- Facilita la búsqueda de contenido existente.

- Permiten su reutilización en diferentes contextos educativos y para diferentes alumnos.
- Son accesibles en el mediano y largo plazo.

Ventajas para los Estudiantes:

- La apariencia de los recursos instruccionales ¹promueve la comodidad.
- Sirve para una variedad de estilos de aprendizaje individuales.

En la Tabla 27: Ventajas de los Objetos de Aprendizaje (ver Anexo 1), pueden verse las demás ventajas que presentan los objetos de aprendizaje, tanto para los estudiantes como para los profesores.

1.3.4. Clasificación de los Objetos de Aprendizaje

Se pueden clasificar los objetos de aprendizaje atendiendo al tipo de contenido pedagógico y al formato (11):

Según los contenidos pedagógicos:

Conceptuales

Hechos, datos y conceptos (leyes, teoremas). Un concepto se adquiere cuando se es capaz de dotar de significado a un material o a una información que se presenta; se trata de traducir el concepto a nuestras propias palabras.

Procedimentales

Un procedimiento es un conjunto de acciones ordenadas, orientadas a la consecución de una meta, por consiguiente hablar de procedimientos implica el aprendizaje de un “saber hacer”, con un propósito claramente definido y que se espera realizar de manera ordenada. Es algo práctico.

Actitudinales

Son tendencias, o disposiciones adquiridas y relativamente duraderas, a evaluar de un modo determinado un objeto, una persona, suceso o situación y actuar en consonancia en dicha evaluación. Los contenidos actitudinales se clasifican en valores, actitudes y normas.

Según el formato:

- Imagen.
- Texto.
- Sonido.
- Multimedia.

¹recursos instruccionales: Se concibe como recurso instruccional a cualquier dispositivo o material que transmita el mensaje requerido para el logro de un aprendizaje. Sirven para transferir la información de manera creativa contribuyendo a la comprensión y asimilación de la información de una manera rápida, fácil y útil para los estudiantes

Selección de los Objetos de Aprendizaje

Teniendo en cuenta la anterior clasificación de los objetos de aprendizaje se define que los más adecuados para apoyar el Proceso de Enseñanza-Aprendizaje de la asignatura Inteligencia Artificial I en el tema Métodos de Solución de Problemas son: según su contenido pedagógico los OA conceptuales y procedimentales, debido a que la combinación de ambos tipos permite enriquecer el contenido informativo del objeto, mostrándole al estudiante los hechos, datos y conceptos de un problema determinado y una forma práctica de aplicarlos en la vida real. Según el formato los OA estarán compuestos por imágenes animadas y texto, permitiendo la visualización de un contenido agradable que capte la atención del estudiante.

1.3.5. Problemas computacionales clásicos

Para el desarrollo de los objetos de aprendizaje definidos es necesario seleccionar un conjunto de problemas computacionales clásicos que puedan ser resueltos con técnicas de inteligencia artificial como son los métodos de solución de problemas. Para ello se describen a continuación algunos problemas clásicos que pueden ser resueltos con dichas técnicas.

El viajante o vendedor viajero

El **problema del viajante** o **problema del vendedor viajero** (TSP, por sus siglas en inglés) es uno de los problemas más famosos en el campo de la optimización combinatoria computacional. Se conoce la forma de resolverlo pero sólo en teoría, en la práctica la solución no es aplicable debido al tiempo que computacionalmente se precisa para obtener su resultado. El mismo consiste en *“encontrar entre un conjunto de ciudades, una ruta que partiendo de una ciudad y regresando a ella, recorra todas las ciudades por el camino más corto.”* (12) El problema tiene considerables aplicaciones prácticas, aparte de las más evidentes en áreas de logística de transporte. En robótica, permite resolver problemas de fabricación para minimizar el número de desplazamientos al realizar una serie de perforaciones en una plancha o en un circuito impreso. También puede ser utilizado en control y operativa optimizada de semáforos. (13)

La mochila

El **problema de la mochila** (KP, por sus siglas en inglés) es un problema de optimización combinatoria. Modela una situación análoga al llenar una mochila, incapaz de soportar más de un peso determinado, con todo o parte de un conjunto de objetos, cada uno con un peso y valor específicos. Los objetos colocados en la mochila deben maximizar el valor total sin exceder el peso máximo. El modelo en sí mismo se puede aplicar en casos tales como la inversión de capital, embalaje, carga de vehículos, presupuesto. (14)

Asignación de tareas

El **problema de asignación de tareas** consiste en encontrar un emparejamiento de peso máximo en un grafo bipartido ponderado. Es uno de los problemas fundamentales de optimización combinatoria de la rama de optimización o investigación operativa en matemática. El problema de asignación tiene que ver con la designación de tareas a empleados, de territorios a vendedores, de contratos a postores o de trabajos a plantas. En otras palabras, a la disposición de algunos recursos (maquinas o personas) para la realización de ciertos productos a costo mínimo. (15)

Coloración de grafos

En teoría de grafos, la **coloración de grafos** es un caso especial de etiquetado de grafos; es una asignación de etiquetas llamadas colores a elementos del grafo. De manera simple, es una coloración de los vértices de un grafo tal que ningún vértice adyacente comparta el mismo color. En representaciones matemáticas y computacionales es típicamente usado enteros no-negativos como colores. En general se puede usar un conjunto finito como conjunto de colores. La naturaleza del problema de coloración depende del número de colores pero no sobre cuáles son. (16)

Selección de problemas computacionales clásicos

Luego de ser realizado el estudio de estos problemas, se determina que los más idóneos para una solución práctica representada en un problema cotidiano son: **el viajante** y **mochila**, dicha selección se debe a que estos dos problemas son situaciones bien conocidas por los estudiantes desde años anteriores, la ventaja es que ya conocen su solución teórica, de modo que el objeto de aprendizaje se enfoca en una solución práctica de los mismos. Por otro lado, ambos problemas están en constante investigación por ser problemas de optimización complejos para los cuales no existe una solución definitiva. Además los mismos representan una clase de problemas, pudiéndose modelar muchos problemas prácticos a partir de la descripción de estos.

1.4. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. En ellas se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

1.4.1. Metodologías Tradicionales

Se centran en la definición detallada de los procesos, tareas y herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Se caracterizan por realizar un tratamiento predictivo de los proyectos y medir el progreso en términos de artefactos entregados, así como de la especificación de los requisitos, los documentos de diseño, planes de pruebas y revisiones de código.

Proceso Unificado Racional (RUP)

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como características esenciales está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera depurada.

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. (17)

- Concepción: 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.

1.4.2. Metodologías Ágiles

Este tipo de metodologías surgen a finales de la década del 90 como alternativa a los inconvenientes de las metodologías tradicionales.

Las principales ideas de la metodología son:

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- Es más importante la obtención de un producto de software que funcione, que la documentación de todo el proceso de desarrollo.
- El cliente está en todo momento colaborando en el proyecto.

Capítulo 1: Fundamentación Teórica

- Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

Existen muchas metodologías ágiles, según un estudio comparativo realizado en el año 2008, destacan las seis que se relacionan a continuación (18):

- Agile Project Management.
- Crystal Methods
- Dynamic Systems Development Method.
- Scrum.
- Test Driven Development.
- Extreme Programming.

XP consigue abordar la mayoría de los parámetros de calidad e introduce un conjunto de herramientas que implementan técnicas concretas, especialmente en las fases de pruebas. (18)

Programación Extrema (XP)

Algunas de las principales características de XP son las siguientes: el cliente es parte del equipo de desarrollo, no existe un contrato tradicional al que darle cumplimiento y es utilizada para proyectos considerados pequeños.

Dicha metodología utiliza un enfoque orientado a objetos como su paradigma de desarrollo preferido, también abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño, codificación y prueba. El equipo de desarrollo estará integrado por tres personas en su totalidad.

Planeación: la actividad de planeación comienza creando una serie de historias (también llamadas historias de usuario) que describen las características y la funcionalidad requeridas para el software que se construirá.

Diseño: el diseño de la Programación Extrema sigue de manera rigurosa el principio Mantenerlo Simple (MS). Siempre se prefiere un diseño simple respecto de una presentación más compleja. Además, el diseño ofrece una guía de implementación para una historia como está escrita, ni más ni menos. Se desaprueba el diseño de funcionalidad extra (porque el desarrollador supone que se requerirá más tarde).

Codificación: la Programación Extrema recomienda que después de diseñar las historias y realizar el trabajo de diseño preliminar el equipo no debe moverse hacia la codificación, sino que debe desarrollar una serie de pruebas unitarias que ejerciten cada una de las historias que vayan a incluirse en el lanzamiento actual (incremento de software). Una vez creada la prueba de unidad, el desarrollador es más capaz de centrarse en lo que debe implementarse para

Capítulo 1: Fundamentación Teórica

pasar la prueba de unidad. No se agrega nada extraño MS. Una vez que el código está completo, la unidad puede probarse de inmediato, y así proporcionar una retroalimentación instantánea a los desarrolladores.

Pruebas: la creación de una prueba de unidad antes de comenzar la codificación es un elemento clave para el enfoque de la Programación Extrema. Las pruebas de unidad que se crean deben implementarse con un marco de trabajo que permita automatizarlas (por lo tanto, pueden ejecutarse de manera fácil y repetida). (19)

Selección de la Metodología

Después de hacer un estudio de las metodologías expuestas anteriormente, se llega a la conclusión que para el desarrollo de este trabajo la más adaptable a la aplicación es la Programación Extrema (XP), una de las más exitosas en la actualidad, utilizada para proyectos de corto plazo y de equipos pequeños. Dentro de los objetivos de esta metodología se encuentra: la satisfacción del cliente, dicha metodología trata de dar al cliente el software que necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del mismo, incluso cuando los cambios sean al final del ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Para guiar el desarrollo de la solución se decide asumir la metodología XP basado en los aspectos vistos anteriormente y los siguientes:

- El equipo de desarrollo estará integrado por tres personas en su totalidad.
- El cliente es parte del equipo de desarrollo.
- No existe un contrato tradicional al que darle cumplimiento.
- Es un proyecto considerado pequeño.

Con el fin de definir cada una de las etapas y sus artefactos en el transcurso del desarrollo de la aplicación, se decidió ajustar el ciclo de vida a la propuesta de solución como sigue:

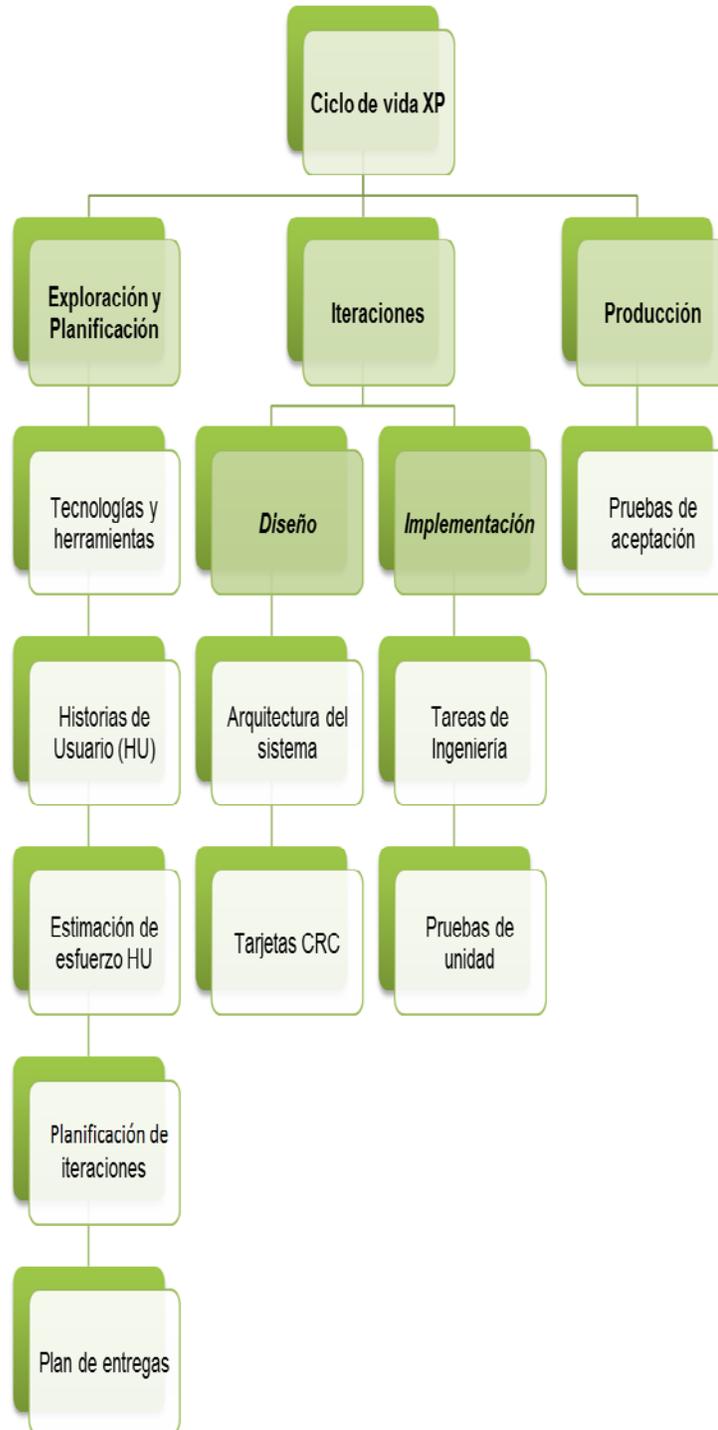


Figura 1 Ciclo de vida XP.

1.5. Conclusiones parciales

En el capítulo se elaboró el marco teórico de la investigación tratándose los distintos tipos de e-learning que existen en la actualidad, se escoge como herramienta de acceso para e-learning un CMS, el cual se utiliza para la creación y la gestión de información en línea. Además se manifiestan aspectos sobre los Objetos de Aprendizaje definiendo los más adecuados para apoyar el PEA de la asignatura IA I: según el contenido pedagógico: procedimental y conceptual y según el formato: imágenes animadas y texto. Como metodología de desarrollo de software se decide utilizar Programación Extrema, debido a que el cliente forma parte del equipo de desarrollo.

Capítulo 2: Planificación y Diseño

Introducción

En este capítulo se presentan los aspectos relacionados a las actividades que propone la metodología XP para el desarrollo de la aplicación “Objetos de Aprendizajes para apoyar la disciplina Inteligencia Artificial”, durante las fases de Exploración y Planificación, dichas actividades están relacionadas con: la selección de las tecnologías y herramientas a utilizar, descripción de los requisitos funcionales a través de historias de usuario y la propuesta del sistemas a desarrollar.

2.1. Tecnologías y Herramientas para el desarrollo de la solución

En esta sección se exponen las herramientas y tecnologías que serán utilizadas para dar cumplimiento al objetivo general de la investigación. La elección de las mismas se basa en el hecho de ser parte del movimiento de Software Libre, además de poseer características que se adecuan a las necesidades del presente trabajo de diploma.

2.1.1. Sistema de Gestión de Contenidos

Un Sistema de Gestión de contenidos es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, que regularmente son páginas web.

Drupal

Es un CMS o Sistema de Gestión de Contenido modular y muy configurable. Es un programa de código abierto, bajo Licencia Pública General (GPL, por sus siglas en inglés), escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. Drupal es multiplataforma desde sus inicios. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. (20)

Permite publicar artículos, imágenes y servicios añadidos como: foros, encuestas, votaciones y blogs, además administra usuarios y permisos. En lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor, los almacena en una base de datos y se editan utilizando un entorno web.

Existen tres tipos de módulos de Drupal, llamadas las "3 C" (21):

- Core (núcleo): son los módulos provistos por Drupal al instalarse, algunos de ellos fueron contribuciones de su comunidad.
- Contributed (contribuciones): son los módulos compartidos por la comunidad.
- Custom (personalizados): son los módulos creados por el desarrollador de la aplicación.

Las ventajas para el cliente son:

- No se pagan licencias.
- Hay independencia de proveedores.
- El software está en constante evolución.
- Posee una comunidad consolidada, con diez años de experiencia.
- Es modular y escalable, se basa en un entorno de desarrollo, el cual hace posible a partir de pequeños módulos, modificar el funcionamiento de la aplicación. De este modo se puede adaptar a las necesidades del cliente.
- Permite publicar, administrar, consultar y acceder al contenido mediante un navegador.

Joomla

Joomla es un CMS (Content Management System), o lo que es lo mismo, Sistema de Gestión de Contenidos, tiene como principal objetivo la edición de páginas web de una forma sencilla.

Esta aplicación se distribuye en código abierto construida en su mayor parte en PHP.

Para utilizar este CMS se requieren ciertos requisitos (22):

- PHP 4.2 con las extensiones activadas de MySQL, XML y Zlib.
- MySQL 3.23.
- Servidor web Apache 1.3 o IIS 6.

Selección del Sistema de Gestión de contenido

Después de hacer un estudio de los Sistemas de Gestión de Contenidos expuestos anteriormente, se llega a la conclusión que para el desarrollo de este trabajo se seleccionará Drupal en su versión 7.19. Entre los principales beneficios que aporta su uso están (23):

- Arquitectura orientada a módulos, existen alrededor de 10,000 módulos contribuidos y el número sigue creciendo, esto permite extender las funcionalidades del sistema según las necesidades del cliente.
- Drupal hace que el desarrollo web y la gestión de contenidos sea rápido y fácil utilizando un completo Paquete de Construcción de Contenido (CCK).
- Al estar bajo una licencia de software libre es posible el desarrollo de mejoras y arreglo de errores por los miembros de su comunidad de desarrolladores.
- Permisos basados en roles.
- Capacidad multi-idomas.
- Control minucioso de permisos de usuarios y módulos.

Drupal se puede personalizar para muchas necesidades diferentes, incluyendo sitios web orientados a la educación.

2.1.2. Lenguajes de programación utilizados.

HTML5

HTML5 es la quinta revisión del lenguaje básico HTML. Esta versión es estable para su uso por parte de los desarrolladores y empresas. HTML5 responde a las necesidades de los últimos años, puede llegar a dispositivos móviles, autos y televisión. Esta nueva versión soporta experiencias de usuario de forma nativa que antes sólo podían ser desarrolladas utilizando Javascript, Java o Flash. Además soporta video, servicios de geolocalización y herramientas táctiles

Novedades de HTML5 (24):

- Incorpora etiquetas (canvas 2D y 3D, audio, video) con codecs para mostrar los contenidos multimedia.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime) y facilidades para validar el contenido sin Javascript.
- Añade etiquetas para manejar la Web Semántica (Web 3.0): header (cabecera), footer (pie de página), article (artículo), nav (navegación), time (fecha del contenido), link rel="" (tipo de contenido que se enlaza).

Entre los elementos que incorpora se encuentra canvas. Canvas es un mapa de bits con una interfaz de programación de aplicaciones (API, por sus siglas en inglés) para gráficos en modo inmediato para dibujar sobre él.

Canvas restituye los gráficos directamente dentro de su mapa de bits, permite que los desarrolladores proporcionen experiencias gráficas nuevas. Algunas de sus características son (25):

- Basado en píxeles.
- Elemento HTML único.
- Modificado mediante script solamente.

PHP 5.4.3

PHP es acrónimo de Hypertext Preprocessor, es usado principalmente en interpretación del lado del servidor. Fue uno de los primeros lenguajes de programación que se podían incorporar directamente en el documento HTML en lugar de utilizar un archivo externo que procesara los datos. El código es interpretado por un servidor web con un módulo de procesador PHP que genera la página web resultante. Además puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. (26)

Dentro de las principales características de PHP se pueden mencionar (27):

- Utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- Provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo php.ini. Dispone de una amplia gama de librerías.
- Puede interactuar con varios motores de bases de datos tales como MySQL, MS SQL, Oracle y PostgreSQL.
- Está escrito en C, esto permite su ejecución rápidamente utilizando poca memoria.

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos y dinámico. Se utiliza principalmente para la programación del lado del cliente. Se diseñó con una sintaxis similar a C aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Los navegadores modernos interpretan el código JavaScript integrado en las páginas y para interactuar con una página web se provee al lenguaje de una implementación del Document Object Model (DOM). (28)

Java 7

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems y publicado en el 1995 como un componente fundamental de la plataforma Java. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java pueden ejecutarse en cualquier máquina virtual Java sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Las características principales que ofrece Java son (29):

- **Lenguaje Simple:** se lo conoce como lenguaje simple porque proviene de la misma estructura de C y C++.
- **Orientado a Objeto:** la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.
- **Distribuido:** permite establecer y aceptar conexiones con los servidores o clientes remotos, facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.
- **Robusto:** en comparación con C, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

- **Seguro:** la seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.
- Indiferente a la arquitectura: java es compatible con los más variados entornos de red, sean estos desde Windows 95, Unix a Windows Nt y Mac, para poder trabajar con diferentes sistemas operativos.
- **Portable:** sus programas se mantienen iguales en cualquiera de las plataformas.
- Interpretado y compilado a la vez: java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.
- **Multihilos:** posee la capacidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo.
- **Dinámico:** las clases solamente actuaran en medida en que sean requeridas o necesitadas, con esto permitirá que los enlaces se puedan incluir incluso desde varias fuentes en la red.

2.1.3. Técnica de programación AJAX

Acrónimo de Asynchronous JavaScript and XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). AJAX es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model(DOM).

AJAX es una combinación de cuatro tecnologías existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones como JavaScript, para mostrar e interactuar dinámicamente con la información presentada.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano y JSON. (30)

2.1.4. Bibliotecas

JGAP 3.6.2

JGAP es una biblioteca de Java que contiene soluciones ya implementadas de los operadores genéticos más utilizados, así como puntos de extensión para crear tus propios algoritmos genéticos. Dicha biblioteca ha mejorado su robustez gracias a un mayor número de test. Su licencia es LGPL. (31)

Nusoap 0.9.5

Nusoap es una biblioteca escrita en PHP crear y consumir servicios. Está compuesto por una serie de clases que facilitan su desarrollo. Provee soporte para el desarrollo de clientes (aquellos que consumen los servicios web) y de servidores (aquellos que los proveen). Nusoap está basado en SOAP 1.1, WSDL 1.1² y HTTP 1.0/1.1. (32)

JQuery 1.7.2

jQuery es un framework (marco de trabajo) Javascript, es un producto que sirve como base para la programación avanzada de aplicaciones, aporta una serie de funciones o códigos para realizar tareas habituales. Estos frameworks desarrollan las tareas más básicas, puesto que contienen implementaciones probadas y funcionales.

Ofrece una infraestructura para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos y aplicaciones que hacen uso de Ajax.

Ventajas de jQuery con respecto a otras alternativas

Cada uno de los frameworks tiene sus ventajas e inconvenientes, jQuery es un producto con una buena aceptación por parte de los programadores y un grado de penetración en el mercado muy amplio, que lo hacen una de las mejores opciones. Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del mismo. Además posee una amplia comunidad de creadores de plugins o componentes, que extienden sus funcionalidades. (33)

2.1.5. Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), permite la representación de sistemas de software en un lenguaje común, es uno de los más conocidos y utilizados en la actualidad. Se utiliza para definir un sistema en cada una de las etapas por las que tiene que transitar, indica lo que supuestamente hará, no como lo hará. Incluye aspectos conceptuales

² WSDL: siglas de *Web Services Description Language*, un formato XML que se utiliza para describir servicios Web

tales como: procesos de negocio, funciones del sistema, y aspectos concretos como: expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Por lo que es muy útil para comprender el diseño del sistema. (34)

Permite modelar diferentes tipos de aplicación, que se ejecuten en cualquier tipo de combinación de hardware, sistema operativo, lenguaje de programación o infraestructura de red. Está diseñado para ser usado sobre los fundamentos de la orientación a objetos, convirtiéndolo en un marco ideal para los lenguajes orientados a objetos como C++, Java y C#, teniendo uso limitado para otros paradigmas de programación. Independientemente de la metodología que se utilice para llevar a cabo el análisis y el diseño, se puede utilizar UML para expresar los resultados, característica que lo ha llevado a ser ampliamente difundido. (35)

2.1.6. Herramienta de modelado: Visual Paradigm

Visual Paradigm es una herramienta ideada para soportar el ciclo de vida del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Resulta esencialmente útil durante el proceso de análisis y diseño por su probada utilidad para el analista.

Este sistema puede manejar gran parte de los diagramas definidos en el lenguaje UML, ya sea creándolos manualmente o importándolos a partir de código en C++, Java, Python, Pascal, Delphi, entre otros, lo cual ofrece posibilidades de ingeniería inversa. Permite crear un modelo y generar el código automáticamente en varios lenguajes para ayudar a comenzar la implementación del proyecto. El código generado incluye definiciones de clases con sus métodos y atributos proporcionando rapidez en el proceso inicial de desarrollo y haciéndolo menos propenso a errores. (36)

2.1.7. Servicios Web de Java 7

Los Servicios Web son un sistema de software diseñado para soportar la interoperabilidad máquina-máquina a través de una red. Este tiene una interfaz descrita en un formato que puede ser procesado por una máquina. Otros sistemas interactúan con el Servicio Web utilizando mensajes SOAP³.

Se puede decir que un Servicio Web es una comunicación por medio de mensajes SOAP entre diferentes equipos a través de una red.

SOAP es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los Servicios Web. (37)

³ SOAP: Simple Object Access Protocol.

2.1.8. Servidor Web Apache 2.2

Apache 2.2 es un servidor web de software libre cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan. Está diseñado para ser una solución de motor completo, servidor de FTP y portátiles basados en protocolos abiertos, pueden ejecutarse como servicio de Windows o de Unix / Linux, o incrustado en una aplicación Java. (38)

2.1.9. Entorno de Desarrollo Integrado

Un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) es una aplicación que consiste generalmente en la combinación de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede ser exclusivo para un solo lenguaje de programación o para varios.

NetBeans 7.3

NetBeans es un ambiente de desarrollo integrado libre, desarrollado principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender sus funcionalidades. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. El cual puede usarse, desde luego, no sólo para Java, sino para PHP, C/C++ y otros lenguajes.

NetBeans es un IDE o Entorno de Desarrollo Integrado, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Es de código abierto escrito completamente en Java usando la plataforma NetBeans. Permite crear aplicaciones web con PHP5 y Java, tiene un potente debugger y además se puede integrar con Drupal y AJAX. Es multiplataforma, funciona en Windows y Linux. (39)

Eclipse

Eclipse es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Cuenta con un editor de texto donde se puede ver el contenido del fichero en el que se trabaja, una lista de tareas, y otros módulos similares. Las características del programa se pueden ampliar y mejorar mediante el uso de plugins. (40)

Selección del IDE

Después de un estudio realizado de los IDEs expuestos anteriormente se llega a la siguiente conclusión:

Según las tecnologías que se utilizan y el entorno de la aplicación “Objetos de aprendizajes para apoyar la disciplina Inteligencia Artificial” se seleccionó NetBeans 7.3, el cual puede ser usado para desarrollar cualquier tipo de aplicación, también en la reutilización de módulos. Por otra parte posee soporte para PHP que es uno de los lenguajes de programación del cual se hace uso en la solución. También tiene las siguientes características (41):

- Herramientas Swing estándar.
- Mejor soporte para PHP.
- Permite importar proyectos de Eclipse y otros IDEs.

2.1.10. Sistema Gestor de Base de Datos

Un sistema gestor de base de datos tiene como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos.

MySQL 5.5.24

Es un sistema de gestión de bases de datos (SGBD) multiusuario, multiplataforma y de código abierto, bajo licencia GNU/GPL desarrollado por Sun Microsystems. Muy utilizado en las páginas web. MySQL es un Sistema de Administración de Base de Datos Relacionales rápido, robusto y fácil de usar. Se Adapta bien a la administración de datos en entornos de red, especialmente en arquitecturas cliente servidor.

Características de MySQL (42):

- MySQL está escrito en C y C++.
- Emplea el lenguaje SQL para consultas a la base de datos.
- MySQL Server está disponible como freeware bajo licencia GPL.
- MySQL Enterprise es la versión por suscripción para empresas, con soporte las 24 horas.
- Trabaja en las siguientes plataformas: AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Microsoft Windows (95, 98, ME, NT, 2000, XP y Vista).

2.1.11. Herramienta utilizada para las pruebas unitarias: JUnit 4.10

JUnit es un marco de trabajo para realizar y automatizar pruebas unitarias para el lenguaje de programación Java. Es decir, se sitúa en la fase de pruebas dentro del ciclo de Ingeniería del Software.

Algunas características de JUnit incluyen (43):

- Afirmaciones para probar resultados esperados.
- Aparatos de prueba para el compartir datos comunes de prueba.
- Corredores de pruebas para pruebas de ejecución.

JUnit está enlazado como un JAR en tiempo de compilación; el marco de trabajo reside bajo los paquetes: JUnit marco de trabajo, para JUnit 3.8 y anteriores y bajo org.junit para JUnit 4 en adelante.

2.2. Historias de usuario

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: constan de tres o cuatro líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. El tiempo de desarrollo ideal para una historia de usuario es entre una y tres semanas. (44)

Leyenda:

Número: número de identificación para las Historias de Usuario, sería incremental en el tiempo.

Nombre Historia de Usuario: es el nombre de la HU, sirve para identificarla fácilmente tanto para los desarrolladores como para los clientes.

Usuario: nombre del programador encargado de implementar la HU.

Prioridad del negocio: qué tan importante es para el cliente, se clasifica en Muy alta, Alta y Media.

Riesgo de desarrollo: qué tan difícil es para el desarrollador (Alto, Medio o Bajo).

Iteración asignada: iteración a que corresponde, número de la iteración en la que se desarrollará la HU.

Puntos estimados: tiempo en semanas que se le asignará. (Estimado)

Descripción: es la descripción de la historia, detallando las operaciones del usuario y las respuestas del sistema.

Capítulo 2: Planificación y Diseño

Observaciones: informaciones de interés, como glosarios o detalles del usuario.

Se detectaron un conjunto de historias de usuarios que responden a las funcionalidades que el sistema debe contener, así también, se redactaron un conjunto de tareas de ingeniería que constituyen los pasos a seguir para dar cumplimiento a las especificidades descritas por las historias de usuario. A continuación se muestran las historias de usuario detectadas.

Tabla 1 HU-1 Representación gráfica para el problema “El viajante”: Elaboración propia.

Historia de usuario	
Número: 1	Usuario: Persona ⁴
Nombre de historia: Representación gráfica para el problema “El viajante”	
Prioridad en el negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 1
Descripciones: La aplicación mostrará: <ul style="list-style-type: none">➤ El problema general de “El viajante”.➤ Un formulario donde el usuario pueda cambiar los datos del algoritmo.➤ Un ranking de soluciones.➤ Una representación de la solución del algoritmo, que muestra las evoluciones de los individuos.	
Observaciones:	

Tabla 2 HU-2: Representación gráfica para el problema “Mochila 01”: Elaboración propia.

Historia de usuario	
Número: 2	Usuario: Persona
Nombre de historia: Representación gráfica para el problema “Mochila 01”	
Prioridad en el negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripciones: La aplicación mostrará: <ul style="list-style-type: none">➤ El problema general de la “Mochila 01”.	

⁴Persona: Puede ser un profesor o un estudiante.

- Un formulario donde el usuario pueda cambiar los datos del algoritmo.
- Un ranking de soluciones.
- Una representación de la solución del algoritmo, que muestra las evoluciones de los individuos.

Observaciones:

2.3. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales para el éxito del mismo.

Software

Cliente

- Se utilizará cualquier navegador que sea compatible con HTML5 para acceder al sistema, se recomienda Google Chrome 26, Opera 12.10, Mozilla Firefox 20, Safari 6.0, Internet Explorer 10.
- Debe estar disponible para los sistemas operativos GNU/Linux y Windows.

Servidor

- Se utilizará un servidor de base de datos MySQL 5.5.24.
- Debe tener instalada la máquina virtual de java: Java Runtime Environment versión 7.0.17.
- Debe poseer el servidor web Apache 2.2.

Hardware

Para el correcto funcionamiento de la herramienta se debe disponer de un ordenador que tenga como mínimo las prestaciones siguientes:

Cliente

- Procesador: Intel(R) Pentium(R) 4 CPU 3.00GHz
- Memoria RAM: 128 MB.

Servidor

- Procesador: Intel(R) Pentium(R) 4 CPU 3.00GHz
- Memoria RAM: 256 MB.
- Capacidad de almacenamiento: 25 MB.

Apariencia o interfaz externa

- El sistema debe contar con una interfaz sencilla, fácil y cómoda que permita a los usuarios interactuar con el sistema, aún sin tener altos conocimientos del funcionamiento de este.
- La fuente de la letra deberá ser uniforme para todas las interfaces.
- Debe existir coherencia entre los íconos, los botones y la función que deben realizar.

Seguridad

- La aplicación debe contar con un control de permisos basado en roles, utilizando el sistema de configuración de permisos de usuarios y módulos que provee Drupal.
- Debe tener instalado el módulo LDAP (Lightweight Directory Access Protocol) Integration versión 7.x-1.0, para la administración de contraseñas (autenticación), y control de acceso (autorización).

2.4. Personas Relacionadas con el sistema

Las personas relacionadas con el sistema difieren únicamente en el nivel de permisos que cada uno tiene. A continuación se muestra una descripción detallada de cada uno:

Tabla 3 Personas relacionadas con el sistema: Elaboración propia.

Personas Relacionadas	Descripción
Estudiante	Podrá ver todo el contenido de la aplicación, realizar modificaciones manuales y ver los resultados con los nuevos datos ingresados.
Profesor	Podrá ver todo el contenido de la aplicación, realizar modificaciones manuales y ver los resultados con los nuevos datos ingresados. También podrá subir tareas para que los estudiantes le den soluciones.
Administrador	Tiene acceso total a la aplicación.

2.5. Estimación de esfuerzo por Historia de Usuario

La siguiente tabla contiene la estimación de esfuerzo por Historia de Usuario según el orden a realizar. Esta estimación incluye todo el esfuerzo asociado a la implementación de las HU, la misma se expresa utilizando como medida puntos los cuales son directamente proporcionales al esfuerzo que se realiza. Un punto se considera como una semana ideal de trabajo, donde se trabaje el tiempo planeado sin ningún tipo de interrupción.

Tabla 4 Estimación de esfuerzo por Historia de Usuario: Elaboración propia.

Historia de Usuario	Puntos Estimados
Representación gráfica para el problema “El viajante”	3
Representación gráfica para el problema “Mochila 01”	3

2.6. Plan de entregas

Este plan se obtiene a través de reuniones entre el equipo de trabajo y el cliente, para definir el marco temporal de la realización del sistema. Según la duración de cada iteración, así será la fecha aproximada de entrega. A continuación se muestra una tabla con las fechas aproximadas de entregas por cada iteración.

Tabla 5 Plan de entrega: Elaboración propia.

Herramienta	Final 1ra Iteración 3ra semana de abril	Final 2da Iteración 4ta semana de abril	Final 3ra Iteración 2da semana de mayo
Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial.	0.4	0.5	1.0

2.7. Propuesta del sistema

Se propone como solución la creación de la aplicación web “Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial”. Dicha aplicación estará desarrollada con el Sistema de Gestión de Contenido Drupal, los Objetos de Aprendizaje se construirán independientemente de la aplicación, para luego integrarse como un módulo de la misma. La aplicación tendrá como salida una visualización gráfica de las soluciones a los problemas: “El viajante” y “Mochila 01”, a través de algoritmos genéticos. Dichas soluciones se mostrarán de forma dinámica utilizando AJAX para las peticiones asíncronas y JavaScript para el manejo de los datos. La construcción de estos objetos de aprendizaje es compleja debido a que los algoritmos genéticos son los que más parámetros utilizan respecto a los demás Métodos de Solución de Problemas, como A*, Best First (Primero el Mejor) y Hill Climbing (Escalador de Colinas), el espacio de soluciones en

cada ejecución es muy amplio debido a que un ligero cambio en los parámetros puede cambiar completamente la solución y es necesario mantener un registro de cada ejecución de manera que permita la comparación de los resultados con diferentes parametrizaciones del algoritmo. Además la representación gráfica de los resultados es compleja, debido al amplio espacio de soluciones que se manejan.

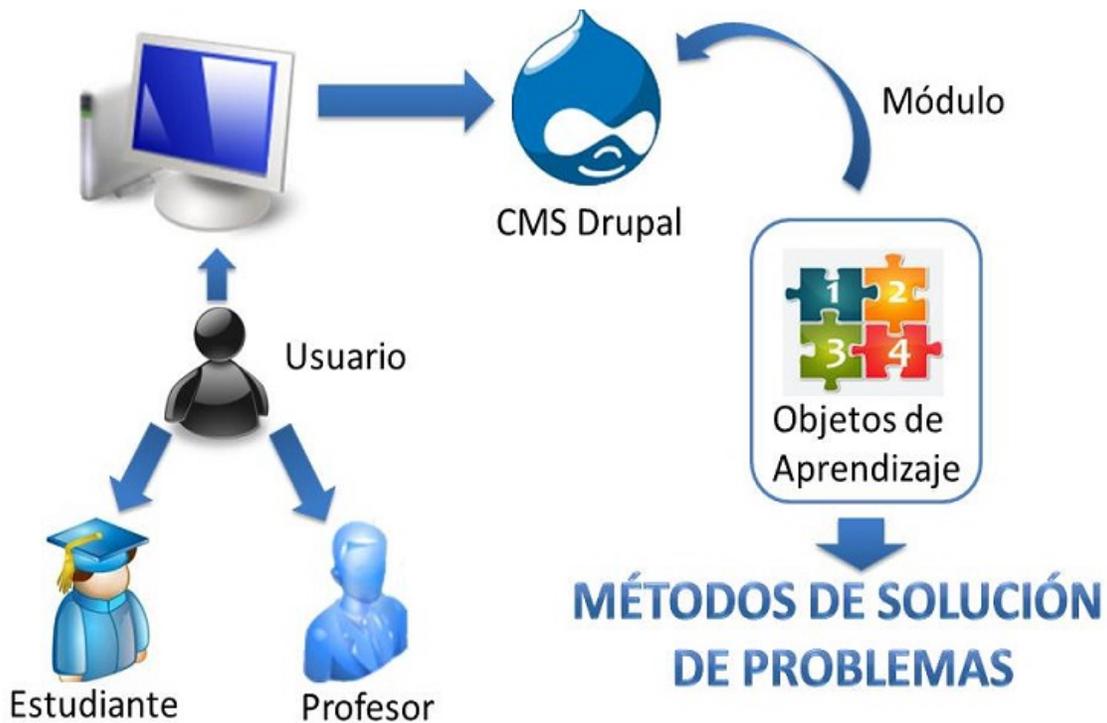


Figura 2 Propuesta del Sistema: Elaboración propia.

2.8. Patrón de Arquitectura

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. También expresan un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones.

Es necesario aclarar que para dar cumplimiento a la propuesta de solución se hizo necesario implementar los algoritmos genéticos como un servicio web de java aprovechando la extensibilidad que brinda a otros lenguajes y el uso de la biblioteca JGAP para el desarrollo de Algoritmos Genéticos, esto es debido a que PHP es un lenguaje interpretado del lado del servidor. Lo cual puede afectar el tiempo de ejecución de los algoritmos y por tanto no es el más adecuado para realizar esta función.

El patrón arquitectónico propuesto es el **Modelo-Vista-Controlador (MVC)**, el mismo separa en varios grupos las complejidades de las distintas partes que componen la aplicación, para aumentar la flexibilidad y reutilización, dichos grupos son, el modelo, encargado de separar los objetos con los datos, la vista, la cual dará una visualización de los resultados y el controlador, quien forma el enlace entre el modelo y la vista, también encargado de percibir el modo en que la interfaz reacciona ante la entrega al usuario, así como el acceso a la base de datos. Esto le permite cambiar el aspecto de la aplicación sin accidentalmente modificar su comportamiento. Además de desacoplar la vista del modelo, mediante un protocolo de suscripción/notificación. Cada vez que cambian los datos del modelo, avisará a las vistas que dependen de él y estas se actualizan.

Modelo: esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. (45)

Este modelo arquitectónico permite el acople de los servicios web de java brindando un especie de motor de algoritmos genéticos que recibe a partir de una solicitud los parámetros necesarios para la ejecución de un algoritmo determinado. Dicha solicitud es enviada a un controlador a partir de una acción del usuario que interactúa con la aplicación, este recibe los datos e invoca a un consumidor que es el encargado de comunicarse con el servicio web solicitado, y proveerle la información necesaria mediante una petición soap para que se ejecute y devuelva una respuesta que luego es procesada y enviada a la vista la cual se encarga de entregarle los datos al usuario que realizó la solicitud de forma dinámica en una interfaz visual. En la Figura 3 Modelo Arquitectónico propuesto se ilustra el proceso descrito.

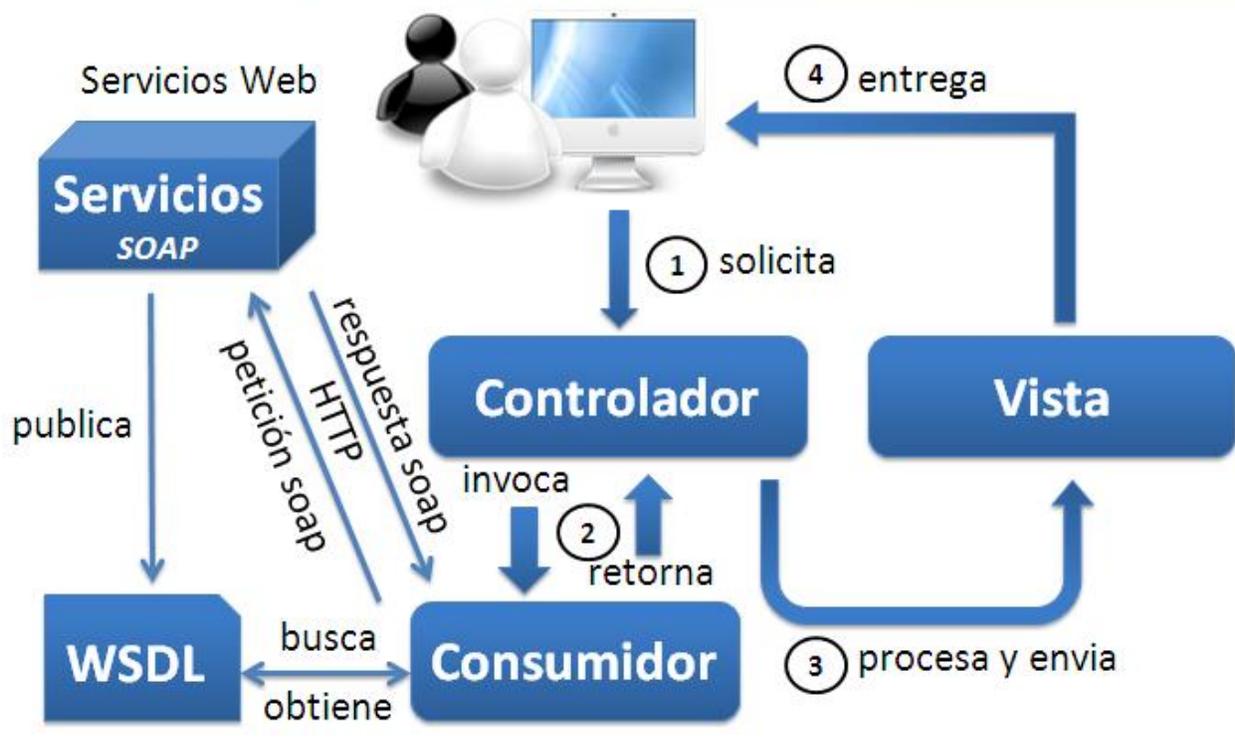


Figura 3 Modelo Arquitectónico propuesto: Elaboración propia.

2.9. Patrones de Diseño

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema, la cual puede ser utilizada en diversas situaciones. Los patrones de diseño reflejan todo el rediseño y remodificación que los desarrolladores han ido haciendo a medida que intentaban conseguir mayor reutilización y flexibilidad en su software.

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema concreto, describiendo cuándo aplicarlo y si tiene sentido hacerlo teniendo en cuenta otras restricciones de diseño, así como las consecuencias, ventajas e inconvenientes de su uso. (46)

El diseño de todas las tarjetas de Clase-Responsabilidad-Colaboradores, hacen uso de las buenas prácticas propuestas por Craig Larman en el libro "UML y patrones" (47), referente a la asignación de responsabilidades (GRASP), que constituyen un elemento primordial cuando se diseña e implementa un sistema orientado a objetos. El patrón Experto se utiliza para asignar la

Capítulo 2: Planificación y Diseño

responsabilidad que tendrá cada clase, en dependencia de la información que maneja. El patrón Creador se utiliza para asignar a determinadas clases la responsabilidad de crear instancias de otras, si cumple con determinadas condiciones. Por otra parte, se tuvo en cuenta asignar responsabilidades manteniendo un bajo acoplamiento y una alta cohesión entre cada clase que compone el sistema.

En el siguiente diagrama de clases se representan los patrones de diseño utilizados en el desarrollo de la aplicación.

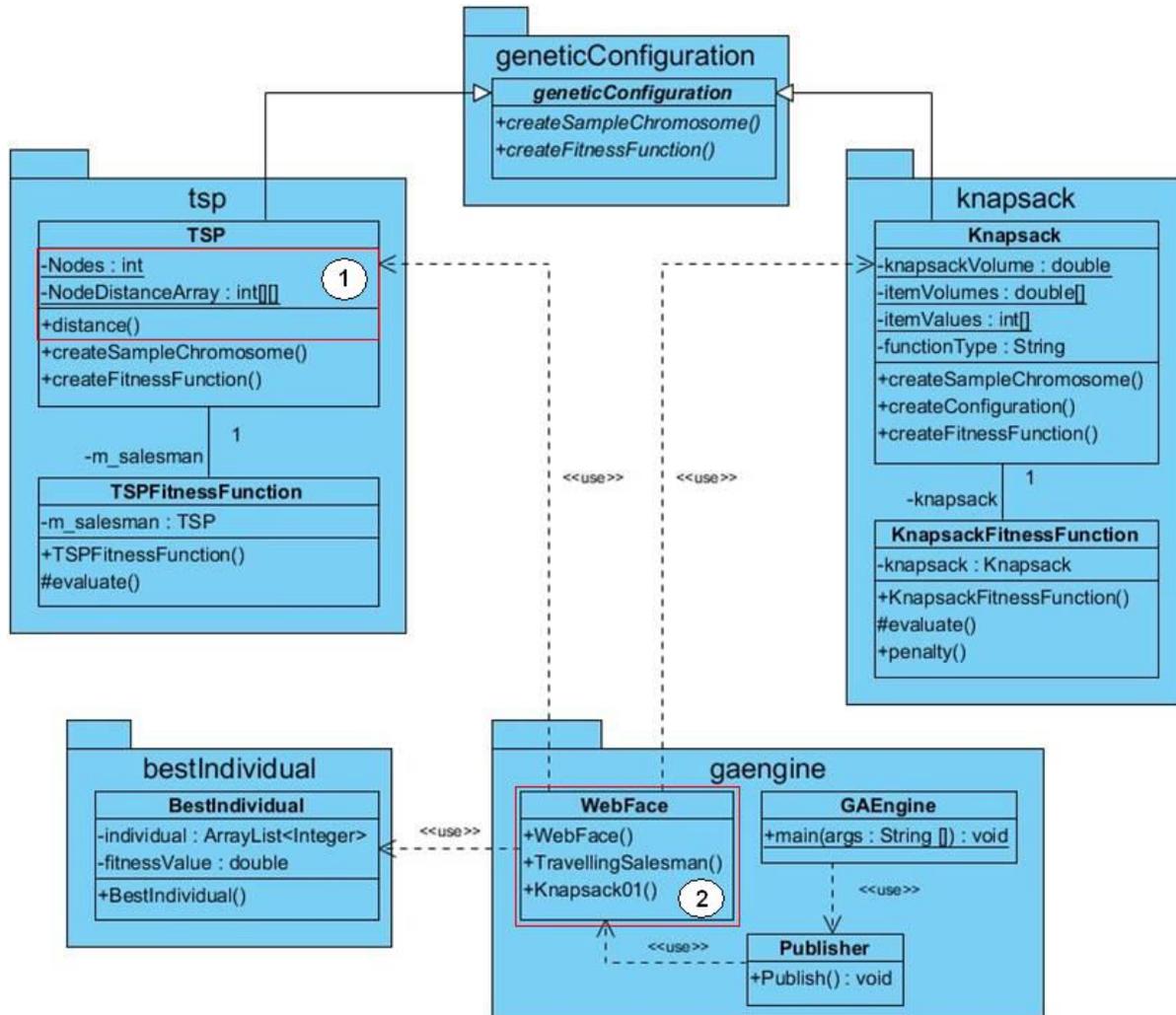


Figura 4 Diagrama de clases para la representación de los patrones: Elaboración propia.

- Experto: indica, por ejemplo, que la responsabilidad de la implementación del método **distance**, debe recaer sobre la clase TSP que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con alta cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). Los objetos utilizan su propia

información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

- Creador: ayuda a identificar, por ejemplo, que la clase WebFace encargada de implementar los algoritmos genéticos es responsable de la creación (o instanciación) de objetos de las clases TSP y Knapsack. Debido a que usa directamente las instancias creadas del objeto. Su utilización permite visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización.

2.10. Tarjetas Clase-Responsabilidades-Colaboradores (CRC)

Las Tarjetas CRC (Class, Responsibilities and Collaboration) sirven para diseñar el sistema en conjunto entre todo el equipo, permiten reducir el modo de pensar procedural y apreciar la tecnología de objetos.

Las siguientes tarjetas CRC describen las clases que conforman el sistema:

Tabla 6 CRC 1.

Clase: WebFace
Responsabilidad: en esta clase estarán implementados los algoritmos genéticos como servicios web.
Colaboradores: Salesman, Knapsack, GeneticConfiguration, BestIndividual, jgap

Tabla 7 CRC 2.

Clase: Publisher
Responsabilidad: publica el servicio web.
Colaboradores: Endpoint, WebFace

Tabla 8 CRC 3.

Clase: GAEngine
Responsabilidad: encargada de iniciar el servicio web.
Colaboradores: Publisher

Tabla 9 CRC 4.

Clase: BestIndividual
Responsabilidad: modela el resultado de los algoritmos genéticos.
Colaboradores: no tiene

Tabla 10 CRC 5.

Capítulo 2: Planificación y Diseño

Clase: TSP
Responsabilidad: modela el problema de “El viajante”.
Colaboradores: jgap, geneticConfiguration

Tabla 11 CRC 6.

Clase: TSPFitnessFunction
Responsabilidad: calcula la Función Objetivo (FO) del algoritmo genético específicamente para el problema “El viajante”.
Colaboradores: jgap

Tabla 12 CRC 7.

Clase: Knapsack
Responsabilidad: modela el problema “Mochila 01”.
Colaboradores: geneticConfiguration

Tabla 13 CRC 8.

Clase: KnapsackFitnessFunction
Responsabilidad: calcula la Función Objetivo (FO) del algoritmo genético específicamente para el problema “Mochila 01”.
Colaboradores: jgap

Tabla 14 CRC 9.

Clase: geneticConfiguration
Responsabilidad: es una clase abstracta que contiene los métodos comunes de las clases TSP y Knapsack
Colaboradores: jgap

Para un mejor entendimiento de cómo están relacionadas estas clases y las funcionalidades que realizan cada una de ellas, se elaboró el siguiente diagrama de clases.

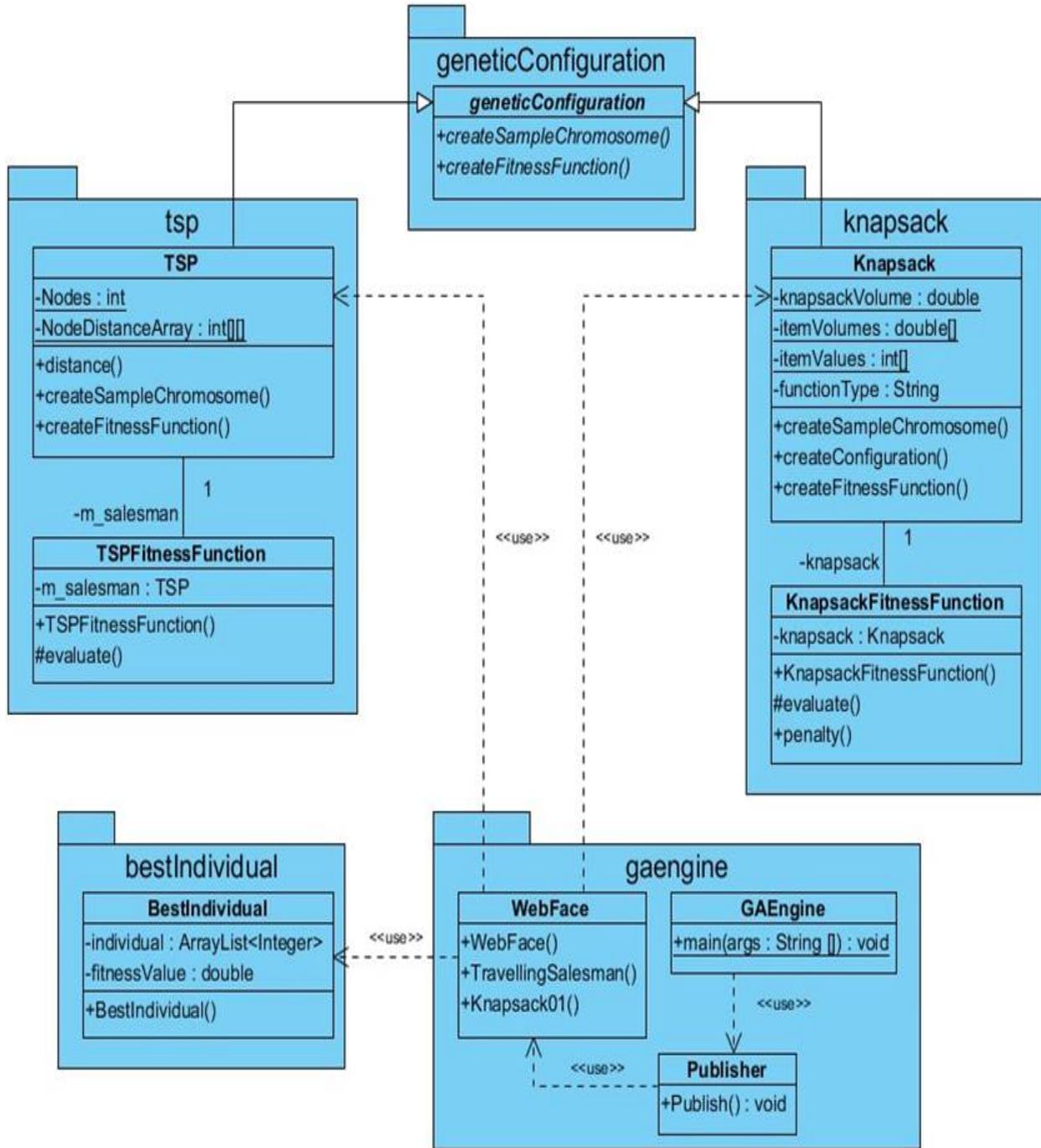


Figura 5 Diagrama de Clases: Elaboración propia.

2.11. Conclusiones Parciales

Se seleccionaron las herramientas, lenguajes y bibliotecas a utilizar para el desarrollo de la solución. Concluyendo que como herramienta CASE⁵ se hará uso de Visual Paradigm y del lenguaje de modelado UML con el objetivo de representar los diagramas necesarios durante el ciclo de vida del proyecto. Los lenguajes de programación de los cuales se hará uso son: PHP y Java bajo el entorno de desarrollo NetBeans. De igual modo, para la lógica del cliente y el control de los procedimientos, se selecciona JavaScript con el objetivo de definir los elementos de la interfaz de usuario. En el caso de las bibliotecas a utilizar se optó por: JGAP 3.6.2 para la implementación de los algoritmos genéticos; para el consumo de los servicios web se seleccionó Nusoap 0.9.5; además de JQuery 1.7.2 para el desarrollo visual de la aplicación, haciendo uso de la técnica de programación AJAX para lograr una interacción dinámica entre usuario-aplicación. También se realizó una captura de requisitos, documentándose, dos historias de usuario que responden a las necesidades del cliente. Por otra parte se realizó una estimación del esfuerzo por historia de usuario y el plan de entregas para el proyecto. Dentro de las actividades referentes al diseño, contenidas por la metodología de desarrollo XP a partir de la solución propuesta, se seleccionó el patrón arquitectónico MVC. De igual modo se definieron los patrones de diseño GRASP, lo que permitió crear la línea base de la arquitectura y una mejor organización y calidad de la solución, evidenciada a través de las tarjetas CRC.

⁵ Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

Capítulo 3: Implementación y Prueba

Introducción

En este capítulo se desarrollan las actividades del marco de trabajo propuestas por la metodología XP, Implementación y Pruebas. Por cada iteración se realiza una planificación para darle cumplimiento a la historia de usuario seleccionada, así como el conjunto de tareas que se ejecutan para su terminación. Finalmente cada iteración concluye con un conjunto de pruebas de aceptación que permiten validar el cumplimiento de los requisitos.

3.1. Planeación

3.1.1. Plan de lanzamiento

Al finalizar la primera fase de exploración ya se tiene una idea básica de cómo debe funcionar el sistema.

Se analizan las historias de usuario generadas gracias a la participación continua del cliente en el proceso de construcción, en vista que son dos historias de usuario complejas se realiza una selección de la que se desarrollaría primero. Para el cliente la historia más sensible y la que requiere mayor prioridad sería la historia de usuario número uno. Por lo que se comienza el desarrollo con la historia de usuario número uno.

3.2. Iteraciones

A continuación se muestra el seguimiento del proyecto al pasar por cada una de las iteraciones definidas por el usuario en el plan de entregas.

3.2.1. Primera iteración

3.2.1.1. Planificación de Iteración

Implementación de las tareas de ingeniería correspondientes a la historia de usuario número uno. Durante el transcurso de la misma se crea la base de la arquitectura del sistema. Como principal resultado se obtienen algunos de los requerimientos críticos especificados por el cliente.

Descripción de las tareas de ingeniería abordadas en la iteración

Tabla 15 Tarea de Ingeniería No1. Implementación de la interfaz principal: Elaboración propia.

Tarea	
Número tarea:1	Número historia:
Nombre tarea: Implementación de la interfaz principal.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: José Dalién Roque Pérez	

Capítulo 3: Implementación y Prueba

Descripción:

Se implementan las funcionalidades necesarias para confeccionar la interfaz principal.

- Autenticar usuario.
- Buscador de elementos.
- Mostrar usuarios conectados.
- Mostrar usuarios nuevos.
- Mostrar artículos creados.
- Mostrar temas del foro.
- Mostrar Objetos de Aprendizaje creados.

Tabla 16 HU-1: Tarea de Ingeniería No2. Implementación del problema “El viajante” por Algoritmos Genéticos: Elaboración propia.

Tarea	
Número tarea: 2	Número historia:1
Nombre tarea: Implementación del problema “El viajante” por Algoritmos Genéticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: José Dalién Roque Pérez	
Descripción:	
Dado un conjunto de parámetros se debe resolver el problema “El viajante” brindando una solución a través de la implementación del algoritmo genético.	

Tabla 17 Tarea de Ingeniería No3. Vinculación de los elementos gráficos a la solución del problema “El viajante” por Algoritmos Genéticos: Elaboración propia.

Tarea	
Número tarea: 3	Número historia:1
Nombre tarea: Vinculación de los elementos gráficos a la solución del problema “El viajante” por Algoritmos Genéticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: José Dalién Roque Pérez	
Descripción:	
En la aplicación existe un formulario que deberá ser llenado por el usuario, si se presiona la opción “Ejecutar” en el formulario, este envía los datos a una clase controladora que es	

la encargada de ejecutar el algoritmo, los parámetros del algoritmo son: Operador de selección (string), Operador de cruce (string), Operador de mutación (string), Población inicial (int) y Evoluciones (int). Algunos de los parámetros tienen opciones las cuales pueden ser seleccionadas por el usuario :

1. El Operador de selección contiene las opciones: Ruleta, Torneo y Mejor individuo, si escoge Torneo deberá insertar Tamaño (int) y Probabilidad (double). En caso de que escoja Mejor individuo deberá insertar Probabilidad (double).

2. El Operador de cruce contiene la opción: Voraz.

3. El Operador de mutación contiene las opciones: Intercambio e Intercambio con rango, si escoge Intercambio deberá insertar Tasa de mutación (int). En caso de que escoja Intercambio con rango deberá insertar Tasa de mutación (int) y Rango (int).

3.2.1.2. Desarrollo

A continuación se muestran los prototipos de interfaz obtenidos una vez concluida la iteración.

En primer lugar se desarrolla la tarea de ingeniería número uno la cual es “Implementación de la interfaz principal”:

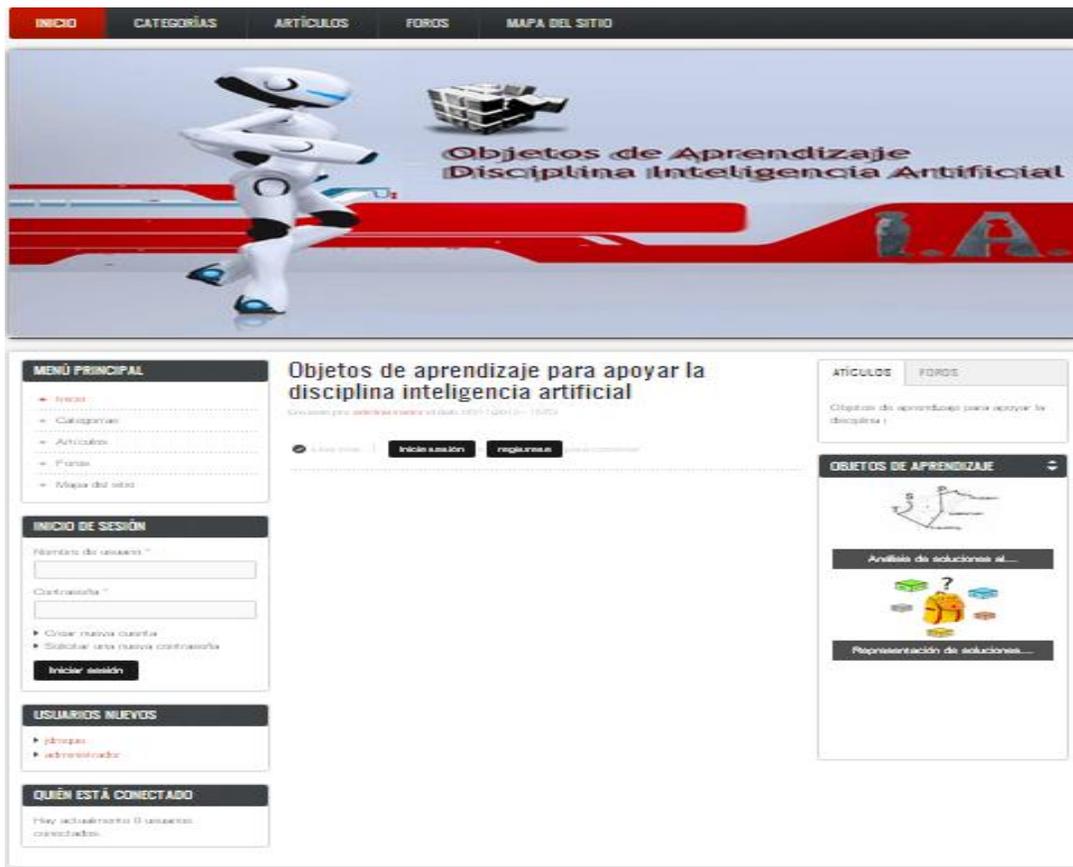


Figura 6 Pantalla de ingreso al sistema: Elaboración propia.

En el transcurso de esta iteración surgen grandes expectativas sobre la aplicación y su alcance. El cliente se muestra motivado y satisfecho por el trabajo desarrollado hasta el momento.

3.2.1.3. Pruebas de Aceptación

Historia de usuario 1

Tabla 18 Ejecutar algoritmo genético para el problema El viajante: Elaboración propia.

Caso de prueba de aceptación 1
Historia de usuario: Representación gráfica para el problema “El viajante”.
Nombre: Ejecutar algoritmo genético para el problema El viajante.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que se ejecute correctamente el algoritmo.
Condiciones de ejecución: debe estar ejecutado el servicio web de algoritmos genéticos.
Entradas/pasos de ejecución: <ul style="list-style-type: none">➤ Seleccionar del menú la opción “Objetos de Aprendizaje”.➤ Seleccionar el OA “Análisis de soluciones al problema El viajante mediante algoritmos genéticos”.
Resultado esperado: los resultados del algoritmo genético deben aparecer en la secciones “Ejecuciones del algoritmo” y “Mejores resultados”.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 19 Mostrar los resultados del algoritmo genético para el problema “El viajante”: Elaboración propia.

Caso de prueba de aceptación 2
Historia de usuario: Representación gráfica para el problema “El viajante”.
Nombre: Mostrar los resultados del algoritmo genético para el problema “El viajante”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que el objeto de aprendizaje cuenta con una representación gráfica de las soluciones.

Capítulo 3: Implementación y Prueba

Condiciones de ejecución: debe seleccionar una de las soluciones.
Entradas/pasos de ejecución: <ul style="list-style-type: none">➤ Seleccionar en la sección Ejecuciones del algoritmo la opción “Mostrar”.
Resultado esperado: representación gráfica de las soluciones al problema “El viajante”.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 20 Mostrar paginado de los resultados del algoritmo genético para el problema “El viajante”:
Elaboración propia.

Caso de prueba de aceptación 3
Historia de usuario: Representación gráfica para el problema “El viajante”.
Nombre: Mostrar paginado de los resultados del algoritmo genético para el problema “El viajante”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que se muestre todas las ejecuciones del algoritmo mediante paginado.
Condiciones de ejecución: el algoritmo debe haberse ejecutado más de una vez.
Entradas/pasos de ejecución: <ul style="list-style-type: none">➤ Seleccionar en la sección Ejecuciones del algoritmo las opciones “Anterior” o “Siguiente”.
Resultado esperado: debe mostrarse la Siguiente o Anterior ejecución del algoritmo.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 21 Mostrar los mejores resultados del algoritmo genético para el problema “El viajante”:
Elaboración propia.

Caso de prueba de aceptación 4
Historia de usuario: Representación gráfica para el problema “El viajante”.
Nombre: Mostrar los mejores resultados del algoritmo genético para el problema “El viajante”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de que se muestre una representación gráfica de

Capítulo 3: Implementación y Prueba

los mejores resultados del algoritmo.
Condiciones de ejecución: debe seleccionar una de las soluciones.
Entradas/pasos de ejecución: <ul style="list-style-type: none">➤ Seleccionar en la secciones Ejecuciones del algoritmo o Mejores resultados la opción “Mostrar”.➤ Seleccionar la opción “Mejores resultados”.
Resultado esperado: debe mostrarse una representación gráfica de los mejores resultados del algoritmo.
Evaluación de la prueba: Prueba satisfactoria.

Al término del desarrollo de la historia de usuario se realizan pruebas para determinar posibles fallas o errores en la forma en que trabajaba el sistema. Las no conformidades detectadas se detallan a continuación:

- El color de los mensajes de error o de advertencia no están en contraste con los de la aplicación.
- La estructura de la información de los Objetos de Aprendizaje no es uniforme.
- La creatividad de las animaciones de los Objetos de Aprendizaje, es pobre.

La mayoría de los problemas antes mencionados son producto de la falta de experiencia en el manejo de las herramientas.

3.2.2. Segunda iteración

3.2.2.1. Planificación de iteración

La presente iteración se realiza con el objetivo de mitigar las no conformidades detectadas en la anterior iteración.

3.2.2.2. Desarrollo

Se detallan los cambios a continuación:

- Todos los mensajes se diseñaron en contraste con los colores de la aplicación.
- El objeto de aprendizaje se estructuró de forma uniforme y coherente.
- Las animaciones se mejoraron utilizando JavaScript, CSS, HTML5 y Canvas.

Luego se desarrollan las tareas de ingeniería número uno y número dos las cuales son: Implementación del problema “El viajante” por Algoritmos Genéticos y Vinculación de los elementos gráficos a la solución del problema “El viajante” por Algoritmos Genéticos.

A continuación se muestran los prototipos de interfaz obtenidos una vez concluida la iteración.

Capítulo 3: Implementación y Prueba

The image shows a software interface for configuring and executing an algorithm. It is divided into two main panels: 'Configurar algoritmo' and 'Ejecuciones del algoritmo'.

Configurar algoritmo:

- * Operador de selección:** Ruleta
- * Operador de cruzamiento:** Voraz
- * Operador de mutación:** Intercambio
- * Tasa de mutación:** 1000
- * Población inicial:** 100
- * Evoluciones:** 50

Buttons: Ejecutar, Cancelar

Ejecuciones del algoritmo:

Página 1 de 1

Parámetros del algoritmo=> Operador selección: Ruleta, Operador cruce: Voraz, Operador mutación: Intercambio, Tasa de mutación: 1000, Población inicial: 100, Evoluciones: 50.

Generación: 0

Ruta: [Parque de la Fraternidad => Museo de la Revolución => Floridita => Museo Nacional de Bellas Artes => Catedral

Mejores resultados:

Ejecución: 1

Ruta: [Parque de la Fraternidad => Floridita => Museo de la Revolución => Museo Nacional de Bellas Artes => Catedral de la Habana Vieja => Museo del Ron => Museo Casa Natal José Martí => Parque de la Fraternidad].

Distancia total recorrida: 4840.0 m.

Figura 7 Formulario del problema "El viajante": Elaboración propia.

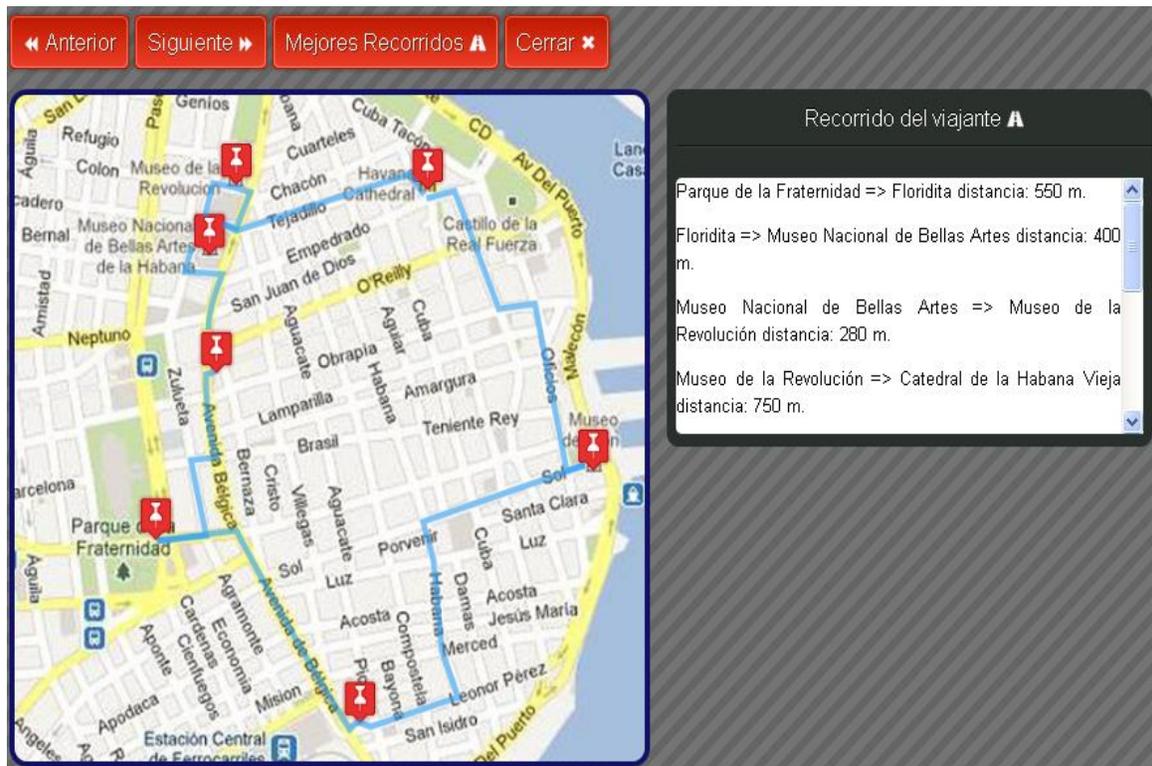


Figura 8 Visualización al problema "El viajante": Elaboración propia.

3.2.2.3. Pruebas de aceptación

En esta ocasión el cliente realiza una exhaustiva revisión a los cambios efectuados. Este manifiesta su conformidad con lo desarrollado, por lo que se da por finalizada la historia de usuario número uno.

3.2.3. Tercera iteración

3.2.3.1. Planificación de iteración

Implementación de las tareas de ingeniería correspondientes a la historia de usuario número dos.

Descripción de las tareas de ingeniería abordadas en la iteración

Tabla 22 HU-2: Tarea de Ingeniería No1. Implementación del problema “Mochila 01” por Algoritmos Genéticos: Elaboración propia.

Tarea	
Número tarea: 1	Número historia: 2
Nombre tarea: Implementación del problema “Mochila 01” por Algoritmos Genéticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: José Dalién Roque Pérez	
Descripción: Dado un conjunto de parámetros se debe resolver el problema “Mochila 01” brindando una solución a través de la implementación del algoritmo genético.	

Tabla 23 HU-2: Tarea de Ingeniería No2. Vinculación de los elementos gráficos a la solución del problema “Mochila 01” por Algoritmos Genéticos: Elaboración propia.

Tarea	
Número tarea: 2	Número historia: 2
Nombre tarea: Vinculación de los elementos gráficos a la solución del problema “Mochila 01” por Algoritmos Genéticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador responsable: José Dalién Roque Pérez	
Descripción: En la aplicación existe un formulario que deberá ser llenado por el usuario, si se presiona la opción “Ejecutar” en el formulario, este envía los datos a una clase controladora que es la encargada de ejecutar el algoritmo, los parámetros del algoritmo son: Operador de selección	

(string), Operador de cruce (string), Operador de mutación (string), Población inicial (int), Evoluciones (int) y Función objetivo (boolean). Algunos de los parámetros tienen opciones las cuales pueden ser seleccionadas por el usuario :

1. El Operador de selección contiene las opciones: Ruleta, Torneo y Mejor individuo, si escoge Torneo deberá insertar Tamaño (int) y Probabilidad (double). En caso de que escoja Mejor individuo deberá insertar Probabilidad (double).
2. El Operador de cruce contiene las opciones: Por punto y Promediado, si escoge Por punto deberá insertar Tasa de cruzamiento (int).
3. El Operador de mutación contiene las opciones: Simple, Gaussiana, Intercambio e Intercambio con rango, si escoge Simple deberá insertar Tasa de cruzamiento (int). En caso de que escoja Intercambio deberá insertar Tasa de cruzamiento (int), si escoge Intercambio con rango deberá insertar Tasa de cruzamiento (int) y Rango (int). En caso de que escoja Gaussiana deberá insertar Desviación (double).

3.2.3.2. Desarrollo

Se desarrolla la historia de usuario número 2. Con las tareas de ingeniería número uno y número dos las cuales son: Implementación del problema “Mochila 01” por Algoritmos Genéticos y Vinculación de los elementos gráficos a la solución del problema ‘Mochila 01’ por Algoritmos Genéticos.

Capítulo 3: Implementación y Prueba

A continuación se muestran los prototipos de interfaz obtenidos una vez concluida la iteración.



Figura 9 Formulario del problema "Mochila 01": Elaboración propia.

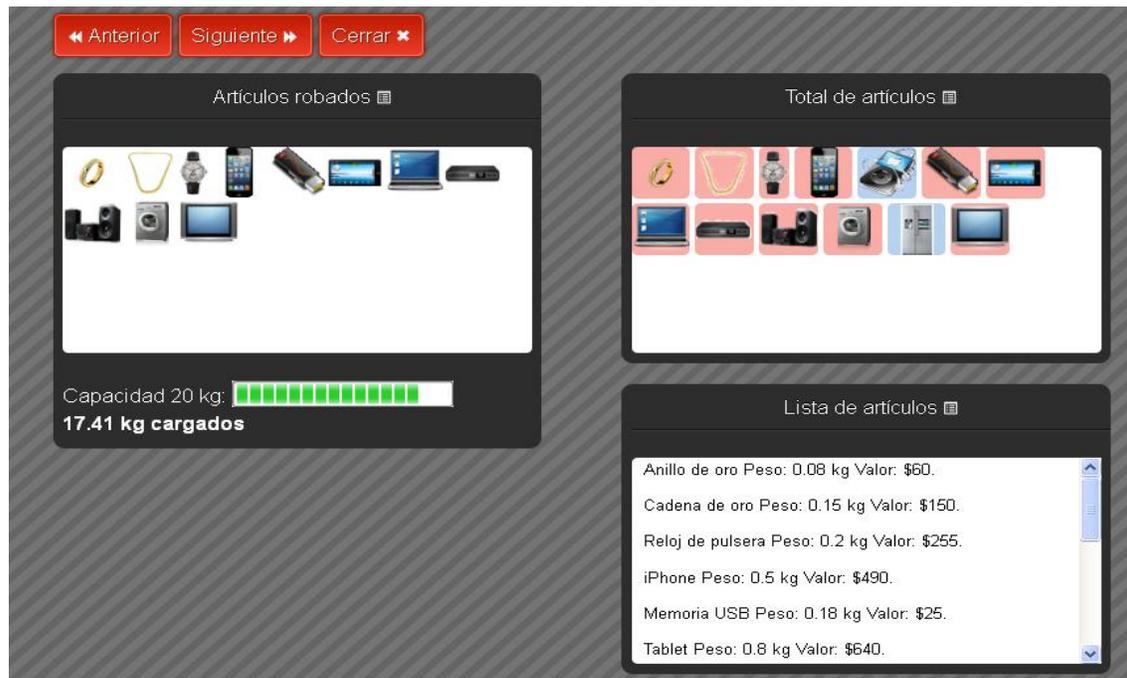


Figura 10 Visualización del problema "Mochila 01": Elaboración propia.

3.2.3.3. Pruebas de aceptación

Historia de usuario 2

Tabla 24 Ejecutar algoritmo genético para el problema “Mochila 01”: Elaboración propia.

Caso de prueba de aceptación 5
Historia de usuario: Representación gráfica para el problema “Mochila 01”.
Nombre: Ejecutar algoritmo genético para el problema “Mochila 01”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que se ejecute correctamente el algoritmo.
Condiciones de ejecución: debe estar ejecutado el servicio web de algoritmos genéticos.
Entradas/pasos de ejecución: <ul style="list-style-type: none">➤ Seleccionar del menú la opción “Objetos de Aprendizaje”.➤ Seleccionar el OA “Análisis de soluciones al problema “Mochila 01” mediante algoritmos genéticos”.
Resultado esperado: los resultados del algoritmo genético deben aparecer en la secciones “Ejecuciones del algoritmo” y “Mejores resultados”.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 25 Mostrar los resultados del algoritmo genético para el problema “Mochila 01”: Elaboración propia.

Caso de prueba de aceptación 6
Historia de usuario: Representación gráfica para el problema “Mochila 01”.
Nombre: Mostrar los resultados del algoritmo genético para el problema “Mochila 01”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que el objeto de aprendizaje cuenta con una representación gráfica de las soluciones.
Condiciones de ejecución: debe seleccionar una de las soluciones.
Entradas/pasos de ejecución:

Capítulo 3: Implementación y Prueba

➤ Seleccionar en la sección Ejecuciones del algoritmo la opción “Mostrar”.
Resultado esperado: representación gráfica de las soluciones al problema “Mochila 01”.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 26 *Mostrar paginado de los resultados del algoritmo genético para el problema “Mochila 01”:*
Elaboración propia.

Caso de prueba de aceptación 7
Historia de usuario: Representación gráfica para el problema “Mochila 01”.
Nombre: Mostrar paginado de los resultados del algoritmo genético para el problema “Mochila 01”.
Responsable: José Dalién Roque
Descripción: esta prueba se realiza con el objetivo de comprobar que se muestre todas las ejecuciones del algoritmo mediante paginado.
Condiciones de ejecución: el algoritmo debe haberse ejecutado más de una vez.
Entradas/pasos de ejecución: ➤ Seleccionar en la sección Ejecuciones del algoritmo las opciones “Anterior” o “Siguiente”.
Resultado esperado: debe mostrarse la Siguiente o Anterior ejecución del algoritmo.
Evaluación de la prueba: Prueba satisfactoria.

El probador realiza las pruebas de caja negra a las funciones agregadas las que cumplen sin errores en el 100% de los casos probados. Con esto se da por finalizada la historia de usuario número 2, así como también el desarrollo de la aplicación. Se da por finalizada la fase de iteración para pasar a la fase de producción.

3.3. Resumen del resultado de las iteraciones

Cada iteración ejecutada concluyó con la realización de un conjunto de pruebas de aceptación para verificar el cumplimiento de los requisitos. Dichas pruebas permitieron al cliente comprobar el grado de satisfacción con el resultado. Las no conformidades detectadas en cada iteración fueron mitigadas antes de dar paso a una nueva iteración.

El gráfico que se muestra a continuación resume los resultados arrojados por cada iteración:

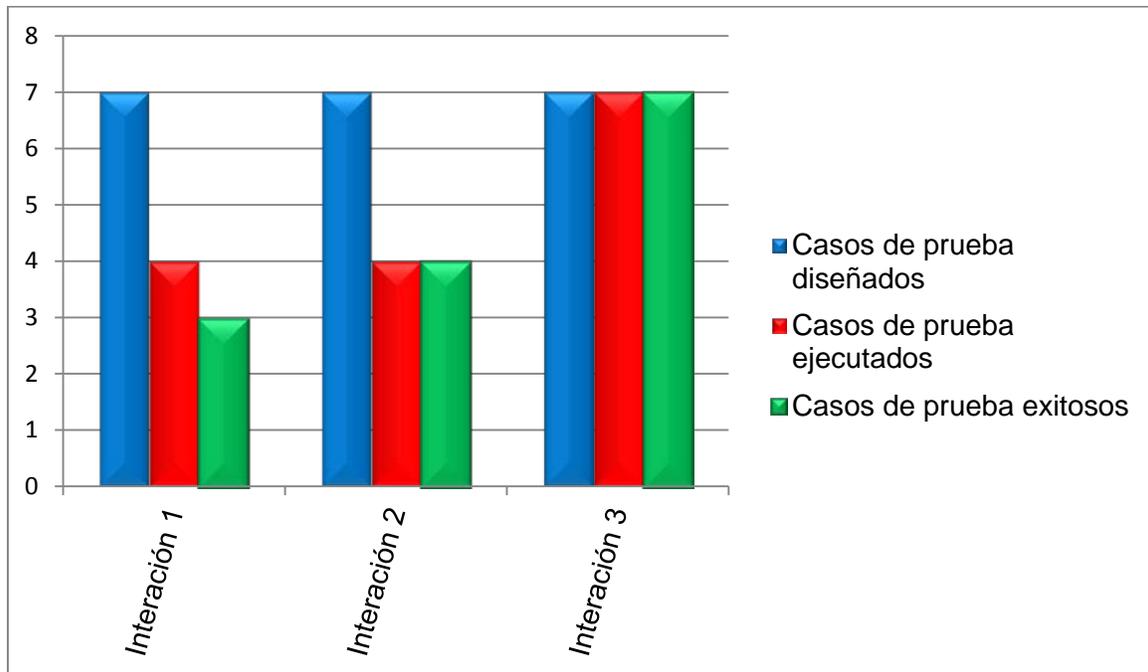


Figura 11 Resumen de los resultados de las pruebas realizadas: Elaboración propia.

3.4. Pruebas al sistema

Las pruebas permiten determinar el estado de la calidad del producto de software. La metodología XP propone la realización de dos tipos de pruebas al sistema, que son las llamadas pruebas unitarias y pruebas de aceptación. El desarrollo de software mediante dicha metodología está compuesto por una serie de iteraciones cortas, cada iteración concluye ejecutando un conjunto de pruebas de aceptación, que permitan al cliente comprobar si está satisfecho con el resultado. En otras palabras, las historias de usuario detectadas tendrán asociadas tantas pruebas de aceptación como sean convenientes para verificar el grado de cumplimiento de cada una. Las pruebas unitarias son definidas antes de realizarse la implementación del código

Pruebas unitarias: enfocada a los elementos probables más pequeños del software. Aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca. Todo el código debe tener pruebas unitarias y debe pasarlas antes de ser liberado.

A continuación un ejemplo de pruebas unitarias que se le realizaron a la aplicación, este es un ejemplo específico a una de las clases de la aplicación, la clase WebFace.

Capítulo 3: Implementación y Prueba

Figura 12 Ejemplo de prueba unitaria con error: Elaboración propia.

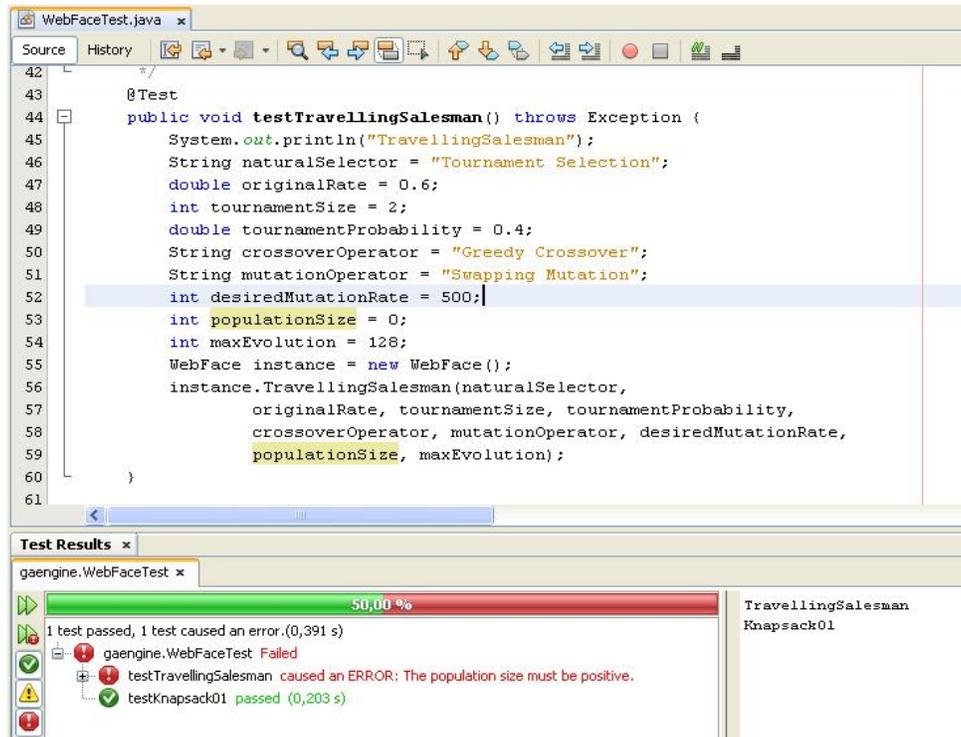
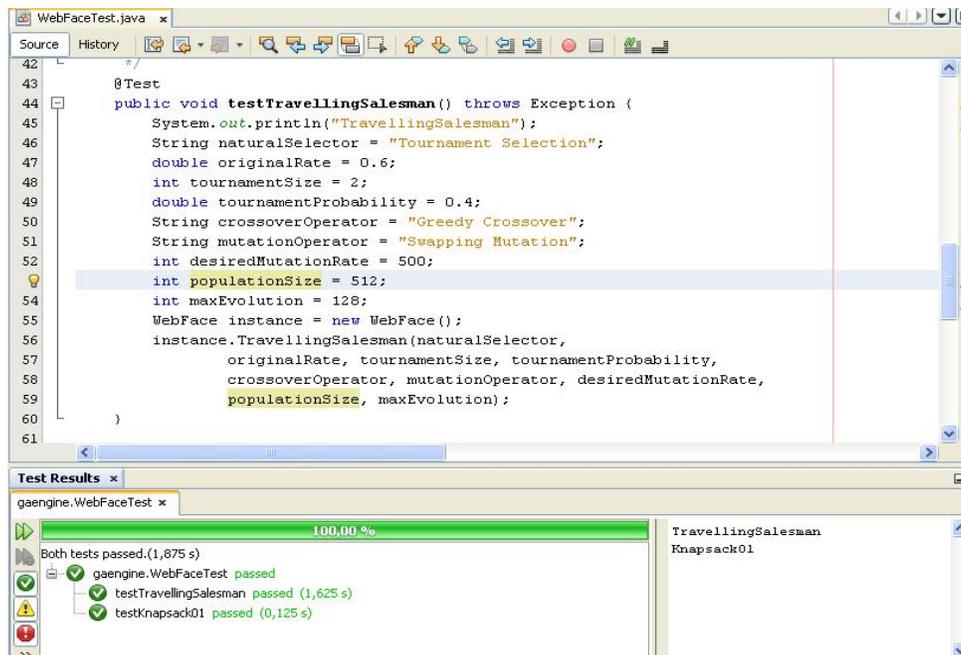


Figura 13 Ejemplo de prueba unitaria sin error: Elaboración propia.



Pruebas de aceptación: con frecuencia se realizan pruebas de aceptación. Estas pruebas son generadas a partir de las historias de usuario elegidas para cada iteración. Son pruebas en donde el cliente verifica que lo que se está probando funcione correctamente. Cuando se pasa la prueba de aceptación se considera la historia de usuario finalizada.

3.5. Caso de estudio del Objeto de Aprendizaje Análisis de las soluciones del problema Viajante mediante algoritmos genéticos.

En este caso de estudio se describen las funcionalidades más importantes del OA, con el objetivo de ilustrar las ventajas de la aplicación de los OA como apoyo al Proceso de Enseñanza-Aprendizaje en la disciplina Inteligencia Artificial.

Principales funcionalidades

Parametrización del algoritmo genético: permite al usuario (estudiante o profesor) obtener diferentes soluciones en dependencia de la configuración de los operadores genéticos que brinda el OA, mediante esta operación se pueden identificar que parámetros influyen positiva o negativamente en la solución, brindando la posibilidad de experimentar distintos escenarios de un mismo problema.

Paginación de los resultados del algoritmo genético: permite al usuario (estudiante o profesor) mantener un registro organizado de las ejecuciones del algoritmo, brindando la posibilidad de comparar soluciones, en función de identificar la posible solución más óptima y los parámetros utilizados para su obtención.

Visualización de los resultados del algoritmo genético: permite al usuario (estudiante o profesor) observar una representación animada de una posible solución al problema, brindando la posibilidad de identificar la aplicación de un algoritmo genético para dar solución a un problema computacional clásico recreado en un ejemplo cotidiano.

Partiendo de las funcionalidades descritas y los beneficios que aportan a estudiantes y profesores se puede corroborar que el desarrollo de la aplicación y el uso de los objetos de aprendizaje elaborados apoyan el Proceso de Enseñanza-Aprendizaje en la disciplina Inteligencia Artificial en correspondencia con el programa analítico de la asignatura IA I en la UCI, fortaleciendo numerosas habilidades en los estudiantes:

Dentro de los Objetivos Generales:

Objetivos Educativos:

- Desarrolla el espíritu de autosuperación que permite mantener actualizado los avances de los temas de la inteligencia artificial.
- Ayuda a aplicar capacidades cognoscitivas, de razonamiento y de las formas del pensamiento lógico, hábitos de lectura y trabajo independiente.

Objetivos Instructivos:

- Caracteriza a los algoritmos genéticos como MSP y determina su idoneidad en la

solución de los problemas clásicos de la inteligencia artificial.

Dentro del Contenido:

Sistema de Conocimientos

- Facilita mediante los MSP las búsquedas metaheurísticas (algoritmos genéticos).

Sistema de Habilidades

- Solución a problemas computacionalmente intratables utilizando métodos de solución de problemas y las diferentes estrategias de búsqueda existentes para ello.

En el tema número dos Métodos de Solución de Problemas en:

Sistema de habilidades:

- Aplica los métodos de solución de problemas y estrategias de búsqueda en la solución de problemas.

3.6. Conclusiones Parciales

Se capturaron un total de cinco tareas de ingeniería, que constituyeron los pasos a realizar para darle cumplimiento a las historias de usuario especificadas por el cliente. Por otra parte el conjunto de pruebas realizadas, tal como muestra la Figura 10, permitió observar la progresión en la resolución de los problemas detectados en cada iteración del ciclo de desarrollo de la aplicación. A partir de los resultados de las pruebas realizadas se puede concluir que la aplicación “Objetos de Aprendizaje para apoyar la asignatura Inteligencia Artificial” cumple satisfactoriamente con los requerimientos especificados por el cliente.

Conclusiones generales

Al finalizar las etapas por las cuales transitó el software “Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial” se puede concluir que:

- A partir de la caracterización de los tipos de Objetos de Aprendizaje se pudo concluir que, los más adecuados son:
 - Según el contenido pedagógico: procedimental y conceptual.
 - Según el formato: imágenes animadas y texto.
- Se implementaron los Objetos de Aprendizaje como módulos externos a la aplicación, permitiendo su reutilización en varios contextos de aprendizaje.
- La realización de las pruebas de aceptación a la aplicación “Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial” permitieron comprobar que la misma cumple satisfactoriamente con los requerimientos especificados por el cliente.
- El principal resultado de la presente investigación fueron dos Objetos de Aprendizaje para apoyar la disciplina Inteligencia Artificial en la Universidad de las Ciencias Informáticas.

Recomendaciones

- Continuar desarrollando Objetos de Aprendizaje de los restantes temas de la asignatura Inteligencia Artificial I, lo cual permitirá aumentar la interactividad del curso y la motivación de los estudiantes.
- Utilizar los Objetos de Aprendizaje desarrollados como apoyo al tema Métodos de Solución de Problemas de la asignatura Inteligencia Artificial I.

Referencias bibliográficas

1. **Gómez, José Ramón.** *Las TIC y sus aportaciones a la sociedad.* [Online] 2005. [Cited: 1 8, 2013.] <http://boj.pntic.mec.es/jgomez46/ticedu.html>.
2. **Fuentes, Miriam Cecilia Gómez y Joel.** *La Tecnología informática en la sociedad cubana.* [Online] [Cited: 1 9, 2013.] <http://www.pedagogiaprofesional.rimed.cu/especial%202/Joe.html>.
3. **J.Salinas.** *Nuevos ambientes de aprendizaje para una sociedad del conocimiento.* 1997.
4. **AulaGlobal.** *Bases del elearning - Que es el elearning?.* [Online] <http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=77>.
5. **Romo, Jesús Uriarte and Benítez, Manuel Gómez.** *e-Learning perspectivas de las plataformas que lo soportan.* [Online] <http://www.elearningworkshops.com/docs/estrategia/lcmsUPVcastellano.pdf>.
6. **Robertson, J.** *How to evaluate a content management system.* [Online] 1 23, 2002. http://www.steptwo.com.au/papers/kmc_evaluate/index.html.
7. **Peñalvo, Francisco García and Clara López Guzmán, Pedro Pernías Peco.** *Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS.REL.* [Online] *Revista de Educación a Distancia*, 2 2005. II.
8. **Alonso Salvador Sánchez, Oscar Sanjuán Martínez.** *Reusabilidad de objetos didácticos mediante el uso de genericidad.* [Online] <http://www.pucp.edu.pe/eventos/sissoft/trabajosacep.html>.
9. **Chiappe A. Segovia, Y. Rincon. H.** *Toward an instructional design model based on learning objects.* s.l. : *Educational Technology Research and Development*, 2007.
10. **Zarate, José Carlos Cano.** *Apuntes de Tecnología Educativa para las NT Cátedras de Comunicación Educativa Universidad Marista y Apuntes de Diseño Instruccional.* México : *Universidad de las Californias*, 2007.
11. *Los objetos de aprendizaje como recurso para la docencia universitaria: criterios para su elaboración.* España : s.n.
12. **Acuña, Jorge.** *Estudio experimental con diferentes operadores de cruce para la resolución del problema del viajante.* 2010.
13. **Espinoza, Daniel.** *El Problema del Vendedor Viajero (TSP) y Programación Entera (IP).* 2006.
14. **Karp, Richard M.** *Reducibility Among Combinatorial Problems.* [ed.] R. E. Miller y J. W. Thatcher. Nueva York : s.n., 1972. pp. 85-103.
15. **Taha, Hamdy A.** *Investigación de operaciones.* 2004.
16. **Rodríguez, Javier Ramírez.** [Online] 2004. [Cited: 5 12, 2013.] <http://eprints.ucm.es/tesis/mat/ucm-t24770.pdf>.

17. **Juan Pablo Gomez Gallego, ing Jorge Galves.** *Fundamentos de la metodologia RUP, Rational Unified Process.* [Online] Universidad Tecnológica de Pereira, 9 16, 2007. [Cited: 1 15, 2013.] <http://es.scribd.com/doc/297224/RUP>.
18. **Carvajal, Jose Carlos Riola.** *Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial.* 2008.
19. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico* 6ta Edición . 2005.
20. **Drupal Hispano.** *Comunidad de usuarios de Drupal.* [Online] 4 11, 2005. [Cited: 1 12, 2013.] <http://drupal.org.es/drupal>.
21. *10 razones para usar Drupal.* [Online] [Cited: 1 12, 2013.] <http://al.quimica.net/10-razones-para-usar-drupal>.
22. **Jommlaos.** *¿Qué es Joomla?* [Online] [Cited: 3 4, 2013.] <http://www.jommlaos.net/ique-es-joomla>.
23. *Las ventajas competitivas de utilizar Drupal.* [Online] 10 3, 2009. [Cited: 1 10, 2013.] <http://www.tratonera.com/?q=node/53>.
24. **Franganillo, Jorge.** *HTML5: el nuevo estándar básico del web.* [Online] 2010. [Cited: 2 8, 2013.] <http://franganillo.es/html5.pdf>.
25. *w3c.* [Online] [Cited: 1 20, 2013.] www.w3c.org.
26. *Historia de PHP.* [Online] [Cited: 1 15, 2013.] <http://php.net/manual/es/history.php>.
27. **Carrero, Hermanos.** *Programación en Castellano.* [Online] 1998. [Cited: 1 25, 2013.] http://www.programacion.com/articulo/por_que_elegir_php_143.
28. *Programming languages on the internet.* [Online] [Cited: 1 15, 2013.] http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php.
29. *Oracle.* [Online] [Cited: 2 10, 2013.] <http://www.oracle.com/lad/technologies/java/features/index.html>.
30. **Garrett, James. A.** *New Approach to Web Applications. adaptive path.* [Online] 2 18, 2005. [Cited: 1 15, 2013.] <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.
31. *javaHispano, tu lenguaje, tu comunidad.* [Online] 9 5, 2006. [Cited: 2 12, 2013.] http://www.javahispano.org/antiguo_javahispano_org/2006/9/5/liberado-jgap-30.html.
32. **Brea, Orlando Fabián.** *Desarrolloweb.com.* [Online] 3 16, 2005. [Cited: 3 15, 2013.] <http://www.desarrolloweb.com/articulos/1884.php>.
33. **Alvarez, Miguel Angel.** *Desarrolloweb.com.* [Online] 3 25, 2009. [Cited: 2 8, 2013.] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.
34. **G. A BOOCH, J. RUMBAUGH, I. JACOBSON, J. J. G. MOLINA.** *UML: el lenguaje unificado de modelado : guía del usuario. s.l. : Pearson Addison Wesley. 2006. ISBN 9788478290765.*

35. **Vallecillo-Moreno, L. Fuentes-Fernández y A.** An Introduction to UML Profiles. 2004. 5-13.
36. **León, E.** Tutorial Visual Paradigm for UML. [Online] 2000. [Cited: 2 11, 2013.] <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm>.
37. **Brea, Orlando Fabián.** Desarrolloweb.com. [Online] 3 3, 2005. [Cited: 3 25, 2013.] <http://www.desarrolloweb.com/articulos/1852.php>.
38. Observatorio tecnologico. [Online] 4 21, 2008. [Cited: 3 12, 2013.] <http://recursostic.educacion.es/observatorio/web/ca/software/servidores/580-elvira-mifsud>.
39. BitsBeta. NetBeants, IDE gráfica para programación. [Online] 9 10, 2010. [Cited: 1 12, 2013.] <http://bitsbeta.com/netbeans-ide-grafica-programacion>.
40. **Gómez, Julián.** Softonic. [Online] [Cited: 3 20, 2013.] <http://eclipse-sdk.softonic.com/>.
41. Zainex.es. [Online] [Cited: 3 15, 2013.] <http://www.zainex.es/tags/eclipse/netbeans-eclipse>.
42. alegsa.com.ar. [Online] [Cited: 1 15, 2013.] <http://www.alegsa.com.ar/Dic/mysql.php><http://www.alegsa.com.ar/Dic/mysql.php>.
43. **Mañas, José A.** Pruebas de Aceptación de Ingeniería de Sistemas Telemáticos. [Online] 2 5, 2004. [Cited: 3 20, 2013.] <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/aceptacion.html>.
44. **Oswaldo Castillo, Daniel Figueroa ,Hector Sevilla.** Programación Extrema.
45. **Venete, Adriana.** Introducción a los Patrones de Arquitectura. s.l. : *Universitat Jaume I (UJI)*, 2010/2011.
46. **CAMPO, I. G. D.** Patrones de Diseño, Refactorización y antipatrones. Ventajas y Desventajas de su Utilización en el Software Orientado a Objetos. s.l. : *Universidad Católica de Salta*, 2009.
47. **Laman, Craig.** UML y Patrones, introducción al análisis y diseño orientado a objetos. México : PRENTICE HALL : s.n., 1999.
48. Los objetos de aprendizaje como recurso para la docencia universitaria: criterios para su elaboración. España : s.n.

Anexos

Anexo 1 Ventajas de los Objetos de Aprendizaje

Tabla 27 Ventajas de los Objetos de Aprendizaje.

(48)

Ventajas	Estudiantes	Profesores
Personalización	Individualización del aprendizaje en función de sus intereses, necesidades y estilos de aprendizaje.	Ofrecen vías de aprendizaje alternativas. Adaptan los programas formativos a las necesidades específicas de los estudiantes.
Interoperabilidad	Acceden a los Objetos independientemente de la plataforma y hardware.	Utilizan materiales desarrollados en otros contextos y sistemas de aprendizaje.
Inmediatez/ accesibilidad	Tienen acceso, en cualquier momento, a los objetos de aprendizaje que se desee.	Obtienen, al momento, los objetos que necesitan para construir los módulos de aprendizaje.
Reutilización	Los materiales ya han sido utilizados con criterios de calidad.	Disminuyen el tiempo invertido en el desarrollo del material didáctico.
Flexibilidad	Se integran en el proceso de aprendizaje. Se adaptan al ritmo de aprendizaje del alumno.	Es de fácil adaptación a: Los distintos contextos de aprendizaje. Las diferentes metodologías de enseñanza-aprendizaje.
Durabilidad/ Actualización	Acceden a contenidos que se adaptan fácilmente a los cambios tecnológicos.	Crean contenidos que pueden ser rediseñados y adaptados a las nuevas tecnologías.