



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 5

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Sistema de Objetos de Aprendizaje Interactivos y
Experimentales para la Matemática Discreta

Autor: Yasniel Alejandro Carballo Martínez

Tutor: MSc. Alién García Hernández

Ciudad de La Habana, junio de 2014

Año 56 de la Revolución

Declaración jurada de autoría

Declaro ser autor de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yasniel Alejandro Carballo Martínez

Firma del autor

MSc. Alién García Hernández

Firma del tutor

Agradecimientos

A mi mamá por estar siempre pendiente de mí, por ser mi guía y luz, por cuidarme, apoyarme y ofrecerme lo mejor de sí en cada momento.

A mi hermano por ser más que un hermano.

A mi familia: mi abuela, mi primo, mi tío y mi tía, mi papá, a todos por su apoyo y cariño.

A mis amigos del IPVCE: Dalílys, Cheila y Luis Miguel, eternos integrantes de la mesa 7 batería B.

A Ailen, por ser mi mejor amiga durante estos 5 años, por apoyarme y aconsejarme cuando lo necesité.

A Reynol, porque de él también aprendí, aunque me dió buenos dolores de cabeza.

A mi novia, por estar a mi lado, soportarme y apoyarme.

A mi tutor por confiar en mi para desarrollar esta investigación aun cuando no me conocía, por su tiempo y apoyo.

Dedicatoria

A mi mamá por todo su esfuerzo, dedicación y cariño

Resumen

La educación es quizás hasta el momento, la que más provecho ha sacado de las potencialidades que ofrece el uso de las Tecnologías de la Informática y las Comunicaciones (TIC), expresado en términos de simplificar y hacer más eficiente el trabajo rutinario, de amplificar el potencial humano y hacer más efectiva la labor del profesor. La presente investigación está dirigida a contribuir al proceso de enseñanza-aprendizaje de la Matemática Discreta (MD) durante el primer año de la carrera de Ingeniería en Ciencias Informáticas (ICI) en la Universidad de las Ciencias Informáticas (UCI). El objetivo de la misma es el desarrollo de un Sistema de Objetos de Aprendizaje Interactivos y Experimentales (OA-IE) para la MD. Para ello fue necesario la realización de un estudio sobre los principales referentes teóricos que sustentan el desarrollo y la utilización de objetos de aprendizaje en el proceso de enseñanza-aprendizaje de la MD en la UCI, el análisis y diseño de los OA-IE para los contenidos seleccionados, la implementación de los OA-IE y la valoración de los mismos como software y como recurso pedagógico. Como metodología de desarrollo de software se seleccionó un híbrido entre la metodología XP e ISD-MeLO, como lenguaje de programación Java y el uso del Netbeans 7.4 como entorno de desarrollo integrado. El resultado obtenido cumple con el objetivo planteado y ello se verifica mediante pruebas unitarias, pruebas de sistema y el grado de aceptación por parte de los usuarios finales.

Palabras clave: Objetos de aprendizaje, Matemática Discreta, Tecnologías de la Informática y las Comunicaciones.

Índice de contenidos

| | |
|---|-----------|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 7 |
| 1.1 Importancia de la Matemática Discreta..... | 7 |
| 1.2 Uso de las TIC en el proceso de enseñanza-aprendizaje..... | 9 |
| 1.3 Recursos y medios didácticos utilizados para la enseñanza de la Matemática Discreta..... | 11 |
| 1.4 Caracterización de los Objetos de Aprendizaje como recurso de apoyo al PEA de la MD. | 14 |
| 1.5 Metodologías para el desarrollo de OA..... | 20 |
| 1.5.1 <i>ISD-MeLO (Instructional Systems Development Methodology based on e-Learning Objects)</i> | 21 |
| 1.5.2 <i>Programación Extrema (Extreme Programming, XP)</i> | 22 |
| 1.6 Herramientas y lenguaje de desarrollo de OA..... | 24 |
| 1.6.1 <i>Netbeans 7.4</i> | 25 |
| 1.6.2 <i>Lenguaje de programación JAVA</i> | 25 |
| Consideraciones generales del capítulo | 26 |
| CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA DE OBJETOS DE APRENDIZAJE ... | 28 |
| 2.1 Análisis..... | 28 |
| 2.1.1 <i>Selección de los contenidos para desarrollar los nuevos OA-IE</i> | 28 |
| 2.2 Diseño | 30 |
| 2.2.1 <i>Prototipo de interfaz del Sistema de Objetos de Aprendizaje</i> | 31 |
| 2.2.2 <i>OA-IE de Teoría Combinatoria</i> | 32 |
| 2.2.3 <i>OA-IE de Relaciones de Recurrencia</i> | 37 |
| 2.2.4 <i>Requisitos Funcionales</i> | 39 |
| 2.2.5 <i>Requisitos no funcionales</i> | 40 |
| 2.2.6 <i>Historias de usuario</i> | 40 |
| 2.2.7 <i>Iteraciones</i> | 42 |
| 2.2.8 <i>Tarjetas CRC</i> | 44 |
| 2.3 Arquitectura de Software | 45 |
| 2.3.1 <i>Patrón Modelo Vista Controlador</i> | 45 |
| 2.4 Patrones de diseño..... | 46 |

| | |
|--|-----------|
| 2.4.1 Patrones GRASP | 46 |
| Consideraciones generales del capítulo | 47 |
| CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE OBJETOS DE APRENDIZAJE | 49 |
| 3.1 Tareas de Ingeniería..... | 49 |
| 3.2 Estándares de codificación | 50 |
| 3.2.1 Indentación | 50 |
| 3.2.2 Comentarios | 51 |
| 3.2.3 Declaraciones | 52 |
| 3.2.4 Sentencias..... | 52 |
| 3.2.5 Espacio en blanco | 53 |
| 3.2.6 Convenciones de nomenclatura..... | 53 |
| 3.3 Validación de los Objetos de Aprendizaje..... | 54 |
| 3.3.1 Pruebas unitarias. Pruebas de caja blanca | 55 |
| 3.3.2 Pruebas de aceptación | 57 |
| 3.3.3 Evaluación de la calidad del sistema desde el punto de vista pedagógico | 61 |
| Consideraciones generales del capítulo | 63 |
| CONCLUSIONES GENERALES | 64 |
| RECOMENDACIONES | 65 |
| BIBLIOGRAFÍA | 66 |
| ANEXOS | 71 |
| Anexo 1: Interfaces para crear ejercicios en el OA-IE de Teoría Combinatoria..... | 71 |
| Anexo 2: Historias de Usuario | 73 |
| Anexo 3: Tarjetas CRC..... | 76 |
| Anexo 4: Tareas de ingeniería..... | 78 |
| Anexo 5: Diagrama de Gantt | 80 |
| Anexo 6: Pruebas de aceptación | 81 |
| Anexo 7: Guía para la validación de los OA-IE como software pedagógico | 84 |

Introducción

El avance acelerado de las Tecnologías de la Información y las Comunicaciones (TIC) y su impacto en la sociedad, están cambiando los patrones de desarrollo económico y social; al ser útiles en casi todas las esferas, en la medicina, la economía, en la automatización de los procesos industriales, la producción de bienes y servicios, la cultura, la educación, entre otras.

María Elena Pardo y otros (Gómez, et al., Febrero 2005) definen a las TIC como un conjunto de recursos tecnológicos que convenientemente asociados, permiten el adecuado registro, tratamiento, transformación, almacenamiento, utilización, presentación y circulación de la información y cuyo paradigma son las redes informáticas (Internet, Intranets), que posibilitan múltiples aplicaciones: correo electrónico, charlas electrónicas, teleconferencias, bibliotecas virtuales, entre tantas otras. Estas tecnologías son de carácter innovador y creativo.

La utilización de las TIC en la educación cobra gran importancia debido a que:

- ✚ Potencian la educación a distancia, al brindar medios de comunicación y adquisición de información de todo tipo, a los cuales las personas pueden acceder por sus propios medios, además de permitir el aprendizaje interactivo.
- ✚ Permiten al alumno poder acceder a toda la información con la ayuda mínima del profesor.

La inserción de las TIC a la Enseñanza-Aprendizaje se hace con el propósito de mediar los recursos tecnológicos que constituyen un medio y no una finalidad. Un medio que contribuye en el marco del modelo pedagógico de los nuevos ambientes de aprendizaje, a optimizar la actividad y la comunicación de los maestros con los alumnos, de estos entre sí y de ellos con el contenido de enseñar (Montenegro, 2007).

Ante el reto que significa el uso de las TIC, Cuba busca estrategias para aprovechar su uso y obtener ventajas del mismo. El país se propuso como objetivos la utilización de las TIC como soporte técnico para los programas educacionales y de formación profesional, introducir a los estudiantes, desde la primaria en el uso de las mismas, así como la creación de Instituciones Especializadas en la Enseñanza de estas tecnologías (Universidad de Ciencias Informáticas (UCI) y Politécnicos de Informática) y la introducción de materias docentes sobre las TIC, en diferentes carreras técnicas y universitarias como parte de su currículo.

El desarrollo de la Educación Superior en Cuba, marcha aparejado a todos estos avances en el uso de las TIC. Las universidades cubanas las han adoptado, destacándose en tres ámbitos

diferentes. El primero de ellos se refiere a su uso en cuestiones administrativas, el segundo al uso de las mismas en la investigación, donde profesionales de prestigio en diferentes países del mundo pueden colaborar sin necesidad de moverse de su sitio y el tercero, se refiere a su uso como medio en la enseñanza (Granda Dihigo, y otros, 2011).

El desarrollo científico y tecnológico y la educación en matemáticas son inseparables. Las matemáticas son el lenguaje de las ciencias y son vitales para su desarrollo y aplicación. Los físicos, químicos, ingenieros, biólogos, sociólogos, administradores de empresas, economistas, necesitan de una u otra forma, conocer y manejar este lenguaje. La enseñanza de las matemáticas ocupa una posición estratégica en el sistema educativo, y el nivel de preparación científica y tecnológica puede elevarse más fácilmente si los conocimientos matemáticos se imparten oportuna y adecuadamente (Silva Díaz, 2003).

Leonardo Da Vinci, afirmó que “No hay ninguna conclusión científica en la que no se apliquen las matemáticas” (Murcia, 2009). Luego, los conocimientos matemáticos se logran cuando los estudiantes son capaces de realizar abstracciones matemáticas a partir de obtener información, observar propiedades, establecer relaciones y resolver problemas concretos. Es por ello que se hace necesario llevar al aula situaciones cotidianas que supongan desafíos matemáticos atractivos y el uso habitual de recursos y materiales didácticos para ser manipulados por los estudiantes.

Las TIC se han convertido en un valioso instrumento didáctico en ese sentido, convirtiendo los ordenadores en estupendos laboratorios matemáticos que le permiten al alumno experimentar, suplir sus carencias en el conocimiento matemático, desarrollar la intuición, conjeturar, comprobar, demostrar y ver las situaciones matemáticas de una forma práctica.

Con el surgimiento de la ciencia de la computación, fue necesario trabajar con conjuntos numerables, la necesidad de computar se hacía imprescindible, por lo que la Teoría de Grafos, la Lógica y la Teoría Combinatoria entre otros temas condujeron al surgimiento de una nueva área de la matemática: las Matemáticas Discretas (MD). La MD es una asignatura que contribuye en gran medida a la formación del Ingeniero en Ciencias Informáticas, ya que permite crear y desarrollar el pensamiento lógico.

La UCI, centro de altos estudios, cuya misión es ser una universidad innovadora, de excelencia científica, académica y productiva que forma de manera continua profesionales integrales, centra su proceso educativo en el aprendizaje, pretendiendo que el estudiante adquiera un papel más

activo y una mayor responsabilidad en el proceso de formación (UCI, 2010). En su plan de estudio, la carrera de Ingeniería en Ciencias Informáticas (ICI) contempla las asignaturas de Matemática Discreta I y Matemática Discreta II durante el primer año de la carrera.

La mayoría de los estudiantes presentan dificultades en el aprendizaje de los contenidos de esta parte esencial de las matemáticas, por lo que se hace necesario apoyar el Proceso de Enseñanza-Aprendizaje (PEA) con medios que permitan al estudiante apropiarse de una manera amena y práctica de los contenidos de esta asignatura.

Los Objetos de Aprendizaje (OA), según propone David Wiley en el año 2001, son cualquier recurso digital que puede ser usado como soporte para el aprendizaje (Wiley, 2002). Actualmente son uno de los recursos que vincula las TIC con el PEA, que facilita el proceso de asimilación de los estudiantes.

En la UCI los estudiantes tienen la posibilidad de explotar estas facilidades que brinda el uso de las TIC para el PEA. Estos pueden realizar su auto-preparación en los laboratorios docentes, además de contar con un Entorno Virtual de Aprendizaje (EVA), al cual pueden acceder y obtener, utilizar o compartir materiales didácticos. Además existe un Repositorio de Objetos de Aprendizaje (ROA), espacio creado para apoyar a la comunidad universitaria en la gestión de los recursos didácticos.

Sin embargo los recursos que se encuentran en el EVA referentes a la MD son principalmente documentos en formato pdf, presentando la información teórica de los temas y con una pobre variedad de ejercicios y tareas, limitando la asimilación adecuada de los contenidos. Por su parte el ROA cuenta con algunos OA sobre MD, pero estos carecen de interactividad, limitando la posibilidad de los estudiantes de experimentar con estos contenidos. Se desarrollaron en la UCI varios OA para apoyar la enseñanza de la MD, pero estos no abarcan todos los contenidos de la asignatura, no están integrados en un sistema y no permiten su publicación en la web.

Los **problemas** detectados en el aprovechamiento de las TIC en el PEA de la MD en la UCI se resumen a continuación:

- ✚ Los materiales que se encuentran en el espacio de la asignatura en el EVA solo proporcionan orientación, sin permitir la experimentación con los contenidos de la MD.
- ✚ Existe dentro del programa de las asignaturas de MD una pobre variedad de ejercicios y tareas, lo que limita la ejercitación adecuada de los contenidos.
- ✚ El aprovechamiento de las potencialidades de las TIC en el PEA de la MD es insuficiente.

- ✚ Es insuficiente la existencia de recursos de enseñanza que apoyen el PEA de la MD para aquellos contenidos de más difícil y lenta asimilación, que tienen la posibilidad de ser representados de manera gráfica permitiendo la interactividad y la experimentación con el contenido.
- ✚ Los OA desarrollados por (Herrera Chica, 2013) para apoyar la enseñanza de la MD no abarcan todos los contenidos de la asignatura, sobre todo aquellos que presentan gran dificultad de aprendizaje para los estudiantes; no están integrados en un sistema, sino que interactúan por separado como entes de conocimiento aislado y no permiten su publicación en la web.

Lo antes expuesto permite identificar como **problema científico** a resolver:

¿Cómo aprovechar las TIC en el PEA de la MD en la UCI mediante recursos que permitan la experimentación con los contenidos?

Por lo que se define como **objeto de estudio** el Proceso de Enseñanza-Aprendizaje de la Matemática Discreta con el uso de las TIC, definiendo como **objetivo general** desarrollar un sistema de objetos de aprendizaje interactivos y experimentales para contribuir al proceso de enseñanza-aprendizaje de la Matemática Discreta en la UCI.

Se determina como **campo de acción** los objetos de aprendizaje para apoyar el PEA de la MD en la UCI.

Para darle cumplimiento al objetivo planteado se responderán las siguientes **preguntas científicas**:

1. ¿Cuáles son los presupuestos teóricos que sustentan el desarrollo y la utilización de Objetos de Aprendizaje Interactivos y Experimentales en el proceso de enseñanza-aprendizaje de la Matemática Discreta en la UCI?
2. ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y diseño del Sistema de Objetos de Aprendizaje Interactivos y Experimentales para la Matemática Discreta en la UCI?
3. ¿Cómo implementar, a partir del análisis y diseño realizado, el Sistema de Objetos de Aprendizaje Interactivos y Experimentales para la Matemática Discreta en la UCI?
4. ¿Qué resultados se obtendrán al intentar validar el Sistema de Objetos de Aprendizaje Interactivos y Experimentales para la Matemática Discreta en la UCI?

Para responder las preguntas anteriores se definen las siguientes **tareas de investigación**:

1. Sistematización de los principales referentes teóricos que sustentan el desarrollo y utilización de Objetos de Aprendizaje Interactivos y Experimentales en el proceso de enseñanza-aprendizaje de la Matemática Discreta en la UCI.
2. Análisis y diseño de los OA-IE para los contenidos seleccionados.
3. Implementación de los OA-IE para los contenidos seleccionados.
4. Valoración de la calidad como software y como recurso pedagógico de los OA-IE elaborados.

Los métodos de investigación a utilizar para dar cumplimiento a las tareas de investigación planteadas serán los siguientes:

Nivel teórico

- ✚ Análisis documental para la revisión bibliográfica, la revisión de las fuentes primarias de investigación, el estudio de documentos normativos, como los programas de las disciplinas, de la asignatura, orientaciones metodológicas, entre otros.
- ✚ Inducción-deducción como procedimiento para hacer las inferencias y establecer las relaciones pertinentes en la estructuración de los OA-IE.
- ✚ Análítico-Sintético para el estudio y sistematización de las tendencias de utilización de recursos didácticos para el PEA de la MD, fundamentos del objeto de investigación y su concreción en la propuesta presentada.
- ✚ Enfoque de sistema para determinar y estructurar algunos de los componentes del PEA a través de los OA-IE.

Nivel empírico

- ✚ Encuesta a profesores y a estudiantes, para recoger datos y criterios que permitieron mantener actualizado el diagnóstico de la situación problemática. Se aplicaron encuestas a los profesores para valorar la factibilidad de aplicación de los OA-IE.

Nivel estadístico

- ✚ Estadística descriptiva para el análisis de las encuestas y la revisión documental de las notas de los estudiantes y para valorar la factibilidad de aplicación de los objetos de aprendizaje interactivos y experimentales.

El informe de la investigación consta de introducción, tres capítulos, conclusiones generales, recomendaciones y referencias bibliográficas utilizadas durante el desarrollo de la misma, por último se incluye un cuerpo de anexos que complementa la investigación.

En el primer capítulo denominado **“Fundamentación teórica”**, se realiza una valoración de la importancia de la MD para la formación de futuros profesionales de las TIC. Se argumenta el uso de las TIC en la educación caracterizando los OA como recursos de apoyo al PEA de la MD. Finalmente se caracterizan las herramientas, lenguajes y metodologías estudiadas para la implementación. El Capítulo 2: **“Análisis y diseño del Sistema de objetos de aprendizaje”**, describe detalladamente el proceso de selección de los contenidos más convenientes para la realización de los OA-IE, dando paso al diseño e implementación de los OA-IE para los contenidos seleccionados garantizando que estos tengan niveles de interacción y experimentación adecuados. En este capítulo se muestra el proceso de integración de los OA-IE desarrollados con los ya existentes. En el tercer y último capítulo titulado **“Implementación y validación del Sistema de Objetos de Aprendizaje”**, se aborda la implementación de los OA-IE para los contenidos seleccionados, logrando finalmente la integración de los mismos en un sistema junto con los OA desarrollados en (Herrera Chica, 2013), y se realizan las pruebas pertinentes a cada OA-IE para verificar que estos cumplan con el objetivo planteado como software y desde el punto de vista pedagógico.

Capítulo 1: Fundamentación teórica

El presente capítulo persigue como objetivo precisar los elementos teóricos que dan sustento a la investigación. Se realiza una valoración de la importancia de la Matemática Discreta en la formación de los futuros profesionales de las TIC, se argumenta sobre el uso de las TIC en el proceso de enseñanza-aprendizaje, y se caracterizan los OA como recursos de apoyo al PEA de la MD. Finalmente se caracterizan las herramientas, lenguajes y metodologías consideradas para dar solución al problema de la investigación.

1.1 Importancia de la Matemática Discreta.

La invención y el continuo progreso de las computadoras, que al comienzo eran máquinas mecánicas analógicas y después máquinas electrónicas, permitieron trabajar con cantidades cada vez más grandes de datos. Surgen entonces áreas como la Teoría de la Computabilidad, la Teoría de la Complejidad Computacional, el procesamiento de señales, el análisis de datos, la optimización y otras áreas de investigación de operaciones. En siglos precedentes, muchos de los focos matemáticos estaban centrados en el cálculo y las funciones continuas, pero el surgimiento de la computación y las tecnologías de las comunicaciones llevan a una importancia creciente los conceptos de las matemáticas discretas y la expansión de la combinatoria, incluyendo la Teoría de Grafos.

Alrededor de los años 60 del siglo XX, se advirtió que los futuros especialistas en el campo de la Informática debían adquirir conocimientos que no estaban contemplados en los cursos clásicos de Álgebra y Cálculo Diferencial e Integral. Surgieron así cursos y libros de texto que incluían temas muy clásicos y otros muy modernos de la Matemática. Se puede mencionar entre los primeros la Teoría Elemental de Números, la Combinatoria de la Enumeración, la Teoría de Grafos, y en particular Árboles, así como temas clásicos de Lógica (lógica proposicional, lógica de predicados, lógica difusa), del Álgebra de Boole, junto con temas más modernos como Lenguajes y Máquinas de Estados Finitos, Teoría de Algoritmos, Complejidad Computacional, por mencionar algunos (García, 2014; Canavelli, y otros, 2007).

La matemática tiene un carácter formativo para un científico cualquiera y también para un ingeniero. Escribir con claridad, formalizar, adquirir destrezas para enfrentar situaciones nuevas, precisión, constancia, son habilidades fundamentales para el científico o el ingeniero y para el que las matemáticas son un instrumento adecuado de desarrollo (ESCET, 2003).

Se está viviendo la “era digital” (computadoras, cámaras digitales, teléfonos celulares, MP3, MP4). Esta “era” trae aparejados cambios muy profundos en los usos y costumbres sociales. Sin embargo, son pocos los que entienden, o simplemente tienen presente, la crucial importancia de la matemática en todas estas realizaciones. (...) quien se imagina cuando está hablando por el celular cuánta matemática está involucrada en ese proceso y cómo opera la Teoría Algebraica de la Codificación para que el mensaje llegue con tanta nitidez. Todos usamos tarjetas de débito, operamos con tarjetas de crédito, se hacen compras por internet, pero no tenemos presente que allí intervienen la Codificación, la Comprensión de Datos y la Criptografía. Pues bien, (...) es la llamada Matemática Discreta, que está en el fundamento de todas las modernas Teorías de la Información y de la Comunicación (Canavelli, y otros, 2007).

La Matemática Discreta es la parte de la matemática que estudia a los objetos discretos, o sea, constituidos por elementos distintos o inconexos. La MD es la puerta a cursos más avanzados en todas las partes de la matemática. Proporciona la base a muchos cursos de Ciencias de la Computación, incluyendo Estructura de Datos, Algoritmos, Teoría de Base de Datos, Teoría de Automatas, Lenguajes Formales, Teoría de Compiladores, Seguridad Informática y Sistemas Operativos (Rosen, et al., 2004).

A menudo los estudiantes encuentran sus cursos en Ciencias de la Computación mucho más difíciles cuando no tienen una apropiada base en MD.

La MD contiene el sustrato matemático necesario para resolver problemas en investigación operativa, incluyendo muchas técnicas de optimización, química, ingeniería, biología, entre otras (Rosen, et al., 2004).

Haciendo uso de las MD se pueden dar respuesta a preguntas como:

- ¿De cuántas formas se puede elegir una clave de acceso a un equipo informático?
- ¿Hay algún enlace entre dos ordenadores en una red?
- ¿Cuál es el camino más corto entre dos ciudades?
- ¿Cómo se puede ordenar una lista de enteros para que se dispongan en orden creciente?
- ¿Cómo se puede demostrar que un algoritmo ordena correctamente una lista?
- ¿Cómo se puede diseñar un circuito para sumar dos enteros?
- ¿Cuántas direcciones válidas de internet existen? (Rosen, y otros, 2004)

La MD ha cobrado gran importancia debido, entre otras cosas, a que en los equipos informáticos la información se almacena y manipula de manera discreta. Está ligada profundamente a las

Ciencias de la Computación pues los objetos con los que trata un ordenador son objetos finitos, y por el triunfo de la tecnología digital, que en muchos aspectos es una representación discreta de la realidad.

La MD se vincula estrechamente con otras materias que se estudian como parte del plan de estudio de la carrera de Ingeniería en Ciencias Informáticas, lo cual hace esencial su estudio antes de adentrarse en otras como Base de Datos, Teleinformática, Arquitectura de Computadoras, Programación II, Inteligencia Artificial, por mencionar algunas. Encontramos por ejemplo que el uso de la Teoría de Grafos permite la resolución e implementación de algoritmos de ruteo o de camino; internamente, los routers, mantienen "en memoria" un grafo para representar la red, cuando el router debe transmitir un paquete aplica algunas operaciones y determina por qué ruta es más económica o viable transmitirlo (Teleinformática), lo cual hace necesario del conocimiento de grafos para comprender estos algoritmos de ruteo. Puede valerse de grafos para representar las disposiciones y conexiones de las calles de una ciudad, y aplicando Dijkstra podría encontrarse el camino entre dos puntos cualesquiera (Programación II). En el área de las Redes Neuronales, el uso de grafo asiste al diseño y representación de éstas; cada neurona (nodo) se comunica con otras y se pueden transmitir información. Muchos sistemas de Inteligencia Artificial se apoyan en el uso de las redes neuronales. Para la Gestión de Proyectos, se utilizan técnicas como PERT en las que se modelan los mismos utilizando grafos y optimizando los tiempos para concretar los mismos (Gestión de Software). La estructura de enlaces de un sitio web puede ser representada por un grafo dirigido, en el que los vértices representan páginas web y las aristas dirigidas representan enlaces de una página a otra. La Teoría de Conjuntos y la Lógica de Predicados de primer orden proporcionan los elementos básicos necesarios para crear una base de datos relacional (Base de datos). La Máquina de Turing es particularmente útil en la explicación de las funciones de un CPU y puede ser adaptada para simular la lógica de cualquier algoritmo de computador. Por su parte las Relaciones de Recurrencia encuentran su aplicación en el estudio de los árboles binarios, pilas y algoritmos de ordenación, constituye una base sólida para el estudio de los algoritmos recursivos (Programación).

1.2 Uso de las TIC en el proceso de enseñanza-aprendizaje.

En la actualidad no se puede hablar sobre educación en el siglo XXI sin hacer alusión al uso de las TIC y las posibilidades que ofrece su utilización en el proceso de enseñanza-aprendizaje, a través de la comunicación mediada por el ordenador y los entornos virtuales.

Las llamadas Tecnologías de la Información y las Comunicaciones (TIC) son el resultado de las posibilidades creadas por la humanidad en torno a la digitalización de datos, productos, servicios y procesos, y de su transportación a través de diferentes medios, a grandes distancias y en pequeños intervalos de tiempo, de forma confiable, y con relaciones costo -beneficio nunca antes alcanzados por el hombre (Castañeda Hevia, 2009).

La incorporación de las TIC en el proceso de enseñanza-aprendizaje permite al estudiante la realización interactiva de simulaciones. Puede repetirse una actividad o experimento tantas veces como se desee, cambiar parámetros y condiciones, analizar situaciones límites, observar los elementos significativos de una actividad, proceso o fenómeno, o descomponer un producto en sus partes o en el proceso seguido para su elaboración.

Las TIC ofrecen diversidad de recursos de apoyo a la enseñanza como materiales didácticos, entornos virtuales, internet, blogs, wikis, webquest, foros, chat, mensajerías, videoconferencias, y otros canales de comunicación y manejo de información, desarrollando la creatividad, la innovación, los entornos de trabajo colaborativo, y promoviendo el aprendizaje significativo, activo y flexible.

Varios autores han investigado acerca de la inclusión de las TIC en la educación entre los que se encuentran (Castañeda Hevia, 2009), (Duart Montoliu, et al., 2000), (Ferro Soto, et al., julio 2009), (García-Valcárcel Muñoz-Repiso, 2009), (Marqués Graells, 2011), (Salinas Ibañez, 2004), haciendo referencia a algunas de las ventajas que trae consigo su uso en este contexto:

- ✚ Posibilitan el acceso a la información a través de las redes.
- ✚ Facilitan el desarrollo de la enseñanza a distancia o semipresencial.
- ✚ Extienden el aprendizaje fuera del marco institucional de la escuela.
- ✚ Potencian el aprendizaje personalizado.
- ✚ Aumentan la motivación al permitir mostrar materiales novedosos y actividades de forma atractiva.
- ✚ Exigen un mayor protagonismo del estudiante en el proceso docente, donde el profesor pasa de ser la principal fuente de información a ser un orientador en la búsqueda del conocimiento.
- ✚ Contribuyen al desarrollo del pensamiento y al trabajo independiente de los estudiantes.
- ✚ Brindan nuevos canales para interactuar, socializar y construir colectivamente conocimientos, venciendo barreras importantes de tiempo y espacio geográfico.

1.3 Recursos y medios didácticos utilizados para la enseñanza de la Matemática Discreta.

Los recursos didácticos son todos aquellos apoyos pedagógicos que refuerzan la actuación del profesor, que optimizan el proceso de enseñanza-aprendizaje, o sea todos aquellos procesos o instrumentos que ayudan al profesor y los alumnos a alcanzar sus objetivos de aprendizaje.

La utilización de los recursos didácticos debe consistir en un proceso organizado y sistematizado que facilite la interpretación de los contenidos que se han de enseñar. La correcta selección y utilización de los diferentes recursos va a condicionar la eficacia del proceso formativo. De este modo pueden convertirse en verdaderos instrumentos del procesamiento, de innovación, de motivación del aprendizaje, facilitando la acción procedimental o metodológica, la expresión de valores, emociones, comunicaciones (Moya Martínez, 2010).

Los recursos didácticos informáticos son medios de comunicación diseñados para interactuar con el usuario, la utilización de estos supone un gran avance en la didáctica general, son recursos que permiten procesos de aprendizaje autónomos en los que se consolidan los principios del aprender a aprender, siendo el alumno partícipe directo o guía de su propia formación (Moya Martínez, 2010).

El proceso de enseñanza-aprendizaje de las matemáticas, y en particular el de la MD, es sumamente complejo. El uso de medios tecnológicos para la enseñanza-aprendizaje de las matemáticas presenta diferentes factores metodológicos favorables:

- ✚ Facilita la adquisición de conceptos. Utilizar el ordenador como instrumento para adquirir conceptos o profundizar en ellos, permite detectar esquemas no suficientemente precisos y transformarlos en otros más adecuados.
- ✚ Permite el tratamiento de la diversidad. Ayuda a crear un ambiente de trabajo grato y estimulante que respeta las peculiaridades y el ritmo de aprendizaje del alumnado.
- ✚ Fomenta el trabajo en grupo. El trabajo en el ordenador se puede realizar en grupo, permitiendo a los alumnos y alumnas explicar a los demás sus ideas, estableciendo la comunicación y el enriquecimiento de pensamientos.
- ✚ Valora positivamente el error. El error no ha de equipararse a fracaso. Poner de manifiesto los errores de los alumnos adquiere una dimensión positiva y es una condición necesaria para superarlos.

- ✚ Realiza con rapidez y facilidad simulaciones de experimentos. El carácter imprevisible y aleatorio que tiene el azar está sujeto a leyes que sólo son perceptibles cuando consideramos un número de datos muy elevado, por lo que el procesador es el instrumento adecuado para manipular dicha información. Las aplicaciones informáticas llevan incorporados programas para generar números aleatorios, lo que permite simular procesos de azar.
- ✚ La capacidad para representar gráficamente la información. La facilidad que tienen ciertas aplicaciones informáticas para simultanear información gráfica y numérica es un apoyo indiscutible para el estudio de funciones o la estadística.
- ✚ Es un elemento motivador. Además, en la actualidad, es el medio habitual del alumno en su vida cotidiana. (Molero Aparicio, y otros, 2008)

En el ámbito internacional se pueden encontrar gran cantidad de recursos de apoyo a la enseñanza-aprendizaje de las Matemáticas. Un ejemplo de esto es el Proyecto Descartes desarrollado por el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF) del Ministerio de Educación, Cultura y Deporte de España. El proyecto Descartes¹ ofrece materiales didácticos para el aprendizaje de las matemáticas, en los niveles de enseñanza no universitaria, que son controlables por el profesor en un tiempo razonable, son fáciles de usar por los alumnos, no tienen que emplear tiempo en aprender su manejo, cubren los contenidos del currículo correspondiente al curso donde se vayan a usar, son adaptables por cada profesor a la didáctica y metodología que crea más conveniente para los alumnos con los que va a trabajar. Descartes es un applet configurable, diseñado para presentar interacciones educativas con números, funciones y gráficas. Puede ser utilizado por los autores de páginas Web educativas para enriquecer sus materiales con una amplia variedad de modelos matemáticos interactivos. Presenta bloques de aplicaciones sobre Álgebra, Geometría, Análisis, Estadística y Probabilidad y Matemáticas Aplicadas.

Otros ejemplos lo constituyen:

- ✚ PUEMAC que es un Proyecto Universitario de Enseñanza de las Matemáticas por Computadora de la Universidad Nacional Autónoma de México. Combina applets con tecnología Flash. Estructurado por niveles. En línea en <http://arquimedes.matem.unam.mx/PUEMAC/index.html>

¹ <http://recursostic.educacion.es/descartes/web/>

- ✚ Simulaciones Estadísticas telemáticas (SET). Applets interactivos que mediante simulaciones desarrollan contenidos usuales de la Estadística básica y universitaria. Han sido elaborados por el alumnado de la Escuela Politécnica Superior de la Universidad de Córdoba bajo la dirección del profesorado del área docente de Estadística. En línea en http://www.uco.es/simulaciones_estadisticas/
- ✚ LEMAT (Libro Electrónico de Matemáticas) es un conjunto de materiales multimedia elaborado con el propósito de proporcionar al usuario una gran interactividad en el proceso de aprendizaje de las Matemáticas. Además, las herramientas de autoevaluación y evaluación que incluye permiten al alumno saber cuándo ha superado un hito determinado en su formación. En línea en <http://www.lemat.unican.es>
- ✚ Curso interactivo Matemáticas 0 y Universidad es una página del profesor de la Universidad de A Coruña José Jesús Cendán que usando JavaScript y Geogebra desarrolla un curso de introducción a la universidad (Matemáticas 0) así como contenidos de ese nivel. En línea en <http://dm.edc.es/elearning/>
- ✚ Applets Java de Matemáticas. Es una colección de applets de Matemáticas de distintos niveles, incluyendo algunos en 3 dimensiones. Los temas son de Geometría elemental, Geometría esférica, Trigonometría, Vectores 3D, Análisis y Números Complejos. Se pueden descargar. En línea en <http://www.walter-fendt.de/m14s/>
- ✚ Cabri Java. Además de las construcciones geométricas que permite Cabri, esta página ofrece la posibilidad de bajar el applet y muestra el código necesario para transformar un archivo de Cabri en un applet que se inserta en una página html. En línea en <http://www-cabri.imag.fr/cabrijava/>
- ✚ Geonex. Es un applet para crear construcciones geométricas con un número variado de herramientas. Las construcciones pueden ser cambiadas al tratarse de una Geometría dinámica. Aunque la página original está en alemán se puede ver en un total de 17 idiomas, entre ellos el español. Se puede descargar de forma gratuita y es de libre uso. En línea en <http://geonext.uni-bayreuth.de/>
- ✚ Cut the knot. Cientos de applets de juegos, misceláneas, curiosidades, pruebas de Matemáticas en general clasificados por temas. En línea en <http://www.cut-the-knot.org/ctk/index.shtml>
- ✚ WWW Interactive Mathematics Server (WIMS). Páginas francesas traducidas, en parte, a otros idiomas. Es un servidor de Internet diseñado para la educación matemática. En línea en http://wims.unice.fr/wims/en_home.html

- ✚ Mathématikos. Applets de Fractales, Geometría, Análisis, Ecuaciones diferenciales, Curvas planas, entre otros. También permite descargar los applets desde la web. En línea en <http://perso.wanadoo.fr/jpq/index.htm>
- ✚ En el sitio web² del Departamento de Matemática Aplicada de España se pueden encontrar algunas aplicaciones en java sobre MD. Estas abordan los contenidos de Aritmética entera y modular, Grafos y Álgebras de Boole.

En la UCI también se han desarrollado varios recursos de apoyo al PEA de la MD, estos se encuentran publicados en un Repositorio de Objetos de Aprendizaje. Algunos de estos son: Ejercicios resueltos sobre Teoría Combinatoria, Entrevista a Alan Turing, Introducción a las deducciones lógicas, Mapas de Karnaugh, Sistema Numérico Binario, entre otros.

A partir del análisis de lo anteriormente expuesto se puede concluir que la utilización de recursos didácticos basados en las TIC permite al estudiante la construcción de su propio aprendizaje, la realización de operaciones complejas, la representación gráfica de la información, el estudiante puede aprender de sus errores a través de la retroalimentación. Es por esto que las TIC constituyen un recurso capaz de acompañar la enseñanza aprendizaje de cualquier materia, lo que supone un gran avance en la didáctica general. En la UCI no se aprovechan adecuadamente las potencialidades que ofrece el uso de las TIC en el PEA de la MD, existe una pobre utilización de los recursos que se encuentran en el Repositorio de Objetos de Aprendizaje referentes a este contenido, debido en gran medida a que estos son en su mayoría recursos estáticos que no permiten la interactividad y experimentación de quien los utiliza con el contenido que se estudia.

1.4 Caracterización de los Objetos de Aprendizaje como recurso de apoyo al PEA de la MD.

La aparición de las TIC y su aplicación en los procesos formativos han dado origen a lo que se conoce como e-learning, la cual se puede definir como un “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y comunicación con otros participantes” (Alejandro de Alvarado, 2006).

La elaboración de recursos y materiales didácticos que apoyen el actuar del educando ha sido siempre una prioridad para los docentes, con el surgimiento del e-learning la forma en que estos

²<http://neumann.dma.fi.upm.es/>

recursos se desarrollan varía en función de alcanzar en ellos características como la reutilización, interoperabilidad, durabilidad y accesibilidad. Los OA son una de las alternativas que actualmente se están utilizando para lograr esas características.

Son muchas las definiciones que se le han dado a este tipo de recursos. El término OA (RLO Reusable Learning Object en la bibliografía sajona) fue introducido por Wayne Hodgins en 1992. En el contexto de la Universidad Politécnica de Valencia se definen como “la unidad mínima de aprendizaje, en formato digital, que puede ser reusada y secuenciada. Se conciben, por tanto, estos pequeños componentes como elementos integrados e integradores del proceso de enseñanza-aprendizaje, ofreciendo a los estudiantes la posibilidad de mejorar su rendimiento y nivel de satisfacción” (PACE, 2013).

Según (López, G) los OA son cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning (López Guzmán, 2005).

Del Moral y Cernea (2005) los definen como unidades mínimas de contenido didáctico con significado propio, constituidas por paquetes de información multiformato y carácter interactivo, orientadas al logro de un determinado objetivo educativo, identificables por metadatos, e integradas por contenidos, recursos, actividades y evaluación. Destacados por su reusabilidad, compatibilidad técnica, adaptabilidad y durabilidad (Del Moral Pérez, et al., 2005).

El Ministerio Nacional de Educación de Colombia los define como todo material estructurado de una forma significativa, asociado a un propósito educativo y que corresponda a un recurso de carácter digital que pueda ser distribuido y consultado a través de la Internet. El OA debe contar además con una ficha de registro o metadato, consistente en un listado de atributos que además de describir el uso posible del objeto, permitan la catalogación y el intercambio del mismo (Colombia Aprende, 2005).

Un concepto más estructurado lo propone el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), el cual conceptualiza los OA como cualquier entidad digital o no digital que puede ser usada, re-usada o referenciada para el aprendizaje soportado en tecnología (IEEE, 2002).

Según se propone en (PACE, 2013) los OA se pueden clasificar atendiendo al tipo de contenido pedagógico y al formato.

Según los contenidos pedagógicos pueden ser:

- ✚ 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 pueden ser imagen, texto, sonido y/o multimedia.

Wiley (Wiley, 1999) afirma que la granularidad y reusabilidad representan las dos propiedades más importantes de los OA. El concepto de granularidad hace referencia a resaltar una concepción de objetos como pequeñas unidades, que pueden ser acopladas y/o adicionadas de diversas maneras.

Por la reutilización y granularidad según como se explica en (Wiley, 2001) los OA se pueden clasificar como:

- ✚ Fundamentales. Son objetos que no pueden ser subdivididos pues están en su estado atómico, por ejemplo una fotografía de un pianista.
- ✚ Combinados-cerrados. Son objetos que pueden ser combinados o ensamblados con un número pequeño de objetos en una relación directa, por ejemplo un objeto de video, acompañado de un objeto de audio.
- ✚ Combinados-abiertos. Son objetos que pueden ser combinados o ensamblados con cualquier otro objeto sin alguna restricción. Por ejemplo una página web que contenga fotos, objetos de audio y objetos texto.
- ✚ Generación de presentación. Este tipo de objetos presentan un nivel de complejidad alto, un ejemplo podría tener un applet de Java que fuera dibujando las notas musicales en un pentagrama.
- ✚ Generación instruccional. Este tipo de objetos son los encargados de instruir y proveer prácticas, por ejemplo enseñar música para y al mismo tiempo entregar ejercicios de práctica musical, para permitir una dinámica entre el sujeto.

En (Aguilar Cisneros, et al., 2004) se proponen como características básicas de los OA:

- ✚ Accesibilidad: El OA debe ser etiquetado, lo que permite su almacenamiento y referencia.
- ✚ Reusabilidad/Adaptabilidad: Debe ser funcional para varios contextos de aprendizaje.
- ✚ Interoperabilidad: Independiente del medio de entrega y del sistema de administración de aprendizaje.

Por su parte en (PACE, 2013) se expresa que los OA deben cumplir con las siguientes características para ser considerados como tal:

- ✚ Formato digital: tiene capacidad de actualización y/o modificación constante; es decir, es utilizable desde Internet y accesible para muchas personas simultáneamente y desde distintos lugares.
- ✚ Propósito pedagógico: su objetivo es asegurar un proceso de aprendizaje satisfactorio. Por tanto, no solo incluye el contenido, sino que también guía el proceso de aprendizaje del estudiante.
- ✚ Contenido interactivo: implica la participación activa de cada individuo (profesor - alumno) en el intercambio de información. Para ello es necesario que incluya actividades, ejercicios, simulaciones, cuestionarios, diagramas, gráficos, diapositivas, tablas, exámenes, experimentos, que permitan facilitar el proceso de asimilación y el seguimiento del progreso de cada alumno.
- ✚ Es indivisible e independiente de otros OA, por lo que: debe tener sentido en sí mismo y no puede descomponerse en partes más pequeñas.
- ✚ Es reutilizable en contextos educativos distintos.

De acuerdo con (Fernandez Rincon, y otros) las características de los OA son:

- ✚ Tamaño: suficiente para ser usado como parte de una lección o módulo.
- ✚ Reutilizables: pueden ser usados en diferentes unidades, o tener la opción de ser utilizado en diferentes actividades de la enseñanza y del aprendizaje.
- ✚ Accesibles: con posibilidad de ser localizados y usados con facilidad.
- ✚ Durables: permiten su uso durante un tiempo considerable.
- ✚ Interoperables: que pueden ser usados en diferentes plataformas tecnológicas o diferentes sistemas de administración de curso.

Los principales atributos de los OA según se propone en (López Guzmán, 2005) son:

- ✚ Reusabilidad: es cuando un objeto tiene capacidad para ser usado en contextos y propósitos educativos diferentes y puede adaptarse y combinarse dentro de nuevas secuencias formativas. También debe tener una tecnología, una estructura y los componentes necesarios para ser incluido en diversas aplicaciones.
- ✚ Accesibilidad: fácil de ser identificados, buscados y encontrados gracias al correspondiente etiquetado a través de diversos descriptores (metadatos) que permitirían la catalogación y almacenamiento en el correspondiente repositorio.
- ✚ Interoperabilidad: poseer capacidad para poder integrarse en estructuras y plataformas diferentes.
- ✚ Portabilidad: que puedan moverse e instalarse en diferentes plataformas de manera transparente, sin cambio alguno en estructura o contenido.
- ✚ Durabilidad: que la información de los objetos se mantenga vigente, sin necesidad de nuevos diseños.
- ✚ Educatividad: con capacidad para generar aprendizaje.
- ✚ Independientes y autónomos: con respecto a los sistemas desde los que fueron creados y con sentido propio.
- ✚ Generatividad: capacidad para construir contenidos, objetos nuevos derivados de él. Capacidad para ser actualizados o modificados, aumentando sus potencialidades a través de la colaboración.
- ✚ Flexibilidad: lograr versatilidad y funcionalidad, con elasticidad para combinarse en diversas áreas del saber.
- ✚ Granularidad: el tamaño de un OA es variable y esto se conoce como granularidad, aunque un OA es “una pieza pequeña” o un recurso “modular” no se puede especificar una dimensión precisa.

Teniendo en cuenta la gran cantidad de definiciones que existen para denominar este tipo de recursos, para los términos de la presente investigación se asume el concepto definido por (García Hernández, 2014), al ser el que más se ajusta al objetivo que se persigue en la investigación, al definir los Objetos de Aprendizaje Interactivos y Experimentales (OA-IE) como un: *“Software de carácter educativo que permite la manipulación de sus parámetros a partir de la interacción de quien lo utiliza y produciendo una retroalimentación. Posee un contenido claramente identificable. Su principal potencial es la reutilización dentro de distintos contextos educativos, además de posibilitar la evaluación automatizada del aprendizaje del contenido.”*

El proceso de aprendizaje basado en OA posibilita la libre exploración y control del propio aprendizaje de cada estudiante, además de estimular a estos a profundizar en los contenidos, al presentar la información de manera interactiva, mediante mapas conceptuales, simulaciones, gráficos, animaciones y otros elementos multimedia, en dependencia del objetivo que se persiga con el OA.

Otras ventajas asociadas al uso de este tipo de recurso es la flexibilidad que permite ser usado en múltiples contextos y reutilizado con mayor facilidad que un material reelaborado para cada nuevo contexto. También facilita la personalización de los contenidos al permitir la recombinación de materiales a la medida de las necesidades formativas del colectivo o del individuo. Otra de las ventajas es que facilita la formación basada en competencias, dado que cada OA responde a un objetivo de aprendizaje concreto, ya sea un conocimiento o habilidad. El incremento del valor del contenido es otra de las ventajas pues la reutilización de un contenido incrementa su valor tanto por el aumento del potencial de su explotación futura como por la compensación del costo de desarrollo (REDAOPA, 2004).

Es por eso que resultan un recurso valioso como apoyo al PEA de cualquier asignatura, y en especial a la MD, la cual resulta de difícil comprensión para los estudiantes, al contener temas complejos, y a la que se le da un papel importante en la formación de los profesionales de las TIC.

La UCI cuenta con una herramienta para la elaboración de OA, denominada CRODA. CRODA es una herramienta de autor que brinda a las comunidades de profesores la posibilidad de elaborar recursos educativos en forma de OA, en busca de su accesibilidad, durabilidad, interoperabilidad y reusabilidad. Los OA creados desde esta herramienta, después de pasar por un proceso de revisión de su calidad técnica y pedagógica, se publican en el ROA. Estos OA están compuestos por los metadatos y el objeto de información. Los objetos de información pueden ser imágenes, audios, videos, ejercicios de selección simple o múltiple, de completamiento de espacios en blanco, de enlazar columnas, de verdadero o falso, de orden cronológico y búsqueda en el texto.

Haciendo un análisis de las características de los OA que se crean en CRODA, se puede concluir que estos carecen de interactividad y no permiten la experimentación con el contenido. Al explorar los OA publicados en el ROA referentes a la MD, se puede constatar lo antes expuesto. Son OA que presentan el contenido sin ningún elemento que permita la actividad práctica o la evaluación por parte del objeto. No ofrecen por ejemplo, actividades de experimentación que permitan

introducir valores o cambiar parámetros para realizar cálculos poniendo en práctica principios matemáticos.

1.5 Metodologías para el desarrollo de OA.

La calidad y éxito de un software depende en gran medida de la elección de la metodología de desarrollo de software. Estas proporcionan un conjunto de técnicas, procedimientos y herramientas, con soporte documental permitiéndole a los desarrolladores contar con una guía para el desarrollo del software con la mayor eficiencia posible. Estas se clasifican en Metodologías Tradicionales o Robustas y Metodologías Ágiles.

En cualquier definición de objeto de aprendizaje se destaca la existencia de un objetivo educativo, el cual debe servir de directriz para el proceso de diseño y desarrollo del objeto.

Como en un objeto de aprendizaje existen elementos instruccionales y computacionales, es necesario vincular los modelos de desarrollo de software con los de diseño instruccional para la obtención de un producto de calidad integral (Pernaletе Chirinos, et al., 2009).

El diseño instruccional se entiende como “un proceso tecnológico que especifica, organiza y desarrolla los distintos elementos de la situación de enseñanza-aprendizaje” (Salinas, et al., 2007). Guardia (Guardia, 2000) utiliza el término diseño formativo en lugar de diseño instruccional, sin embargo al analizar la definición dada se evidencia la equivalencia entre los términos. El diseño formativo se utiliza para describir el proceso en que se analizan las necesidades de aprendizaje y el entorno donde se manifestarán, se definen los objetivos de la formación, se escogen los recursos más adecuados teniendo en cuenta los procesos de aprendizaje, se desarrollan los contenidos y las actividades; y se diseña la evaluación.

En (Guardia, 2000) se plantea que el diseño instruccional debe considerar que: (a) los materiales didácticos deben comprender diversas formas de entregar los contenidos; (b) el estudiante es el centro de atención, por lo que el nivel de interacción y tipo de la misma es un elemento importante; y (c) debe promoverse un rol activo en los aprendices a fin de asegurar la apropiada adquisición de conocimientos y estimular la investigación.

Aunque la base de la metodología para el diseño de OA es el modelo pedagógico, estos se pueden considerar como un producto de software y como tal obedecen a un modelo de ciclo de vida del desarrollo de software. Éste cumple con el objetivo de guiar los pasos del desarrollo. Estos modelos de ciclo de vida de desarrollo de software se basan en la naturaleza del proyecto

y de la aplicación, en los métodos y las herramientas que serán usados en el proyecto, los controles y los productos o servicios desarrollados (Pressman, 2002).

1.5.1 ISD-MeLO (Instructional Systems Development Methodology based on e-Learning Objects).

ISD-MeLO fue propuesta por Lucia Blondet Baruque³ y Rubens Nascimento Melo⁴ para apoyar el proceso de enseñanza-aprendizaje de los estudiantes de maestría, diseñadores y desarrolladores de aplicaciones, bajo el dominio de comunidades virtuales, con usabilidad y sociabilidad para otros investigadores que estén interesados en el tema. Está basada en ADDIE (*Analysis, Design, Development, Implementation, Evaluation*), proceso de diseño instruccional iterativo. Es una metodología enfocada al diseño y desarrollo de contenido educativo a ser entregado a través de la web. Permite incorporar mediante un enfoque sistemático principios de diferentes Teorías de Aprendizaje y tiene además un enfoque ecléctico desde la perspectiva de dichas teorías, lo cual le proporciona flexibilidad para aplicarse en una amplia diversidad de situaciones de aprendizaje.

ISD-MeLO contempla que el ensamblaje de los OA sea realizado por el diseñador instruccional en base a su criterio y experiencia (Blondet Baruque, et al., 2004). Incluye 5 fases: Análisis, Diseño, Desarrollo, Implementación y Evaluación.

Tabla 1: Fases de ISD-MeLO

| Fases | Tareas |
|----------|---|
| Análisis | Análisis del perfil del aprendiz Análisis del problema Búsqueda de OA que se ajusten a los objetivos educativos Análisis del entorno Mantenimiento de los metadatos |
| Diseño | Análisis de la tarea Análisis de contenido Identificar la estructura de los objetos de aprendizaje Establecer la secuencia de la instrucción |

³Técnica de base de datos de la Pontificia Universidad Católica de Rio de Janeiro (PUC).

⁴Ingeniero electrónico y doctor en Ciencias. Es uno de los investigadores pioneros en el campo de la Base de Datos y Computación Gráfica en Brasil. Trabaja actualmente en la Pontificia Universidad Católica de Rio de Janeiro (PUC).

| | |
|----------------|---|
| | <ul style="list-style-type: none"> Categorizar los objetos de aprendizaje Especificar los objetos de aprendizaje Mantenimiento de los metadatos en el diseño del contenido Modelar al usuario para diseñar la interfaz de los objetos de aprendizaje Análisis de la tarea del usuario Encontrar una metáfora Diseño del “look” de la interfaz Diseño del “feel” de la interfaz Desarrollar prototipo y evaluar Mantenimiento de los metadatos en el diseño de la interfaz |
| Desarrollo | <ul style="list-style-type: none"> Búsqueda de objetos de aprendizaje en ambientes de bases de datos o en la web Construir los objetos de aprendizaje Efectuar el control de la calidad Almacenar los objetos de aprendizaje en un ambiente de base de datos Mantenimiento de los metadatos |
| Implementación | <ul style="list-style-type: none"> Seleccionar la estrategia para integrar los objetos de aprendizaje en un producto Seleccionar el modo de entrega más adecuado Crear un plan de gestión Ejecutar el producto con la estrategia de entrega seleccionada Hacer seguimiento del progreso |
| Evaluación | <ul style="list-style-type: none"> Llevar a cabo una evaluación formativa Llevar a cabo una evaluación sumativa |

1.5.2 Programación Extrema (*Extreme Programming, XP*)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para

enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Beck, 2000).

Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Utiliza historias de usuario como técnica para especificar los requisitos de software, su tratamiento es muy dinámico y flexible, cada una es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas pocas semanas. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Esta metodología se caracteriza por: Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras. Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera es más importante que la posible pérdida de productividad inmediata, el código es revisado y discutido mientras se escribe. Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo. Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes. Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo. Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados. Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La Programación Extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo (Figueroa, et al., 2008.).

Luego de analizar las características de las metodologías antes expuestas se llega a las siguientes consideraciones:

La metodología ISD-MeLO es una metodología ampliamente utilizada para el desarrollo de OA. Permite el desarrollo de estos de manera organizada según las tareas que contempla en cada fase, logrando que estos se realicen con la calidad deseada, incorpora mediante un enfoque

sistemático principios de diferentes Teorías de Aprendizaje, se enfoca en el diseño y desarrollo de contenido educativo a ser entregado a través de la Web. Su gran inconveniente es que no contempla al OA como un software.

XP es una metodología ágil apropiada para proyectos pequeños, que busca simplificar el proceso de desarrollo de software y reducir los costos, es más fácil de implementar y aprender, presenta un diseño evolutivo que hace que no se le dé importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento. En XP se realizan entregas continuas y discretas que permiten evaluar el sistema durante todo su proceso de construcción.

El uso de estas metodologías por sí solas para el desarrollo de OA-IE no es factible para cumplir los objetivos de esta investigación. Por lo que se considera el uso de un híbrido entre estas, como se propone en (García Hernández, et al., 2013). Al ser XP una metodología fácil de implementar, con una documentación ligera, flexible y eficiente es conveniente su utilización para guiar el proceso de desarrollo, al ser los OA-IE pequeños proyectos de desarrollo de software auto contenidos, que deben realizarse en un corto plazo de tiempo. Por otra parte la metodología ISD-MeLO puede ser muy útil para guiar el proceso de desarrollo de los OA-IE desde el punto de vista pedagógico, al ser flexible para aplicarse en diversas situaciones de aprendizaje.

1.6 Herramientas y lenguaje de desarrollo de OA.

Algunas de las herramientas más utilizadas para el desarrollo de OA son Macromedia Flash, ToolBook, Macromedia Authorware, NeoBook, eXe Learning. Todas estas herramientas se limitan al desarrollo de multimedias, animaciones flash, libros animados, presentaciones animadas, entre otros.

Existen otras herramientas denominadas de autor, las cuales constituyen aplicaciones para el desarrollo de software que facilitan a diseñadores instruccionales, educadores, maestros y aprendices, el diseño de cursos interactivos, ambientes de aprendizaje y OA, sin el conocimiento de lenguajes de programación (Dabbagh, 2001).

Son aplicaciones que disminuyen el esfuerzo a realizar por los profesores, maestros, educadores, ofreciéndoles indicios, guías, elementos predefinidos, ayudas y una interfaz amigable para crear materiales educativos, cursos digitales y OA, cargando con la responsabilidad en los aspectos mecánicos o la tarea, guiando al autor y ofreciéndole elementos predefinidos que puede

relacionar conjuntamente para satisfacer una necesidad particular (Montero O’Farrill, et al., 2008). Entre estas herramientas están eXe Learning, Autore, SoftChalk, GLO Maker.

Cuando se le considera a los OA como un producto de software, es necesario para su desarrollo el uso de herramientas más robustas como son los Entornos de Desarrollo Integrado (en inglés *Integrated Development Enviroment* o IDE).

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

1.6.1 Netbeans 7.4

Es un entorno de desarrollo integrado de código abierto (*Open Source*) y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma JAVA. Entre las características que proporciona tenemos: desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux, Add-ons para desarrollo Móvil, desarrollo Web gráfico, integración con SOA (*Services Oriented Architecture*), optimización de aplicaciones y desarrollo con C y C++, cliente CVS integrado y crecimiento de plataforma por medio de plugins. Entre los plugins que existen se tienen los siguientes: herramientas java que sirven para la mejora de desarrollo de aplicaciones, herramientas de modelado UML, herramientas XML, Struts, JSF, EJB, Web Services (Oracle Corporation, 2013).

1.6.2 Lenguaje de programación JAVA

Java es un lenguaje de programación orientado a objetos, creado por Sun Microsystems⁵ para poder funcionar en distintos tipos de procesadores. Su sintaxis es muy parecida a la de C o C++, e incorpora como propias algunas características que en otros lenguajes son extensiones: gestión de hilos, ejecución remota, entre otras. El código Java, una vez compilado, puede llevarse sin modificación alguna sobre cualquier máquina, y ejecutarlo. Esto se debe a que el código se ejecuta sobre una máquina hipotética o virtual, la Máquina Virtual de Java (en inglés *Java Virtual Machine* o *JVM*), que se encarga de interpretar el código (ficheros compilados .class) y convertirlo a código particular de la CPU que se esté utilizando (siempre que se soporte dicha máquina

⁵<http://www.oracle.com/us/sun/index.htm>

virtual). Algunos de los elementos que consiguen que Java sea diferente son los applets: aplicaciones pequeñas, dinámicas, seguras, multiplataforma, activas y en red (Dpto CCIA, 2005).

Los applet Java son programas incrustados en otras aplicaciones, normalmente una página Web que se muestra en un navegador. Existe gran cantidad de OA desarrollados como applet java. En el sitio web⁶ del profesor alemán Walter Fendt se pueden encontrar gran cantidad de applet sobre Matemáticas, Física y Astronomía. También en el sitio web⁷ del Departamento de Matemática Aplicada de España se pueden encontrar gran cantidad de estos sobre MD, Cálculo Infinitesimal, Sistemas Dinámicos, entre otros.

Al analizar las características que se pueden lograr con los applets, el nivel de experimentación con los contenidos que presentan y la interactividad que brindan los applets explorados, es factible la utilización del lenguaje java para el desarrollo de los OA-IE que se persigue como objetivo de la presente investigación, en forma de applet y utilizando como entorno de desarrollo Netbeans 7.4.

Consideraciones generales del capítulo

A partir del estudio y profundización en el objeto de estudio y el campo de acción definidos para la presente investigación, a través del análisis de los conceptos, la revisión documental y la aplicación de los métodos de investigación, se puede llegar a las siguientes conclusiones:

- ✚ La MD es una asignatura que contribuye en gran medida a la formación del ingeniero en Ciencias Informáticas, debido a que permite crear y desarrollar el pensamiento lógico. Se vincula estrechamente con otras materias que se estudian como parte del plan de estudio de la carrera de Ingeniería en Ciencias Informáticas, lo cual hace esencial su estudio antes de adentrarse en otras.
- ✚ La incorporación de las TIC en el proceso de enseñanza-aprendizaje permite al estudiante la realización interactiva de simulaciones. Puede repetirse una actividad o experimento tantas veces como se desee, cambiando parámetros y condiciones.
- ✚ El proceso de aprendizaje con el uso de OA posibilita la libre exploración y control del propio aprendizaje de cada estudiante. Los OA representan un recurso didáctico que

⁶<http://www.walter-fendt.de/>

⁷<http://neumann.dma.fi.upm.es/>

contribuyen al PEA en la Educación Superior por sus capacidades de flexibilidad y utilización en diferentes escenarios educativos.

- ✚ La utilización de un híbrido entre la metodología XP e ISD-MeLO para guiar el proceso de desarrollo de los OA-IE es factible para cumplir los objetivos de esta investigación, al ser estos pequeños proyectos de desarrollo de software que deben ser entregados en un corto período de tiempo y de contenido pedagógico.
- ✚ Al considerar los OA-IE como un producto de software se necesita para su desarrollo de herramientas robustas como son los Entornos de Desarrollo Integrado, siendo el uso del NetBeans 7.4 y el lenguaje de programación Java los considerados para lograr el objetivo de la presente investigación.

Capítulo 2: Análisis y Diseño del Sistema de objetos de aprendizaje

El presente capítulo tiene como objetivo presentar los resultados obtenidos en las fases de Análisis y Diseño según se propone en la metodología seleccionada. Se seleccionan los contenidos de Matemática Discreta a incluir en el sistema, se describe la estructura que tendrán los OA-IE de acuerdo a criterios pedagógicos, se especifican los requisitos funcionales y no funcionales, y se describen los artefactos generados necesarios para dar solución al problema planteado.

2.1 Análisis

En esta fase se realiza un estudio de los contenidos que contendrán los OA-IE, teniendo en cuenta criterios pedagógicos. Se estudia el análisis del perfil del aprendiz (García Hernández, y otros, 2013).

2.1.1 Selección de los contenidos para desarrollar los nuevos OA-IE.

En la selección de los contenidos que conformarán los OA-IE, para constituir finalmente el sistema que se desea, se tomó como punto de partida los criterios de evaluación de profesores que imparten la asignatura de Matemática Discreta. El proceso de evaluación parte de una encuesta presentada en (García Hernández, 2014), los aspectos contenidos en la encuesta quedan reflejados en la siguiente tabla. A los aspectos se les otorga un valor numérico en la encuesta realizada acorde a su influencia en el coeficiente de factibilidad (K_f) propuesto (Ver tabla 3).

Tabla 2. Grado de influencia de los aspectos a evaluar.

| No. | Aspectos para evaluar los contenidos | Alto | Medio | Bajo |
|-----|---|------|-------|------|
| 1 | Complejidad del contenido | 0,05 | 0,04 | 0,03 |
| 2 | Posibilidad de representar gráficamente | 0,30 | 0,20 | 0,10 |
| 3 | Posibilidad de experimentar | 0,30 | 0,20 | 0,10 |
| 4 | Grado de dificultad de asimilación | 0,20 | 0,10 | 0,05 |
| 5 | Grado de dificultad de impartición | 0,05 | 0,04 | 0,03 |
| 6 | Resultados de las evaluaciones | 0,03 | 0,05 | 0,10 |

Se encuesta a los profesores en los aspectos 1), 2), 3), 5) y 6). A los estudiantes se les encuesta en los aspectos 2), 3) y 4). Los resultados de esta encuesta son los que determinan el coeficiente

de factibilidad (K_f). El K_f es obtenido al sumar los valores de cada uno de los aspectos. Este coeficiente indicará la factibilidad de elaborar un OA-IE para el contenido evaluado:

Tabla 3. Coeficiente de factibilidad

| Coeficiente de factibilidad | Decisión |
|------------------------------------|---------------|
| Bajo ($K_f < 0,4$) | No factible |
| Medio ($0,4 \leq K_f < 0,7$) | Poco factible |
| Alto ($0,7 \leq K_f < 0,9$) | Factible |
| Muy Alto ($0,9 \leq K_f \leq 1$) | Muy Factible |

Se aplicó la encuesta propuesta a un grupo de diez (10) profesores de MD y a un total de diez (10) estudiantes que ya habían cursado la asignatura, escogidos todos de manera aleatoria. Se hizo referencia a los indicadores siguientes: grado de complejidad (GC), posibilidad de representarse gráficamente (PGR), posibilidad de experimentación (PE), grado de dificultad en la asimilación (GDA), grado de dificultad en la impartición (GDI) y resultados en evaluaciones (RE). El procesamiento de las encuestas arrojó la información que se muestra en la tabla siguiente.

Tabla 4. Resultados de la evaluación de los contenidos de MD.

| Tema | GC | PRG | PE | GDA | GDI | RE | Total | Evaluación |
|---------------------------|------|------|------|------|------|------|-------------|----------------------|
| Teoría de Conjuntos | 0.04 | 0.30 | 0.30 | 0.20 | 0.04 | 0.05 | 0.93 | Muy Factible |
| Relaciones Binarias | 0.03 | 0.20 | 0.20 | 0.05 | 0.03 | 0.03 | 0.54 | Poco Factible |
| Lógica | 0.05 | 0.10 | 0.20 | 0.20 | 0.05 | 0.05 | 0.65 | Poco Factible |
| Máquina de Turing | 0.05 | 0.30 | 0.30 | 0.20 | 0.05 | 0.10 | 1.00 | Muy Factible |
| Teoría Combinatoria | 0.05 | 0.20 | 0.30 | 0.20 | 0.05 | 0.10 | 0.90 | Muy Factible |
| Relaciones de recurrencia | 0.04 | 0.30 | 0.20 | 0.20 | 0.04 | 0.05 | 0.83 | Factible |
| Teoría de grafos | 0.04 | 0.3 | 0.3 | 0.1 | 0.04 | 0.05 | 0.83 | Factible |

A partir de este proceso de evaluación y de los OA desarrollados en (Herrera Chica, 2013), se seleccionan como contenidos a incluir en los nuevos, para luego ser integrados en un sistema junto a los anteriormente mencionados:

Teoría Combinatoria: Este contenido presenta un elevado grado de complejidad, puede ser representado gráficamente, además de presentar la posibilidad de experimentación. Es de difícil asimilación y un elevado grado de dificultad de impartición. Además en el EVA de la asignatura solo aparece el basamento teórico de este contenido y enlaces a centros de aprendizaje en línea, a los cuales la mayoría de los estudiantes no podrá acceder al no tener en el primer año de la carrera cuenta de internet, limitando la posibilidad de experimentación por parte de los estudiantes. No existen tampoco OA en el ROA de la UCI relacionados con esta temática.

Relaciones de Recurrencia: Este contenido presenta un alto grado de complejidad, tiene la posibilidad de ser representado gráficamente y permite la experimentación. Presenta un elevado grado de dificultad de impartición y de asimilación por parte de los estudiantes. Al igual que el anterior, su asimilación por parte de los estudiantes es limitada por la falta de recursos que apoyen su proceso de enseñanza-aprendizaje.

En general la constitución de un Sistema de Objetos de Aprendizaje Interactivos y Experimentales que incluya los contenidos anteriormente mencionados, además de los desarrollados en (Herrera Chica, 2013), contribuiría en gran medida al proceso de enseñanza-aprendizaje de la MD en la UCI, al brindar la posibilidad a los estudiantes de interactuar y experimentar con los contenidos, ver sus representaciones gráficamente, introducir valores y comprobar sus resultados. Además, ser evaluados por el propio sistema de acuerdo a su desempeño, permite al estudiante valorar en qué medida está avanzando en el proceso de aprendizaje y en qué medida debe esforzarse más para aprender.

2.2 Diseño

En esta fase se definen la unidad de información, la unidad didáctica y la unidad de evaluación y retroalimentación. Se propone un prototipo de interfaz de usuario donde exista espacio para la teoría, la experimentación y la evaluación. Se definen los Requisitos Funcionales (RF) y los No Funcionales (NRF) (García Hernández, y otros, 2013).

La unidad de información estará conformada por los elementos teóricos del contenido tratado (definiciones, operaciones, ejemplos resueltos, entre otros). Esta unidad sirve de base teórica al estudiante para su posterior interacción y experimentación con el contenido. Está conformada por contenidos multimedia individuales (texto e imagen). El diseño de esta unidad puede ser común para todos los OA-IE, a pesar de que los contenidos y temas varíen, la manera de presentar la información, de navegar en ella, así como la forma de generar los contenidos visuales se puede

realizar de manera similar en todos, este elemento potencia la reutilización y disminuye el tiempo de desarrollo.

La unidad didáctica está orientada a la experimentación con el contenido, de manera que el estudiante pueda ejercitar, introducir valores y comprobar la respuesta dada a una interrogante planteada, que le permita reflexionar y llegar a conclusiones acerca de lo aprendido con la teoría. Estará orientada a un estilo de aprendizaje concreto.

La unidad de evaluación y retroalimentación permite comprobar lo aprendido mediante diferentes estrategias. Se incluirán en el objeto diferentes actividades que servirán de evaluación, en las que el estudiante podrá comprobar si aprendió o no, además de brindar la posibilidad de que él mismo pueda crear sus ejercicios de evaluación.

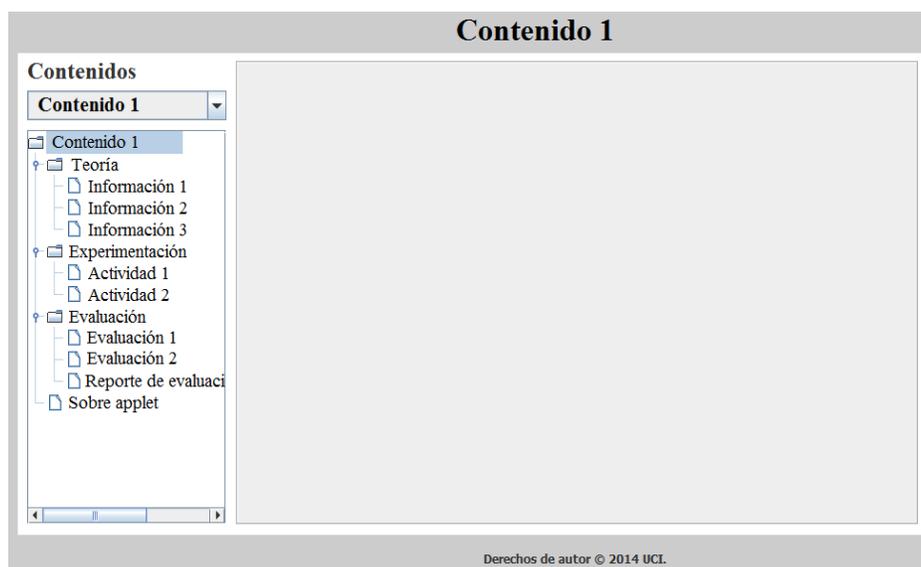
Luego de analizar todos estos elementos se decide modificar el diseño de interfaz de los OA desarrollados en (Herrera Chica, 2013), para lograr integrar en un solo sistema todos los contenidos seleccionados tanto en esta investigación como en aquella. Este nuevo diseño permite la adición de nuevos contenidos de una manera más factible.

2.2.1 Prototipo de interfaz del Sistema de Objetos de Aprendizaje

El prototipo está conformado por un applet java, que será el que contendrá todos los contenidos seleccionados y permitirá agregar nuevos en dependencia de las necesidades de aprendizaje que surjan. Cada OA-IE que integra el sistema estará conformado por una unidad de información, una unidad didáctica y una unidad de evaluación y retroalimentación. El diseño se presenta amigable al usuario y de fácil navegación por los contenidos, presentando textos que indican en todo momento en qué sección del sistema se encuentra.

El usuario selecciona en un menú desplegable que se titula “Contenidos” el contenido que desea estudiar e inmediatamente en el menú de la izquierda aparecen las unidades definidas para los objetos, además de presentar una opción para visualizar información referente a la aplicación. En el panel central aparecerá cada una de las interfaces visuales asociadas a las diferentes opciones del menú de la izquierda. A continuación se muestra el prototipo general de interfaz para el sistema.

Figura 1: Prototipo de interfaz del Sistema de OA-IE.



Unidad de información (UI): Mediante el menú “Teoría” se controla lo referente a la información teórica que se presenta acerca del contenido que se selecciona en el menú “Contenidos”. Para cada una de las opciones de este menú se carga a manera de diapositivas la información definida para esa opción, permitiendo al usuario el control sobre la presentación de las mismas, a través de una barra de controles.

Unidad didáctica (UD): Mediante el menú “Experimentación” se puede acceder a las diferentes simulaciones y actividades de experimentación propuestas. Esta sección permite la reflexión, la interacción y la experimentación con el contenido. Para cada uno de los elementos del menú se definen actividades que complementan los conocimientos teóricos.

Unidad de evaluación y retroalimentación (UER): Mediante el menú “Evaluación” se puede acceder a actividades de control del conocimiento adquirido mediante diferentes criterios de evaluación. A través de este menú se puede acceder además a diferentes interfaces que permiten la creación de ejercicios que serán incluidos como parte de las evaluaciones que vienen por defecto.

2.2.2 OA-IE de Teoría Combinatoria

En la **UD** del OA-IE de Teoría Combinatoria aparecen una serie de actividades que permiten poner en práctica diferentes principios de la Teoría Combinatoria. En la sección **Permutación** de la sección **Experimentación** se puede ejercitar el cálculo de la permutación de r objetos de una colección de n de ellos. El usuario introduce los valores de estos parámetros y el posible resultado

de la operación realizada por él mismo, para comprobar si lo hizo de manera correcta, o puede calcular directamente el resultado. En la sección **Combinación** se puede ejercitar el cálculo de la combinación de r objetos de una colección de n de ellos. El usuario introduce los valores de estos parámetros y el posible resultado de la operación realizada por él mismo, para comprobar si lo hizo de manera correcta, o puede calcular directamente el resultado. A continuación se presentan las vistas de las interfaces que permiten la realización de estas operaciones.

Figura 2: Interfaz Permutación.

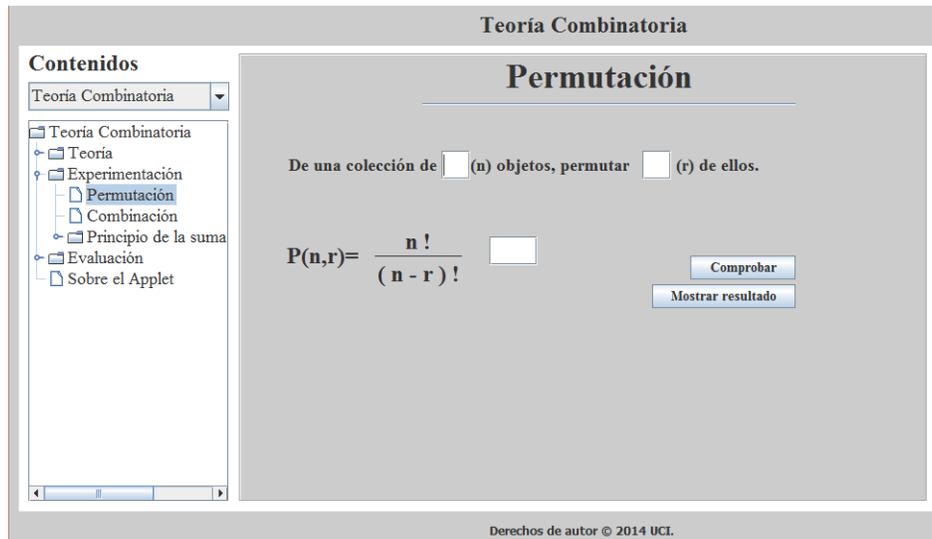
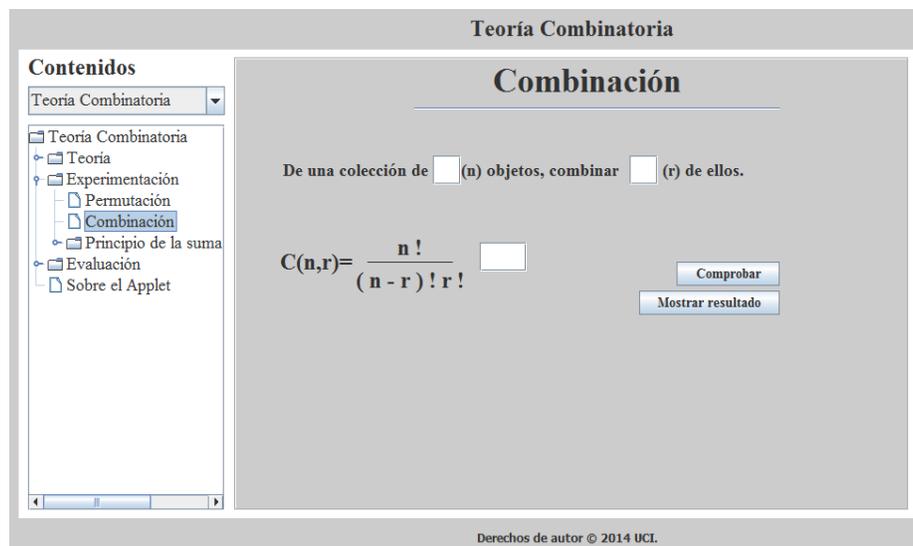


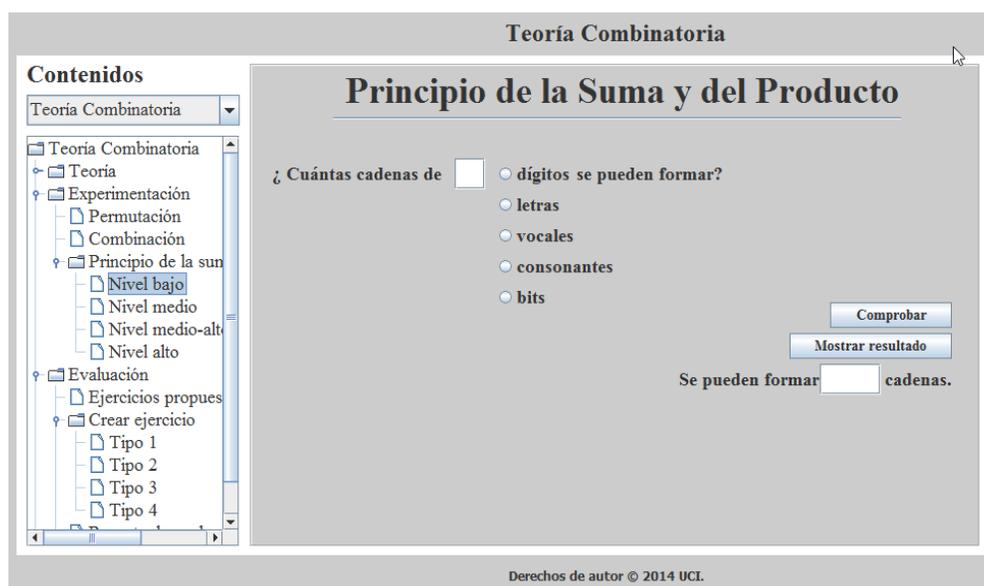
Figura 3: Interfaz Combinación.



El **Principio del Producto** constituye uno de los principios básicos de la combinatoria y enuncia que si una tarea se realiza en dos etapas, donde la primera se puede realizar de m formas

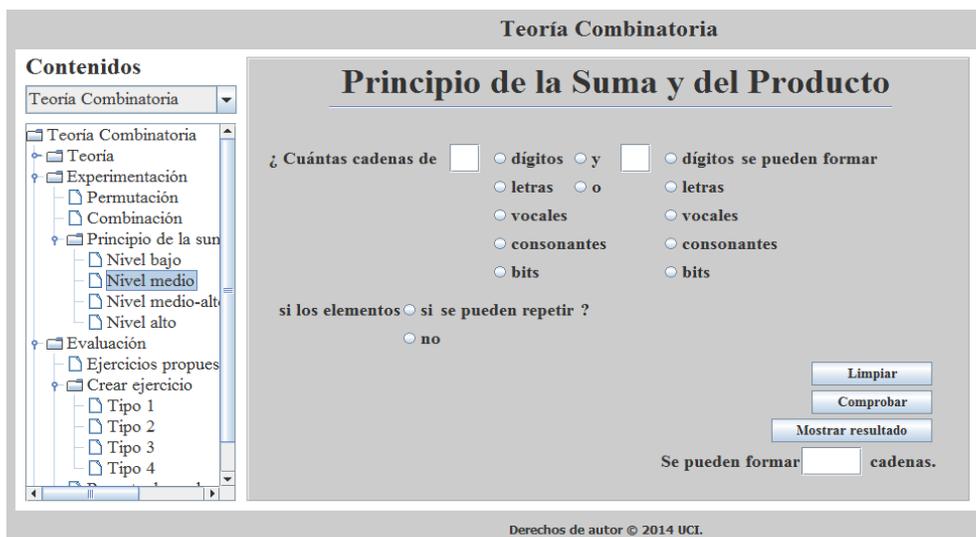
posibles y, si para cada una de ellas la segunda se puede realizar de n distintas formas, entonces la tarea completa se puede realizar de $m*n$ formas posibles. Este principio puede ser ejercitado en la sección **Principio de la Suma y del Producto**. En la interfaz que se presenta a continuación el usuario introduce en un campo de texto la cantidad de caracteres que desea tenga una cadena y a su vez selecciona los elementos que conforman esa cadena, además introduce el resultado de la operación y comprueba su respuesta o puede hacer que el propio sistema calcule el resultado. En esta sección se ejercita este principio en su forma más elemental.

Figura 4: Interfaz Principio de la Suma y del Producto con un nivel bajo.



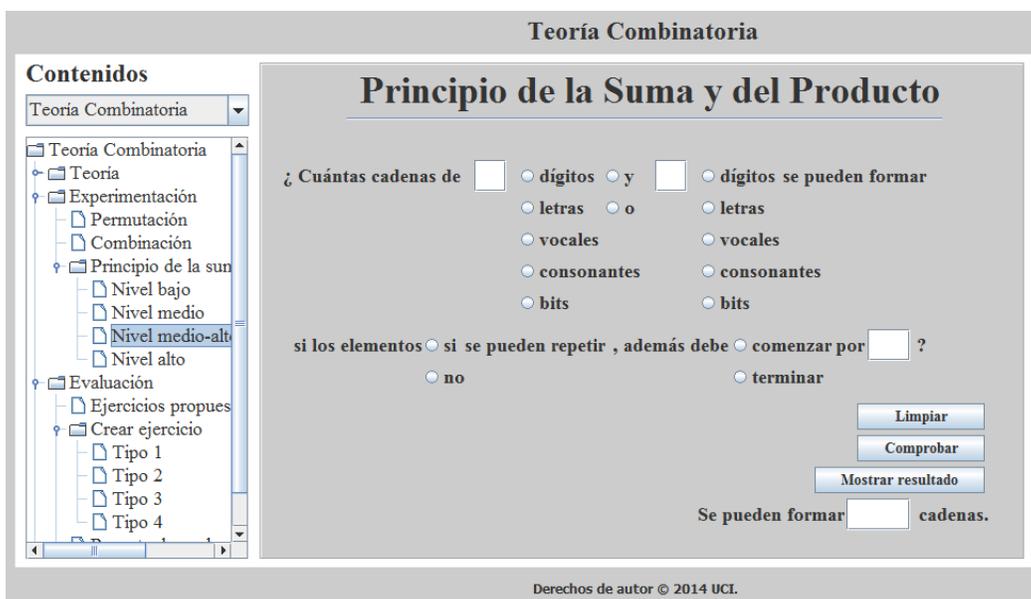
El **Principio de la suma** es uno de los principios fundamentales de conteo y enuncia que si una tarea se puede realizar de m formas distintas, mientras que una segunda se puede efectuar de n formas distintas, y no se pueden realizar las dos tareas simultáneamente, entonces se puede lograr realizar cualquiera de ellas de $m+n$ maneras distintas. En la sección **Principio de la Suma y del Producto** se puede ejercitar además este principio combinado con el del producto. A medida que el usuario va avanzando por las diferentes subsecciones de esta sección la complejidad de la práctica aumenta. A continuación se muestran cada una de las interfaces asociadas a estas subsecciones.

Figura 5: Interfaz Principio de la Suma y del Producto con un nivel medio.



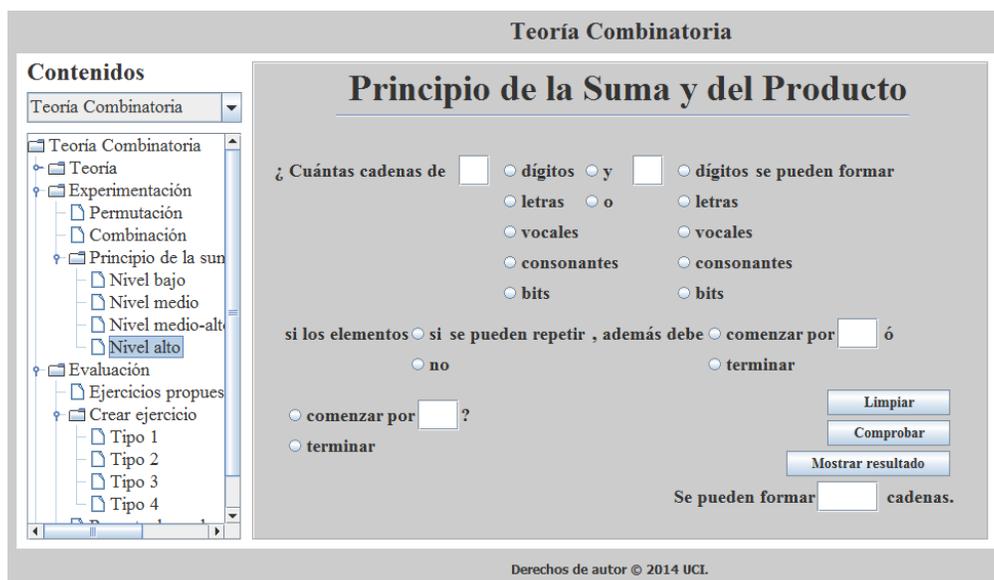
En este caso se calcula la cantidad de cadenas de dos tipos de elementos que se pueden formar, estos elementos deberán ser seleccionados por el usuario. Igualmente se deben introducir la cantidad de elementos de cada tipo que tendrá la cadena, y si los elementos se pueden o no repetir. En dependencia de si se selecciona que la cadena estará constituida por elementos tipo1 y/o elementos tipo2, se deberá aplicar el principio del producto o el de la suma en cada caso. Esto permite al usuario experimentar con estos principios, al tener que identificar mediante qué principio hacer el cálculo y la posibilidad de comprobar si la respuesta que da es correcta o no.

Figura 6: Interfaz Principio de la Suma y del Producto con un nivel medio - alto.



En este caso se eleva el nivel de complejidad de la práctica, básicamente se hace el mismo cálculo que en el caso anterior, pero ahora el usuario debe tener en cuenta que la cadena debe *comenzar/terminar* con un determinado elemento que él debe introducir.

Figura 7: Interfaz Principio de la Suma y del Producto con un nivel alto.



Esta interfaz viene a poner en práctica estos principios de todas las posibles formas. Además se agrega otro importante principio de la combinatoria, el **Principio de Inclusión-Exclusión**, que enuncia que cuando dos tareas se pueden realizar simultáneamente no se puede aplicar el principio de la suma, pues en ese caso se estarían contando dos veces dichas tareas, para contar de forma correcta las maneras de realizar una de las dos tareas, se suman las maneras de realizar cada una de ellas y de aquí se restan las formas de realizar las dos tareas simultáneamente. El usuario deberá dominar este principio pues es, en solo dos casos de los posibles que se pueden formar mediante la interfaz que se evidencia este principio, es en el caso de seleccionar **terminar en o comenzar por** y el caso **comenzar por o terminar en**. Los restantes casos se calculan mediante los principios de suma y producto.

En la **UER** se presentan una serie de actividades que permiten controlar el conocimiento adquirido en las unidades anteriores, y el sistema será capaz de evaluar las respuestas dadas por el usuario. Mediante la sección **Ejercicios propuestos** de la sección **Evaluación** se puede acceder a estas actividades. El usuario selecciona desde un menú desplegable la evaluación que desea realizar, introduce su respuesta y se evalúa. Para visualizar el resultado de su evaluación deberá ir a la sección **Reporte de evaluaciones**. Esta interfaz contiene evaluaciones predeterminadas, pero se podrán agregar otras mediante la sección **Crear ejercicio**.

Figura 8: Interfaz Ejercicios propuestos.

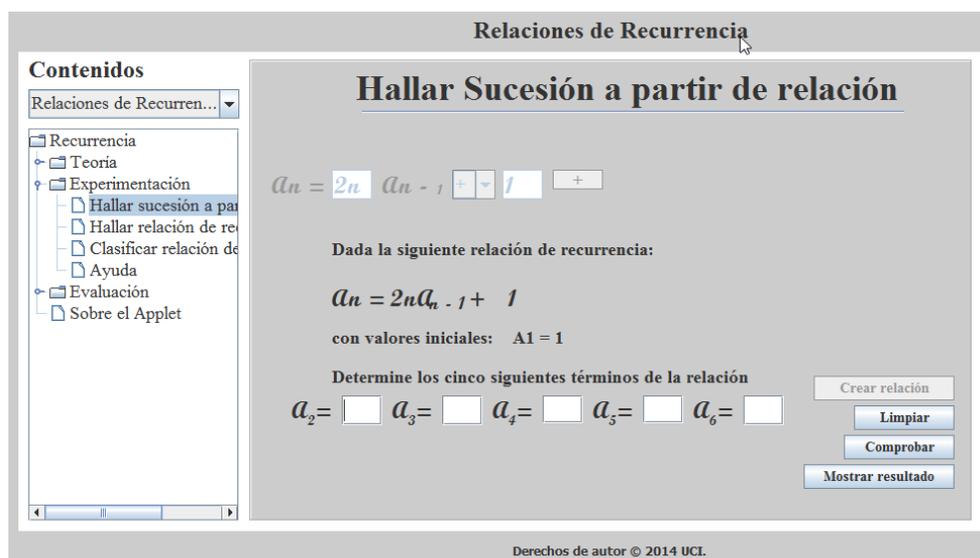


Mediante la sección **Crear ejercicios** se pueden crear ejercicios que servirán de comprobación del conocimiento, en esta sección se crean cuatro tipos de ejercicios y al ser creados satisfactoriamente se añaden a la lista de evaluaciones de la interfaz **Ejercicios propuestos**. Las interfaces relativas a estos tipos de ejercicios se pueden encontrar en el Anexo 1.

2.2.3 OA-IE de Relaciones de Recurrencia

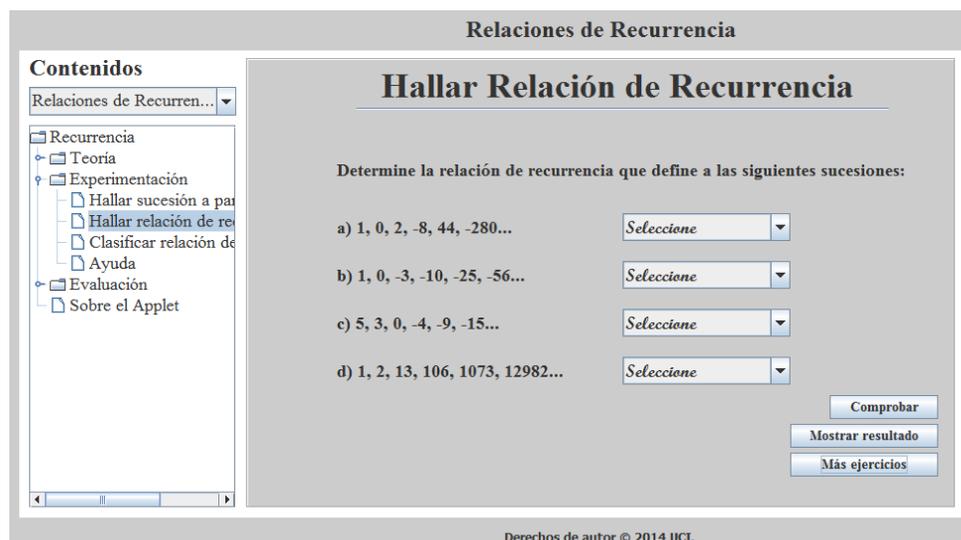
En la **UD** de Relaciones de Recurrencia aparecen una serie de ejercicios que servirán al usuario para ejercitar lo aprendido en la UI del OA-IE en cuestión. En la sección **Hallar sucesión a partir de relación** el usuario puede calcular los valores de una sucesión a partir de una relación de recurrencia. El sistema permite que el usuario cree la relación de recurrencia y en dependencia del grado de la relación creada el sistema brinda los valores iniciales necesarios para hallar los cinco siguientes términos de la sucesión solución de esa relación. El usuario puede comprobar si los valores entrados por él son correctos o visualizar directamente estos valores. A continuación se muestra la vista asociada a esta sección.

Figura 9: Interfaz Resolver relación de recurrencia.



En la sección **Hallar Relación de Recurrencia** el usuario puede determinar la relación de recurrencia que define una sucesión dada. El sistema ofrece una serie de ejercicios de este tipo, el usuario selecciona de una lista desplegable la relación que define la sucesión en cuestión y al finalizar puede comprobar si su respuesta es correcta o mostrar el resultado de manera automática por el sistema. A continuación se muestra la vista asociada a esta sección.

Figura 10: Interfaz Hallar relación de recurrencia.



La **UER** del OA-IE de Relaciones de Recurrencia funciona de manera similar a la de Teoría Combinatoria. Se presentan una serie de actividades en la sección **Evaluación**, el usuario las realiza y puede visualizar su evaluación en **Reporte de evaluaciones**, donde se muestran las

actividades realizadas clasificadas en correctas o incorrectas, en el caso de las incorrectas se muestra la respuesta dada por el usuario y la respuesta correcta. En la sección **Crear ejercicio** se pueden añadir nuevos ejercicios de evaluación.

2.2.4 Requisitos Funcionales

Los requisitos funcionales definen el comportamiento interno del software, es decir, las funcionalidades específicas de la aplicación. A continuación se presentan los definidos para la inclusión de los contenidos seleccionados.

RF1 Realizar permutación.

RF2 Realizar combinación.

RF3 Experimentar con los principios básicos de Teoría Combinatoria con un nivel de complejidad bajo.

RF4 Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios de complejidad media.

RF5 Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad medio alta.

RF6 Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta.

RF7 Evaluar contenido de Teoría Combinatoria.

RF8 Crear ejercicio de Teoría Combinatoria tipo 1.

RF9 Crear ejercicio de Teoría Combinatoria tipo 2.

RF10 Crear ejercicio de Teoría Combinatoria tipo 3.

RF11 Crear ejercicio de Teoría Combinatoria tipo 4.

RF12 Mostrar reporte de evaluaciones de Teoría Combinatoria.

RF13 Hallar sucesión a partir de una relación de recurrencia.

RF14 Hallar relación de recurrencia a partir de una sucesión.

RF15 Clasificar relaciones de recurrencia.

RF16 Evaluar identificar sucesiones numéricas y relaciones de recurrencia.

RF17 Evaluar clasificación de relaciones de recurrencia.

RF18 Crear ejercicios de relaciones de recurrencia.

RF19 Mostrar reporte de evaluaciones de Relaciones de Recurrencia.

RF20 Mostrar ayuda para el trabajo con el objeto de aprendizaje de Relaciones de Recurrencia.

RF21 Integrar los OA-IE en un mismo sistema.

2.2.5 Requisitos no funcionales

Los requisitos no funcionales son aquellos que especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, facilidad de mantenimiento, extensibilidad y fiabilidad. Describen aspectos del sistema que son visibles por el usuario, que no incluyen una relación directa con el comportamiento funcional del sistema (Jacobson, y otros, 2000). A continuación se listan los identificados para los OA-IE a desarrollar.

Usabilidad

El sistema podrá ser utilizado por cualquier usuario con conocimientos básicos de informática y el trabajo en ambiente web. Su objetivo es asegurar un proceso de aprendizaje satisfactorio, por tanto, no solo incluye el contenido, sino que también guía el proceso de aprendizaje.

Apariencia o interfaz externa

El sistema debe contar con una interfaz amigable, fácil de usar por usuarios con conocimientos básicos en informática. Debe presentar el mismo formato en los textos.

Portabilidad

El sistema debe ser multiplataforma y disponible en plataforma web.

Disponibilidad

El sistema debe estar disponible para ser accedido en todo momento por cualquier usuario.

Restricciones en el diseño y la implementación

Se utiliza un híbrido entre las metodologías XP e ISD-MeLO, puesto que los OA-IE son considerados pequeños proyectos de desarrollo de software que deben ser entregados en un corto plazo de tiempo y de contenido pedagógico.

2.2.6 Historias de usuario

Uno de los artefactos generados por la metodología XP son las historias de usuario (HU). Las HU son la técnica utilizada para representar y especificar los requisitos del software. Se trata de un conjunto de tablas en las cuales el cliente describe brevemente las características que el sistema debe poseer. Su tamaño no excede de unas pocas líneas de texto. El tratamiento de las HU es muy dinámico y flexible. Tienen el mismo propósito que los casos de uso, las escriben los propios clientes tal y como ven ellos las necesidades del sistema. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración (Canós, y otros, 2003).

Para el desarrollo de esta investigación se obtuvieron un total de 21 historias de usuario. A continuación se muestran 3 de estas y en el Anexo 2 se pueden encontrar otras 7.

Tabla 5: HU_6 Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta.

| Historia de Usuario | |
|---|--|
| Número: HU_6 | Nombre: Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Alta | Tiempo estimado: 1 |
| Riesgo en desarrollo: Medio | Tiempo real: 1 |
| Descripción: Permite realizar cálculos aplicando los Principios de la Suma, del Producto y de Inclusión-Exclusión, de la forma “¿Cuántas cadenas de ‘n1’ elementos tipo1 y/o ‘n2’ elementos tipo2 se pueden formar si los elementos si/no se pueden repetir, además debe comenzar/terminar con ‘c1’ o comenzar/terminar con ‘c2’?”. El usuario deberá introducir la cantidad de elementos, el caracter de inicio/fin de la cadena y seleccionar los tipos de elementos que la conformarán para proceder al cálculo. Brinda la posibilidad de introducir un posible resultado y comprobar si este es correcto o mostrarlo directamente, además de la posibilidad de restablecer los datos introducidos y/o seleccionados. | |
| Observaciones: En caso de no introducir las cantidades de elementos, el caracter de inicio/fin, ni seleccionar los elementos, se mostrará un mensaje de error al hacer clic en cualquiera de los botones presentes. En caso contrario al hacer clic en el botón “Comprobar” se mostrará un mensaje advirtiendo si el resultado dado por el usuario es correcto o no. | |

Tabla 6: HU_7 Evaluar contenido de Teoría Combinatoria.

| Historia de Usuario | |
|---|--|
| Número: HU_7 | Nombre: Evaluar contenido de Teoría Combinatoria. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Alta | Tiempo estimado: 0.4 |

| | |
|--|-------------------------|
| Riesgo en desarrollo: Medio | Tiempo real: 0.4 |
| Descripción: Permite realizar ejercicios sobre el contenido de Teoría Combinatoria, para evaluar los conocimientos del usuario. El usuario selecciona el ejercicio que desea realizar, da una respuesta y crea su evaluación. | |
| Observaciones: Al hacer clic en el botón “ <i>Terminar</i> ” se muestra un mensaje que confirma que se ha guardado la evaluación. En caso de no seleccionar ningún ejercicio o no introducir el resultado, se mostrará un mensaje advirtiendo el error. | |

Tabla 7: HU_13 Hallar una sucesión a partir de una relación de recurrencia.

| Historia de Usuario | |
|---|--|
| Número: HU_13 | Nombre: Hallar sucesión a partir de una relación de recurrencia. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 2 |
| Prioridad: Alta | Tiempo estimado: 0.5 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.5 |
| Descripción: Permite hallar una sucesión a partir de una relación de recurrencia. Brinda la posibilidad al usuario de crear la relación de recurrencia, el sistema crea aleatoriamente los valores iniciales de la sucesión y el usuario puede entrar los cinco siguientes valores de la sucesión. El sistema permite comprobar si la sucesión entrada por el usuario es correcta o calcular directamente estos valores. | |
| Observaciones: Al hacer clic en el botón “Crear relación” el sistema crea la relación y se muestran los valores iniciales en dependencia del grado de la relación creada. Si el usuario no entra los valores de la sucesión al hacer clic en el botón “Comprobar” se muestra un mensaje de error, sino muestra un mensaje advirtiendo si la respuesta dada es correcta o no. Al hacer clic en el botón “Mostrar resultado” se muestran los valores de la sucesión. Estos botones aparecen deshabilitados al cargar la vista y se activan al crear la relación de recurrencia, y a su vez se desactiva el botón “Crear relación”. Para deshacer y comenzar otra vez la práctica se debe presionar el botón “Limpiar”. | |

2.2.7 Iteraciones

Luego de tener definidas las historias de usuario, es necesario crear un plan de iteraciones, que no es más que una planificación donde los desarrolladores y clientes establecen los tiempos de

implementación ideales de las historias de usuario, la prioridad de estas y cuáles serán implementadas en cada versión del programa (Beck, 2000).

Se definieron tres iteraciones para el desarrollo del sistema. En la primera iteración se implementan las funcionalidades del OA-IE de Teoría Combinatoria, en la segunda las funcionalidades del OA-IE de Relaciones de Recurrencia y en la tercera se realiza la integración de los OA-IE en un único sistema.

Plan de duración de las iteraciones.

Se creó el plan de duración de cada iteración para el desarrollo del sistema, con el objetivo de mostrar la duración de cada iteración y el orden en que serían implementadas las historias de usuario.

Tabla 8: Plan de duración de las iteraciones.

| Nombre de la HU | Iteración | Duración |
|---|-----------|----------|
| Realizar permutación. | 1 | 2 días |
| Realizar combinación. | 1 | 2 día |
| Experimentar con los principios básicos de Teoría Combinatoria con un nivel de complejidad bajo. | 1 | 3 días |
| Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios de complejidad media. | 1 | 6 días |
| Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad medio alta. | 1 | 8 días |
| Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta. | 1 | 8 días |
| Evaluar contenido de Teoría Combinatoria. | 1 | 4 días |
| Mostrar reporte de evaluaciones. | 1 | 2 días |
| Crear ejercicio de Teoría Combinatoria tipo 1. | 1 | 4 días |
| Crear ejercicio de Teoría Combinatoria tipo 2. | 1 | 4 días |
| Crear ejercicio de Teoría Combinatoria tipo 3. | 1 | 4 días |
| Crear ejercicio de Teoría Combinatoria tipo 4. | 1 | 4 días |
| Hallar sucesión a partir de una relación de recurrencia. | 2 | 5 días |
| Hallar relación de recurrencia a partir de una sucesión. | 2 | 5 días |

| | | |
|--|---|--------|
| Clasificar relaciones de recurrencia. | 2 | 2 días |
| Evaluar identificar sucesiones numéricas y relaciones de recurrencia. | 2 | 5 días |
| Evaluar clasificación de relaciones de recurrencia. | 2 | 2 días |
| Crear ejercicios de relaciones de recurrencia. | 2 | 2 días |
| Mostrar reporte de evaluaciones de Relaciones de Recurrencia. | 2 | 3 días |
| Mostrar ayuda para el trabajo con el objeto de aprendizaje de Relaciones de Recurrencia. | 2 | 1 día |
| Integrar los OA-IE en un solo sistema. | 3 | 6 días |

2.2.8 Tarjetas CRC

Para el diseño de aplicaciones informáticas, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC (del inglés *Class-Responsibility-Collaboration*) (Beck, 2000). El uso de las tarjetas CRC permite al programador centrarse en el desarrollo orientado a objetos, rompiendo con la clásica programación procedural. Se dividen en tres secciones: el nombre de la clase, sus responsabilidades y sus colaboradores. Una clase describe cualquier objeto o evento, mediante los atributos y los métodos, las responsabilidades son las tareas que realizan o los métodos correspondientes a la clase y los colaboradores son las demás clases con las que trabaja conjuntamente para cumplir con sus responsabilidades. A continuación se muestran algunas de las tarjetas definidas para la implementación de las historias de usuario. El resto se pueden encontrar en el [Anexo 3](#).

Tabla 9: Tarjeta CRC Panel_Combinatoria.

| Panel_Combinatoria | |
|---|---|
| Responsabilidades | Colaboraciones |
| Clase encargada de representar cada una de las vistas asociadas a las opciones que aparecen en las unidades definidas para el OA-IE de Teoría Combinatoria. | Principal, Ventana, Evaluacion_Comb, Ejercicio_Comb |

Tabla 10: Tarjeta CRC Evaluacion_Comb.

| Evaluacion_Comb | |
|--|--------------------|
| Responsabilidades | Colaboraciones |
| Clase modelo encargada de representar evaluaciones de Teoría Combinatoria. | Panel_Combinatoria |

Tabla 11: Tarjeta CRC Panel_Recurrencia.

| Panel_Recurrencia | |
|---|---|
| Responsabilidades | Colaboraciones |
| Clase encargada de representar cada una de las vistas asociadas a las opciones que aparecen en las unidades definidas para el OA-IE de Relaciones de Recurrencia. | Principal, Ventana, Evaluacion_Recurrencia, Ejercicio_RelacionR |

2.3 Arquitectura de Software

Según el estándar IEEE (*Institute of Electrical and Electronics Engineers*): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

2.3.1 Patrón Modelo Vista Controlador

Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

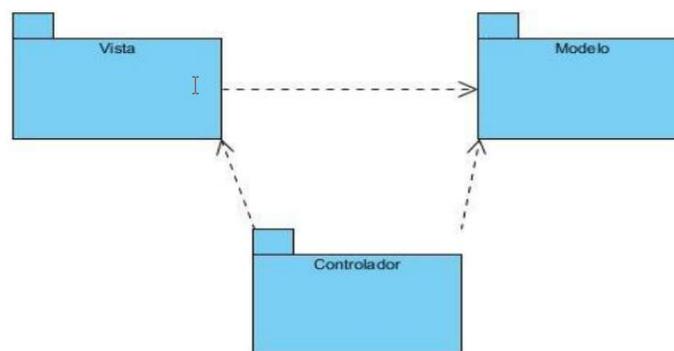
Modelo: Es la representación de los datos que maneja la aplicación. No tiene conocimiento específico del Controlador o de las Vistas, ni siquiera contiene referencias a ellos (GONZÁLEZ, Y., 2012). Ejemplo de estas clases son Evaluacion_Comb, Ejercicio_Comb, Ejercicio_RelacionR y Evaluacion_Recurrencia.

Vista: Es el elemento que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador. Es responsable de recibir datos procesados por el

controlador o del modelo y mostrarlos al usuario (GONZÁLEZ, Y., 2012). Dentro de estas clases están Panel_Combinatoria, Panel_Conjunto, Panel_Grafo, Panel_Turing y Panel_Recurrencia.

Controlador: Es el elemento que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Maneja y entrega las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario (GONZÁLEZ, Y., 2012). Ejemplo de estas clases lo constituyen Generadora_Conjuntos, Generadora_Cadena, Generadora_Grafos y Principal.

Figura 11: Diagrama de paquetes.



2.4 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. Los patrones documentan y explican problemas de diseño, y luego discuten una buena solución a dicho problema (Campo, 2009).

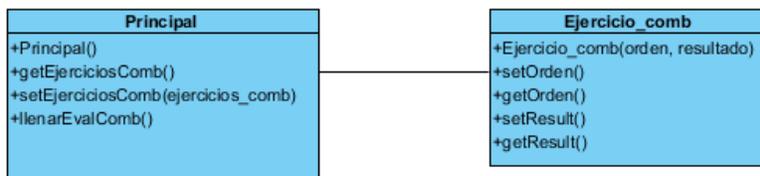
2.4.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns*. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Grosso, 2011).

Experto en información: Asignar una responsabilidad al experto en información; la clase que tiene la información necesaria para llevar a cabo la responsabilidad. Este patrón se ve evidenciado por ejemplo en la clase Grafo, ya que los métodos que contiene solo pueden ser implementados en esta clase, al ser esta la que contiene la información necesaria para llevar a cabo responsabilidades tales como *insertarVertice()*, *eliminarVertice()*, *insertarArco()*, *existeVertice()*, por mencionar algunos.

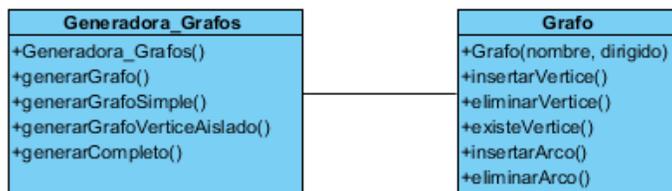
Creador: Ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Tiene la información necesaria para realizar la creación del objeto que es una de las actividades más comunes en un sistema orientado a objetos. Almacena o maneja varias instancias de la clase. En la clase Principal en el método *llenarEvalComb()* se crea una instancia de la clase Ejercicio_Comb, este es uno de los ejemplos donde se evidencia este patrón.

Figura 12: Diagrama de clase del sistema propuesto donde se evidencia el uso del patrón Creador.



Controlador: Asigna la responsabilidad de gestionar un mensaje de un evento del sistema a una clase controladora. Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Responde a la pregunta de ¿Quién debería encargarse de atender o controlar un evento del sistema?

Figura 13: Diagrama de clase del sistema propuesto donde se evidencia el uso del patrón Controlador.



Consideraciones generales del capítulo

En este capítulo se analizaron elementos referentes al contenido y diseño del Sistema de OA-IE que se construye, por lo que se puede concluir que:

- ✚ En la selección de los contenidos de Teoría Combinatoria y Relaciones de Recurrencia para desarrollar los nuevos OA-IE se tuvo en cuenta los criterios de evaluación de profesores que imparten la asignatura de MD en la UCI. Estos contenidos presentan un elevado grado de complejidad, pueden ser representados gráficamente, además de presentar la posibilidad de experimentación, son de difícil asimilación y un elevado grado de dificultad de impartición, y su asimilación por parte de los estudiantes es limitada por la falta de recursos que apoyen su proceso de enseñanza-aprendizaje.
- ✚ El diseño que se realiza, permite la integración de los OA desarrollados en (Herrera Chica, 2013) con los elaborados en esta investigación. Cada OA-IE estará conformado por tres unidades: la unidad de información, la unidad didáctica y la unidad de evaluación y retroalimentación; lo que posibilita la interacción con los contenidos y la retroalimentación de lo aprendido.
- ✚ La identificación de 21 requisitos funcionales y 5 no funcionales, unido a la división del trabajo de implementación de las historias de usuario en tres iteraciones permite definir las funcionalidades finales y organizar el trabajo de desarrollo del Sistema de OA-IE.
- ✚ La utilización del patrón Modelo Vista Controlador permite separar los datos, la interfaz de usuario y la lógica de control en tres componentes distintos, logrando una mayor organización entre los componentes del Sistema de OA-IE que se desarrolla. El uso de los patrones de asignación de responsabilidades permite lograr un diseño flexible y reutilizable.

Capítulo 3: Implementación y validación del Sistema de Objetos de Aprendizaje

La implementación y validación de un producto informático son fundamentales en su proceso de desarrollo. Como resultado de estas fases se obtiene un producto funcional, con el que debe garantizarse el cumplimiento de los requisitos identificados con la calidad requerida. El presente capítulo persigue como objetivo presentar los resultados obtenidos en estas fases, se exponen las tareas de ingeniería definidas para cada historia de usuario con vistas a su implementación, se presenta además el estándar de codificación utilizado, así como una descripción de las pruebas realizadas a la solución y los resultados de su ejecución.

3.1 Tareas de Ingeniería

Las tareas de ingeniería surgen de la división de las historias de usuario en tareas más sencillas para facilitar el trabajo de implementación de las mismas (Canós, y otros, 2003). A continuación se describen las tareas definidas para la historia de usuario HU_13 Hallar sucesión a partir de una relación de recurrencia. En el [Anexo 4](#) se describen otras cinco tareas y en el [Anexo 5](#) aparece un diagrama de Gantt generado con el objetivo de facilitar la planificación de la implementación de las tareas por cada historia de usuario.

Tabla 12: TI_1 HU_13 Definir y crear los componentes visuales necesarios para hallar una sucesión a partir de una relación de recurrencia.

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 1 | Número de Historia de Usuario: 13 |
| Nombre Tarea: Definir y crear los componentes visuales necesarios para hallar una sucesión a partir de una relación de recurrencia. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 14/4/2014 | Fecha Fin: 18/4/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se realiza el diseño e implementación de la vista que permita hallar una sucesión a partir de una relación de recurrencia. | |

Tabla 13: TI_2 HU_13 Validación de los datos.

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 2 | Número de Historia de Usuario: 13 |
| Nombre Tarea: Validación de los datos. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 19/4/2014 | Fecha Fin: 19/4/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se realiza la implementación de los métodos necesarios que validen que los datos entrados por el usuario son correctos y se corresponden con los tipos de datos a entrar. | |

3.2 Estándares de codificación

Los estándares de codificación, también conocidos como estilos de programación o convenciones de código, son convenios para escribir código fuente en ciertos lenguajes de programación. Permiten que el código en consecuencia sea mantenible y que todos los participantes lo puedan entender en un menor tiempo (Beck, 2000).

La metodología XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo y que facilite la recodificación. A continuación se describen cada uno de los estilos utilizados en la implementación y ejemplos de su utilización.

3.2.1 Indentación

Se deben emplear cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco o tabuladores) no se especifica. Los tabuladores deben ser exactamente cada 8 espacios (no 4). Se deben garantizar los siguientes elementos:

- ✚ Longitud de la línea: líneas de no más de 80 caracteres.
- ✚ Rompiendo líneas: cuando una expresión tenga más de 80 caracteres, romperla de acuerdo con los siguientes principios:
 - ✓ Romper después de una coma.
 - ✓ Romper antes de un operador.
 - ✓ Preferir roturas de alto nivel (más a la derecha que el «padre») que de bajo nivel (más a la izquierda que el «padre»).

- ✓ Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

Figura 14: Ejemplo correcto de rotura de línea en expresiones aritméticas.

```
} else if (cant_var == 4) {  
    cant_restar = Double.valueOf(jTextField12.getText())  
                + Double.valueOf(jTextField11.getText()) +  
    Double.valueOf(jTextField13.getText()) + Double.valueOf(jTextField14.getText());
```

Figura 15: Ejemplo correcto de rotura de líneas para sentencias.

```
if (cant_var_ej2 == 2) {  
    if (jTextField87.getText().equals("") || jTextField90.getText().equals("")  
        || jTextField89.getText().equals("") || jTextField94.getText().equals("")  
        || jTextField20.getText().equals("") || jTextField95.getText().equals("")  
        || jTextField1.getText().equals("")) {  
        flag_text = false;
```

3.2.2 Comentarios

Los comentarios se usan para dar descripciones de código y facilitar información adicional que no es legible en el código mismo (Sun Microsystems Inc, 1999). Los comentarios deben contener sólo información que es relevante para la lectura y entendimiento del programa.

- ✚ **Comentarios de bloque:** Los comentarios de bloque se usan para dar descripciones de archivos, métodos, estructuras de datos y algoritmos. Los comentarios de bloque se podrán usar al comienzo de cada archivo o antes de cada método. También se pueden usar en otros lugares, tales como el interior de los métodos. Estos últimos deben ser indentados al mismo nivel que el código que describen.
- ✚ **Comentarios de fin de línea:** El delimitador de comentario (//) puede convertir en comentario una línea completa o una parte de una línea. No debe ser usado para hacer comentarios de varias líneas consecutivas; sin embargo, puede usarse en líneas consecutivas para comentar secciones de código.

Figura 16: Ejemplo correcto de comentarios.

```
/*  
 *Este método crea un ejercicio de tipo 2  
 */  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    double result = 0;  
    String orden = "";  
    double cant_restar = 0;    //cantidad que se le resta al resultado
```

3.2.3 Declaraciones

Una declaración es la asignación de un identificador a una variable, clase o método, para utilizarlas en el transcurso de un programa (Sun Microsystems Inc, 1999). Las declaraciones deben cumplir con las siguientes normas:

- ✚ Hacer una declaración por línea, ya que facilita los comentarios.
- ✚ No poner diferentes tipos en una misma línea.
- ✚ Inicialización: Inicializar las variables locales donde se declaran. La única razón para no inicializar una variable donde se declara es si el valor inicial depende de cálculos posteriores.
- ✚ Colocación: Situar las declaraciones solo al principio de los bloques y no esperar a su primer uso para declararlas. Esto evita confusiones y no limita la portabilidad del código dentro de su ámbito. Igualmente evitar las declaraciones locales que oculten declaraciones de niveles superiores, no declarar el mismo nombre de variable en un bloque interno.
- ✚ Declaraciones de clases e interfaces: No dejar espacio en blanco entre el nombre de un método y el paréntesis que abre su lista de parámetros. La llave de apertura aparece al final de la misma línea de la sentencia de declaración. La llave de cierre siempre empieza en una nueva línea indentada para ajustarse a su sentencia de apertura, excepto cuando no existen sentencias entre ambas; en esos casos debe aparecer inmediatamente después de la de apertura. Los métodos se separan con una línea en blanco.

Figura 17: Ejemplo correcto de declaraciones.

```
private void jTextField4KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char a = evt.getKeyChar();  
  
    if ((int) a >= 48 && (int) a <= 57) {  
        if (jTextField4.getText().length() == 2) {  
            evt.consume();  
            jTextField4.setText(String.valueOf(a));  
        }  
    } else {  
        evt.consume();  
        jTextField4.setText("");  
    }  
}
```

3.2.4 Sentencias

Las sentencias realizadas deben cumplir con las siguientes normas:

- ✚ Cada línea debe contener solo una sentencia simple.
- ✚ En las sentencias IF-ELSE debe haber siempre un espacio en blanco entre la palabra reservada IF y el paréntesis de la condición, entre el paréntesis que cierra dicha condición y las llaves de la sentencia, así como entre la palabra reservada ELSE y las llaves que le anteceden y prosiguen. Las sentencias a ejecutar deben aparecer en la línea siguiente, cumpliendo las reglas de indentación según las normas definidas en el epígrafe 3.1.1.

3.2.5 Espacio en blanco

Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas.

Se debe usar siempre una línea en blanco en las siguientes circunstancias:

- ✚ Entre métodos.
- ✚ Entre las variables locales de un método y su primera sentencia.
- ✚ Entre las distintas secciones lógicas de un método para facilitar la lectura.

Se deben usar espacios en blanco en las siguientes circunstancias:

- ✚ Una palabra clave del lenguaje seguida por un paréntesis debe separarse por un espacio.
- ✚ Debe aparecer un espacio en blanco después de cada coma en las listas de argumentos.
- ✚ Todos los operadores binarios excepto «.» se deben separar de sus operandos con espacios en blanco. Los espacios en blanco no deben separar los operadores unarios, incremento («++») y decremento («--») de sus operandos.
- ✚ Las expresiones en una sentencia *for* se deben separar con espacios en blanco.
- ✚ Los «*Cast*» deben ir seguidos de un espacio en blanco.

Figura 18: Ejemplo correcto de aplicación de los espacios en blanco.

```
if (items.Longitud() != 0) {  
    for (int i = 0; i < items.Longitud(); i++) {  
        jComboBox2.addItem(items.Obtener(i));  
    }  
}
```

3.2.6 Convenciones de nomenclatura

Las convenciones de nomenclatura hacen que el código sea más inteligible al hacerlo más fácil de leer. También pueden dar información sobre la función de un identificador (Sun Microsystems Inc, 1999). Se deben cumplir las siguientes normas:

- ✚ Los nombres de las clases deben ser sustantivos. Cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas y estarán separadas por guiones bajos.

Figura 19: Ejemplo correcto de convención de nombres de clases.

```
public class Panel_Combinatoria extends javax.swing.JPanel {
```

- ✚ Los métodos deben ser verbos. Cuando son compuestos tendrán la primera letra en minúscula y la primera letra de las siguientes palabras que lo forma en mayúscula.

Figura 20: Ejemplo correcto de convención de nombres de métodos.

```
public void cargarPermutacion() {
```

- ✚ Todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) estarán separadas por guiones bajo. Los nombres de las variables deben ser cortos pero significativos. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador ocasional su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.

Figura 21: Ejemplo correcto de convención de nombres de variables.

```
ListaSE<Evaluacion_Comb> lista_evaluaciones = new ListaSE<>();
```

3.3 Validación de los Objetos de Aprendizaje

La fase de pruebas añade valor al producto que se desarrolla, todos los programas tienen errores y la fase de pruebas los descubre; ese es el valor que añade. El objetivo específico de la fase de pruebas es encontrar la mayor cantidad de errores posibles (Mañas, 1994).

La metodología XP propone que se realicen tantas pruebas como sea posible y de esta forma permitir aumentar la calidad del sistema y disminuir el número de errores. Para realizar las pruebas a los OA-IE se tomaron en cuenta las pruebas unitarias, encargadas de verificar el código y diseñada por los programadores y las pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de cada iteración se consiguió la funcionalidad requerida.

3.3.1 Pruebas unitarias. Pruebas de caja blanca

El método de caja blanca denominado a veces prueba de caja de cristal, requiere del conocimiento de la estructura interna del programa. Permite comprobar los caminos lógicos del software y examinar el estado del programa en varios puntos para determinar si el estado es real y coincide con el esperado (Departamento de Ingeniería y Gestión de Software, 2009).

Mediante los métodos de prueba de caja blanca, se pueden obtener casos de prueba que:

- ✚ Garanticen que se ejecutan por lo menos una vez todos los caminos independientes de cada módulo.
- ✚ Se utilicen todas las decisiones lógicas en sus vertientes verdadera y falsa.
- ✚ Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- ✚ Se utilicen las estructuras internas de datos para asegurar su validez. (Esquivel, 2009)

Dentro de las técnicas de diseño de pruebas de caja blanca está la prueba del camino básico, que permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Para aplicar esta técnica, el flujo de control es representado mediante un Grafo de Flujo. Cada círculo denominado **nodo del grafo** de flujo, representa una o más sentencias procedimentales. Un solo nodo puede corresponder a una secuencia de cuadros de proceso y a un rombo de decisión. Las flechas del grafo denominadas **aristas o enlaces**, representan flujo de control. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental. Las áreas delimitadas por aristas y nodos se denominan **regiones**. Cuando se contabilizan las regiones se incluye el área exterior del grafo, contando como otra región más.

La complejidad ciclomática forma parte del método del camino básico. Define el número de caminos independientes del conjunto básico de un programa y da un límite inferior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

La **complejidad ciclomática V(G)** se puede calcular de tres formas:

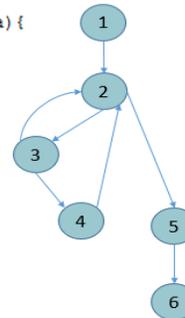
- ✚ El número de regiones del grafo de flujo coincide con la complejidad ciclomática, es decir $V(G) = R$.
- ✚ Aristas - Nodos + 2, es decir $V(G) = A - N + 2$.

✚ Nodos Predicado + 1 (un nodo predicado es el que representa una condicional *if* o *case*, es decir, que de él salen varios caminos), $V(G) = P + 1$.

A continuación se muestra el método `compCharCadena()`, al cual se le aplicó el método del camino básico. El análisis de la funcionalidad arrojó como resultado el grafo de ejecución al cual se le aplicó la técnica de la complejidad ciclomática.

Figura 22: Método `compCharCadena()` y grafo de ejecución resultante.

```
public boolean compCharCadena(char caracter, String cadena){
    boolean resp = false;
    int cont = 0;
    for (int i = 0; i < cadena.length(); i ++){
        if (cadena.toCharArray()[i] == caracter) {
            cont ++;
        }
    }
    resp = cont > 0;
    return resp;
}
```



Cálculo de la complejidad ciclomática $V(G)$:

$$V(G) = R \quad V(G) = 3$$

$$V(G) = (A - N) + 2 \quad V(G) = 7 - 6 + 2V(G) = 3$$

$$V(G) = P + 1 \quad V(G) = 2 + 1 \quad V(G) = 3$$

El cálculo de la complejidad ciclomática del código es 3, lo que significa que existen tres posibles caminos por los que circula el código, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

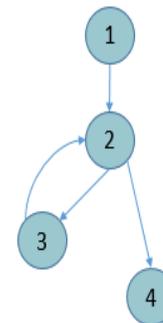
$$C1 = 1-2-5-6 \quad C2 = 1-2-3-2-5-6 \quad C3 = 1-2-3-4-2-5-6$$

En la figura 24 se muestra la aplicación del camino básico al método `desordenarLista()`:

Figura 23: Método `desordenarLista()` y grafo de ejecución resultante.

```
public ListaSE<Integer> desordenarLista(ListaSE<Integer> lista){
    ListaSE<Integer> desordenados_aux = new ListaSE<>();
    ListaSE<Integer> aleatorios_aux = new ListaSE<>();
    aleatorios_aux = generarAleatorios(lista.Longitud());

    for (int i = 0; i < 4; i++) {
        desordenados_aux.Adicionar(lista.Obtener(aleatorios_aux.Obtener(i)));
    }
    return desordenados_aux;
}
```



Cálculo de la complejidad ciclomática $V(G)$:

$$V(G) = R \quad V(G) = 2$$

$$V(G) = (A - N) + 2 \quad V(G) = 4 - 4 + 2 \quad V(G) = 2$$

$$V(G) = P + 1 \quad V(G) = 1 + 1 \quad V(G) = 2$$

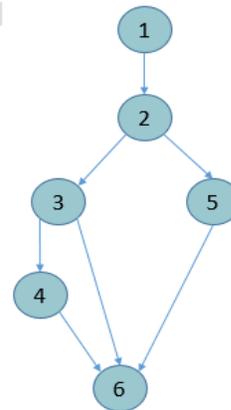
El cálculo de la complejidad ciclomática del código es 2, lo que significa que existen dos posibles caminos por los que circula el código:

$$C1 = 1-2-4 \quad C2 = 1-2-3-2-4$$

En la figura 25 se muestra la aplicación del método del camino básico al método `(jTextField8KeyTyped())`:

Figura 24: Método `jTextField8KeyTyped()` y grafo de ejecución resultante.

```
private void jTextField8KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char a = evt.getKeyChar();
    if ((int) a >= 48 && (int) a <= 57 || (int) a == 45) {
        if (jTextField8.getText().length() == 20) {
            evt.consume();
            jTextField8.setText(String.valueOf(a));
        }
    } else {
        evt.consume();
        jTextField8.setText("");
    }
}
```



Cálculo de la complejidad ciclomática $V(G)$:

$$V(G) = R \quad V(G) = 3$$

$$V(G) = (A - N) + 2 \quad V(G) = 7 - 6 + 2 \quad V(G) = 3$$

$$V(G) = P + 1 \quad V(G) = 2 + 1 \quad V(G) = 3$$

El cálculo de la complejidad ciclomática del código es 3, lo que significa que existen tres posibles caminos por los que circula el código:

$$C1 = 1-2-5-6 \quad C2 = 1-2-3-4-6 \quad C3 = 1-2-3-6$$

3.3.2 Pruebas de aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuario. Especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente, y que la aplicación cumpla lo realmente solicitado por el cliente.

La ejecución de las pruebas diseñadas permitió la evaluación de las funcionalidades de los OA-IE desarrollados antes de implantarlos en su entorno real de explotación. Los resultados obtenidos en tres de estas se muestran a continuación, en el Anexo 6 se muestran otras 3:

Tabla 14: Caso de prueba de aceptación de la HU_6.

| Caso de prueba de aceptación | | |
|--|--|-------------------------------|
| Número: 1 | Nombre de historia de Usuario: Experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita realizar cálculos aplicando los principios básicos de la Teoría Combinatoria con una complejidad alta, brindando la posibilidad de comprobar la respuesta ofrecida por el usuario o mostrarla directamente. | | |
| Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar en el menú “<i>Contenidos</i>” el contenido de Teoría Combinatoria. 2. Seleccionar “<i>Nivel alto</i>” del menú inferior izquierdo, siguiendo la ruta <i>Teoría Combinatoria-> Experimentación-> Principio de la suma y del producto-> Nivel alto</i>. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Ingresar la cantidad de elementos que formarán la cadena, así como seleccionar los tipos de elementos y los caracteres de inicio y fin de la cadena. Todos los campos son obligatorios, excepto el campo “ <i>resultado</i> ” para los casos que se desee conocer su valor directamente. | Luego de ingresados los datos, si el proceso ha sido correcto al presionar el botón “ <i>Comprobar</i> ” se muestra un mensaje que advierte si la respuesta es correcta o no y al hacer clic en el botón “ <i>Mostrar resultado</i> ” se muestra el resultado correcto de la operación en el campo “ <i>resultado</i> ”. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| Campos requeridos vacíos y/o no seleccionados. Caracteres | Se debe mostrar un mensaje de error indicando que hay campos | Satisfactorio |

| | | |
|--|--|--|
| alfabéticos en campos que no lo requieran. | vacíos o que ha introducido caracteres no válidos. | |
| Evaluación de la prueba: Satisfactorio. | | |

Tabla 15: Caso de prueba de aceptación de la HU_7.

| Caso de prueba de aceptación | | |
|---|--|-------------------------------|
| Número: 2 | Nombre de historia de Usuario: Evaluar contenido de Teoría Combinatoria. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita evaluar contenido de Teoría Combinatoria mediante la resolución de ejercicios. | | |
| Pasos de ejecución: | | |
| <ol style="list-style-type: none"> 1. Seleccionar en el menú “<i>Contenidos</i>” el contenido de Teoría Combinatoria. 2. Seleccionar “<i>Ejercicios propuestos</i>” del menú inferior izquierdo, siguiendo la ruta <i>Teoría Combinatoria->Evaluación->Ejercicios propuestos</i>. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Seleccionar del menú “ <i>Evaluaciones</i> ” que aparece en la vista, el ejercicio de evaluación a desarrollar. Luego ingresar el resultado del ejercicio y presionar el botón “ <i>Terminar</i> ”. | Luego de seleccionado el ejercicio e introducido su resultado, al presionar el botón “ <i>Terminar</i> ” se muestra un mensaje que confirma que se ha guardado la evaluación satisfactoriamente. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| No haber seleccionado ningún ejercicio. Campo “ <i>resultado</i> ” vacío. | Se debe mostrar un mensaje de error indicando que no se ha seleccionado ningún ejercicio o que hay campos vacíos. | Satisfactorio |
| Evaluación de la prueba: Satisfactorio. | | |

Tabla 16: Caso de prueba de aceptación de la HU_13.

| Caso de prueba de aceptación | | |
|---|---|-------------------------------|
| Número: 3 | Nombre de historia de Usuario: Hallar sucesión a partir de una relación de recurrencia. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita hallar una sucesión a partir de una relación de recurrencia creada por el usuario. | | |
| Pasos de ejecución: | | |
| <ol style="list-style-type: none"> 1. Seleccionar en el menú “<i>Contenidos</i>” el contenido de Relaciones de Recurrencia. 2. Seleccionar “<i>Hallar sucesión a partir de relación</i>” del menú inferior izquierdo, siguiendo la ruta <i>Relaciones de Recurrencia->Experimentación-> Hallar sucesión a partir de relación</i>. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Introducir los coeficientes de los términos de la relación de recurrencia, según su orden. Seleccionar los operadores aritméticos entre cada término. Presionar “ <i>Crear relación</i> ”. Introducir los valores de los términos de la sucesión. Estos campos son obligatorios en el caso que se quiera comprobar si el resultado es correcto. | Al presionar “ <i>Crear relación</i> ” el sistema debe generar y mostrar los valores iniciales de la relación, de acuerdo a su orden. Luego de creada la relación de recurrencia e introducidos los valores de los términos de la sucesión, al presionar “ <i>Comprobar</i> ” se muestra un mensaje que advierte si la sucesión es correcta o no. Al presionar “ <i>Mostrar resultado</i> ”, se muestra la sucesión correcta, resultante de la relación creada. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| Introducir todos los coeficientes de los términos de la relación iguales a cero. No introducir los términos de la | Se debe mostrar un mensaje de error indicando que en ese caso no es una relación de recurrencia. Se debe mostrar un mensaje de error | Satisfactorio |

| | | |
|--|--|--|
| sucesión al presionar "Comprobar". | indicando que se deben introducir los términos de la sucesión. | |
| Evaluación de la prueba: Satisfactorio. | | |

3.3.3 Evaluación de la calidad del sistema desde el punto de vista pedagógico

La evaluación de la calidad es un proceso necesario e importante, sobre todo cuando se trata de OA-IE que serán utilizados como apoyo al PEA. Es un proceso, que a partir de las características que tienen los OA-IE a desarrollar, se puede considerar complejo, por la mezcla de metodologías al verlos en su dualidad como productos pedagógicos y de software (Herrera Chica, 2013).

Se decide utilizar como estrategia de evaluación de la calidad la Guía propuesta por (Toll Palma, 2011) al contener los elementos que permiten evaluar la calidad de los OA-IE.

La guía está dada por tres aspectos (Ver [Anexo 7](#)): el formativo, el de diseño y presentación, y el tecnológico. Definiendo un conjunto de 34 indicadores de evaluación. Se ha considerado que todos los indicadores deben ser de cumplimiento básico para cualquier OA-IE. Para los 34 indicadores agrupados en los tres aspectos se utiliza la escala de Excelente, Bien, Regular y Mal, donde se le otorga una puntuación de entre tres (3) y cero (0) respectivamente. El objetivo fundamental es obtener una evaluación final que permita evaluar el OA-IE según los rangos de la escala definida en Muy Adecuado, Adecuado, Poco Adecuado y No Adecuado, determinando el nivel de calidad alcanzado por el OA-IE.

La evaluación de los OA-IE fue realizada por seis (6) profesores del colectivo de Matemática Discreta. A los evaluadores se les entregó un fichero MS Excel™ con los criterios, se les explicó que significaba cada uno de ellos y la forma en que debían evaluarlos. La evaluación se realizó de forma anónima.

El objetivo fundamental es poder a partir de las puntuaciones otorgadas por los especialistas a los indicadores de la guía, evaluar cada uno de estos indicadores de manera independiente y luego analizar la suma de ellos para otorgar en la escala definida por (García Hernández, 2014), la evaluación final (Ver Tabla 17).

Tabla 17: Escala de nivel de calidad de los OA-IE.

| Rangos de escala | Nivel de calidad alcanzado por el OA-IE |
|------------------|---|
| $E \geq 95$ | Muy Adecuado |
| $85 \leq E < 95$ | Adecuado |
| $65 \leq E < 85$ | Poco Adecuado |
| $E < 65$ | No adecuado |

Luego de consolidar la información de los OA-IE sobre los contenidos de Teoría Combinatoria y de Relaciones de Recurrencia se obtuvieron las siguientes tablas:

Tabla 18: Calificación de evaluadores para el OA-IE de Teoría de Combinatoria.

| Aspecto | Evaluador | | | | | |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| | E1 | E2 | E3 | E4 | E5 | E6 |
| Formativo | 40 | 39 | 40 | 41 | 40 | 39 |
| Diseño | 29 | 30 | 28 | 29 | 29 | 29 |
| Técnico | 28 | 26 | 28 | 29 | 28 | 28 |
| Evaluación | 97 | 95 | 96 | 99 | 97 | 96 |

Tabla 19: Calificación de evaluadores para el OA-IE de Relaciones de Recurrencia.

| Aspecto | Evaluador | | | | | |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| | E1 | E2 | E3 | E4 | E5 | E6 |
| Formativo | 39 | 39 | 40 | 41 | 38 | 38 |
| Diseño | 30 | 30 | 29 | 26 | 29 | 29 |
| Técnico | 26 | 28 | 28 | 29 | 28 | 28 |
| Evaluación | 95 | 97 | 97 | 96 | 95 | 95 |

A partir de estos valores, los seis profesores consideran que ambos tiene categoría de **Muy Adecuado**, resultado que valida la importancia y el valor de los mismos.

Consideraciones generales del capítulo

En este capítulo se abordaron las fases de implementación y validación del Sistema de OA-IE para la MD que se propone como objetivo de la investigación, por lo que se concluye que:

- ✚ La división de las historias de usuario en un total de 43 tareas de ingeniería facilitó el trabajo de implementación de las funcionalidades definidas para el Sistema de OA-IE.
- ✚ El estándar de codificación que se utiliza para el desarrollo del Sistema garantiza que este sea entendible y facilite su recodificación.
- ✚ La realización de pruebas unitarias o de caja blanca, utilizando la técnica del camino básico, permitió verificar que el código funciona de manera correcta. Las pruebas de aceptación realizadas permitieron determinar que cada historia de usuario fue implementada correctamente y se consiguieron todas las funcionalidades definidas.
- ✚ El proceso de evaluación de los OA-IE utilizando la guía de Toll Palma (Toll Palma, 2011), a partir del criterio de profesores de MD arrojó que los OA-IE desarrollados son muy adecuados según el valor obtenido en la escala propuesta en la guía utilizada.

Conclusiones generales

El presente trabajo tuvo como base la investigación y el desarrollo científico encaminados al desarrollo de un Sistema de Objetos de Aprendizaje Interactivos y Experimentales para apoyar el PEA de la MD en la UCI. Se cumplió con el objetivo general a través de las tareas de investigación por lo que se puede arribar a las siguientes conclusiones:

- ✚ La investigación permitió identificar a la MD como contenido esencial en la formación de los profesionales de las Ciencias de la Computación, así como la complejidad de su aprendizaje en la educación superior. Se analizaron las ventajas que ofrece el uso de OA en el PEA de la MD en la UCI y el insuficiente aprovechamiento del uso de las TIC en este proceso.
- ✚ La selección de los contenidos a incluir en los OA-IE se realizó a partir de los desarrollados en (Herrera Chica, 2013), y de una encuesta presentada en (García Hernández, 2014). Este análisis permitió la selección de los contenidos de Teoría Combinatoria y Relaciones de Recurrencia, teniendo en cuenta criterios tales como el grado de complejidad de impartición y asimilación, la posibilidad de representación gráfica y la posibilidad de experimentar.
- ✚ El diseño del Sistema de OA-IE permite la integración de los desarrollados en (Herrera Chica, 2013) con los de la presente investigación, definiendo para cada objeto una Unidad de Información donde se brinda la posibilidad de navegación por la información relacionada con el contenido; una Unidad Didáctica, para la experimentación y una Unidad de Evaluación, para la comprobación de la asimilación de los contenidos.
- ✚ Las funcionalidades de los OA-IE fueron identificadas con la ayuda de un grupo de profesores de MD vinculados a la investigación. Estas funcionalidades fueron traducidas a Historias de Usuario que posteriormente fueron implementadas logrando como resultado un Sistema de OA-IE que posibilita la interactividad y experimentación, además de permitir la evaluación y autoevaluación de lo aprendido.
- ✚ La calidad del Sistema de OA-IE se comprobó a través de pruebas de caja blanca, pruebas de aceptación y mediante la utilización de la guía de evaluación de (Toll Palma, 2011), a partir del criterio de profesores de MD. Se verificó que se cumplieron las funcionalidades definidas y que los OA-IE obtenidos como resultado son muy adecuados desde el punto de vista pedagógico, lo que garantiza la factibilidad de su utilización en el PEA de la MD en la UCI.

Recomendaciones

Independientemente de haberse logrado el objetivo propuesto se recomienda:

- ✚ Aplicar técnicas de inteligencia artificial para poder adaptar el contenido del Sistema de OA-IE a las características y condiciones de los estudiantes que lo utilizan.
- ✚ Implementar las funcionalidades necesarias para que los OA-IE envíen los resultados de las evaluaciones al profesor.

Bibliografía

1. **García-Valcárcel Muñoz-Repiso, A. 2011.** Integración de las TIC en la Docencia Universitaria . [En línea] 2011. http://books.google.com.cu/books/about/Integraci%C3%B3n_de_Las_Tic_en_la_Docencia.
2. **Aguilar Cisneros, Jorge, Muñoz Arteaga, Jaime y Pomares Hernández, Saúl. 2004.** Guías de diseño para el desarrollo de objetos de aprendizaje. 2004.
3. **Alejandro de Alvarado, Almudena Rodríguez, [ed.]. 2006.** *La formación sin distancia: estudio realizado por el grupo de trabajo de "e-Learning" 05 de la Red TTnet España.* s.l. : Servicio Público de Empleo Estatal, 2006.
4. **ÁVILA, D. T. 2009.** Elementos fundamentales sobre la concepción de los Objetos de Aprendizaje en la Universidad de las Ciencias Informáticas. 2009.
5. **Beck, Kent . 2000.** *Extreme Programming Explained. Embrace Change.* [trad.] Addison Wesley. s.l. : Pearson Education, 2000. .
6. **Benítez, Gerardo Meneses. 2007.** Las nuevas tecnologías de la información. *NTIC, INTERACCIÓN Y APRENDIZAJE EN LA UNIVERSIDAD.* 2007.
7. **Blondet Baruque, Lucía y Nascimento Melo, Rubens . 2004.** Applying Learning Theory in the Design of Learning Objects. [En línea] June de 2004. ftp://ftp.inf.puc-rio.br/pub/docs/techreports/04_19_baruque.pdf. 2009.
8. **Campo, G.D. 2009.** Patrones de diseño, Refactorización y Antipatrones. Ventajas y Desventajas de su Utilización en el Software Orientado a Objetos . 2009.
9. **Canavelli, Juan Carlos, de Carrera, Elena T.F. y Gaitán, María Mercedes. 2007.** *Actualizar el Currículo de Matemática. Una Necesidad Perentoria en las Ingenierías.* Universidad Nacional del Centro de la Provincia de Buenos Aires – Facultad de Ciencias Humanas : V Encuentro Nacional y II Latinoamericano LaUniversidad como objeto de investigación, 2007.
10. **Canavelli, Prof Juan Carlos, Gaitán, Ing. María Mercedes y Maris Vaira, Lic. Stella. 2007.** ICME 11 Mexico 2008, 11th International Congress on Mathematical Education. [En línea] 2007. <http://icme11.org>.
11. **Canós, José H, Letelier, Patricio y Penadés, M^a Carmen. 2003.** Método logías Ágiles en el Desarrollo de Software. *No Quality Inside - Cursos.* [En línea] 2003. http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf.
12. **Castañeda Hevia, Angel Emilio . 2009.** El papel de las tecnologías de información y las comunicaciones (TIC) en el proceso de enseñanza aprendizaje a comienzos del siglo XXI. *Fundamentos Didácticos de la Educación Superior Cubana .* La Habana : Felix Varela, 2009.

13. **Colombia Aprende. 2005.** Colombia Aprende. s.l. : Primer Concurso Nacional de Objetos de Aprendizaje, 2005.
14. **Da Vinci, Leonardo. 1735.** *El arte y las matemáticas.* Roma : s.n., 1735.
15. **Dabbagh, N. H. 2001.** Authoring tools and Learning Systems: A Historical Perspective. 2001.
16. **Del Moral Pérez, Maria Esther y Cernea, Doina Ana. 2005.** Diseñando Objetos de Aprendizaje como facilitadores de la construcción del conocimiento. Barcelona, España : II Simposio Pluridisciplinar sobre Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables (SPDECE05), 2005.
17. **Departamento de Ingeniería y Gestión de Software. 2009.** Flujo de trabajo de pruebas. La Habana : UCI, 2009.
18. **1991.** *Diccionario de Santillana de Tecnología Educativa.* Madrid : s.n., 1991.
19. **Dpto CCIA. 2005.** Lenguaje Java y Entorno de Desarrollo. 2005.
20. **Duart Montoliu, J. M y Sangrá Morer, A. 2000.** Formación universitaria por medio de la web: un modelo integrador para el aprendizaje superior. *Aprender en la virtualidad.* s.l. : Barcelona: Gedisa, 2000.
21. **ESCET. 2003.** *Matemática Discreta. Apuntes ESCET.* 2003.
22. **Esquivel, Karina. 2009.** ESTRATEGIAS DE PRUEBA DEL SOFTWARE. 2009.
23. **Fernandez Rincon, Hector H, y otros.** *Objetos de Aprendizaje y la Escritura Universitaria: Una Propuesta Para la Elaboracion de Monografias.*
24. **Ferro Soto, C, Martínez Senra, A. I y Otero Neira, M. C. julio 2009.** Ventajas del uso de las TICs en el proceso de enseñanza aprendizaje desde la óptica de los docentes universitarios españoles. s.l. : EDUTECH, julio 2009.
25. **Figuroa, Roberth G, Solís, Camilo J y Cabrera, Armando A. . 2008. .** Metodologías Tradicionales vs. Metodologías Agiles. *Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.* [En línea] 2008. . <http://tg-tatiana-oquendo.googlecode.com/svn/trunk/articulo-metodologia-de-sw-formato.doc>.
26. **García Hernández, Alién . 2014.** Estrategia metodológica para la elaboración y utilización de objetos de aprendizaje interactivos y experimentales en el proceso de enseñanza-aprendizaje de la Matemática Discreta en la UCI. 2014.
27. **García Hernández, Alién, Verdecia Martínez, Edistio Yoel y Herrea Chica, Isael. 2013.** Pedagogía virtual: Metodología para el desarrollo de Objetos de Aprendizaje interactivos y experimentales. s.l. : Pedagogía 2013, Universidad de Ciencias Informáticas, 2013.

28. **García Hernández, Alién, Verdecia Martínez, Edistio Yoel y Herrera Chica, Isael. 2013.** *Metodología para el desarrollo de Objetos de Aprendizaje interactivos y experimentales.* s.l. : Pedagogía 2013, 2013.
29. **García-Valcárcel Muñoz-Repiso, A. 2009.** Educación y Tecnología. [En línea] 2009. <http://web.usal.es/~anagv/arti1.htm>.
30. **Gisbert, otros y M. 1992.** Technology based traingning. Formador de formadores en la dimensión ocupacional. Tarragona : s.n., 1992.
31. **Gómez, María Elena Pardo, y otros. Febrero 2005.** *Las Tecnologías de la Información y las Comunicaciones en la Dinámica del Proceso Docente Educativvo en la Educación Superior.* Universidad de Oriente. Santiago de Cuba : CiberEduca, Febrero 2005. V Congreso Internacional Virtual de Educación.
32. **GONZÁLEZ, Y. 2012.** Patrón Modelo-Vista-Controlador. *Revista Telemática.* 2012.
33. **Granda Dihigo, Ailec y Santos Ramirez, Yunier. 2011.** Las TIC en la enseñanza de la Ingeniería de Software en la Universidad de Ciencias Informáticas. Pasado, presente y futuro. s.l. : Edutec--e, Revista Electronica de Tecnologia Educativa, 2011. 37.
34. **Grosso, Andrés. 2011.** Prácticas de Software, Experiencias sobre la Ingeniería y Management del Software. [En línea] 2011. <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
35. **Guardia, L. 2000.** El diseño formativo: Un nuevo enfoque de diseño pedagógico de los materiales didáctico en soporte digital. *Aprender en la virtualidad.* s.l. : Editorial Gedisa,S.A, 2000.
36. **Herrera Chica, Isael . 2013.** *Conjunto de Objetos de Aprendizaje para la Matemática Discreta.* 2013.
37. **IEEE. 2002.** IEEE, Learning Object Metadata. [En línea] 2002. <http://ltsc.ieee.org/wg12>.
38. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** El proceso unificado del software. 2000.
39. **López Guzmán, Clara. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e -Learning, Tesina Doctoral.* s.l. : Universidad de Salamanca, 2005.
40. **Mañas, José A. 1994.** Prueba de Programas. *Laboratorio de Programación. Material de estudio. Apuntes. Prueba de programas.* 1994.
41. **Marqués Graells, Pere. 2008.** Impacto de las TIC en la enseñanza universitaria. DIM: Didáctica, innovación y multimedia. 2008.

42. **Marqués Graells, Pere. 2011.** Impacto de las TIC en educación: funciones y limitaciones. Tecnología Educativa - Web *Pere Marqués*. [En línea] 2011. <http://peremarques.pangea.org/siyedu.htm>.
43. **Molero Aparicio, María y Salvador Alcaide, Adela. 2008.** Los medios tecnológicos y la enseñanza de las Matemáticas. Instituto Juan de la Cierva de Madrid : Segundo Congreso Internacional de Matemáticas en la Ingeniería y la Arquitectura, 2008.
44. **Montenegro, Dr.C. Sylvia Lima. 2007.** Educación a distancia y preparación de materiales educativos con uso de la hipermedia. 2007. 1.
45. **Montero O’Farrill, J. L y Herrero Tunis, E. 2008.** Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital. 2008.
46. **Moya Martínez, María Antonia. 2010.** Recursos didácticos en la enseñanza. s.l. : Revista Digital: Innovación y Experiencias Educativas, 2010. ISSN: 1988-6042.
47. **Murcia, Olga Sofia Lopez. 2009.** Estrategias Metodológicas en Matemáticas. [En línea] 30 de Abril de 2009. <http://olgasofialopez.blogspot.com/2009/04/estrategias-metodologicas-en.html>.
48. *Nuevas tecnologías, comunicación y educación.* **Cabero Almenara, Julio. 1996.** 1, s.l. : Universidad de Sevilla, 1996, EDUTEC: Revista Electrónica de tecnología educativa 1.
49. **Oracle Corporation. 2013.** Netbeans. [En línea] 2013. www.netbeans.org.
50. **PACE. 2013.** Plan de Acciones para la Convergencia Europea. *Agència per a la Qualitat del Sistema Universitari de Catalunya*. [En línea] 2013. www.aqu.cat/doc/doc_22391979_1.pdf.
51. **Pernalte Chirinos, Doris y Delgado Herrera, Mayela. 2009.** Desarrollo de un Objeto de Aprendizaje con un Enfoque de Calidad sobre árboles binarios de búsqueda. [En línea] 2009.
52. **Pressman, Roger S. 2002.** *Software engineering. A Practitioner's Approach*. Madrid : McGraw-Hill Ryerson, 2002.
53. **REDAOPA. 2004.** Estado de la cuestión: uso de Objetos y diseños para el aprendizaje. 2004.
54. **Rosen, Kenneth H y AT&T, Laboratorios. 2004.** *Matemática Discreta y sus Aplicaciones*. 5. s.l. : Mc Graw Hill, 2004.
55. **Rovira Prieto, Imirys y González Páez, Luis Ramón. 2012.** Multimedia para la enseñanza de Matemática Discreta en la Universidad de las Ciencias Informáticas. 2012.
56. **Salinas Ibañez, J. 2004.** Cambios metodológicos con las TIC. Estrategias didácticas y entornos virtuales de enseñanza-aprendizaje. 2004.
57. **—. 2009.** Hacia nuevas formas metodológicas en e-learning. Formación XXI. s.l. : Revista de Formación y empleo, 2009.

-
58. —. **2008**. Innovación educativa y uso de las TIC. s.l. : Sevilla: Universidad Internacional de Andalucía., 2008.
59. **Salinas Ibañez, J. 2007**. El papel de las TIC en el sistema educativo. *Presentado en MA D 2.0 Cicle de Conferències sobre Noves Tecnologies i Societat, Palma de Mallorca*. 2007.
60. **Salinas, J y Urbina, S. 2007**. Bases para el diseño, producción y la evaluación de procesos de Enseñanza-Aprendizaje mediante nuevas tecnologías. *Nuevas tecnologías aplicadas a la educación*. 2007.
61. **Silva Díaz, Jorge . 2003**. Diagnóstico de saberes y movimientos gnoseológicos en el área de las Matemáticas. *Revista virtual Universidad Católica del Norte*. [En línea] 2003. <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/293/556>.
62. **Spiegel, Alejandro**. *Recursos Didácticos: Herramientas para enseñar, aprender y evaluar*. Competencia Laboral.
63. **Sun Microsystems Inc. 1999**. *Convenciones de código para el lenguaje de programación JAVA*. 1999.
64. **Toll Palma, Yuniet del Carmen y Ril Gil, Yohandri . 2010**. La calidad de los objetos de aprendizaje producidos en la Universidad de las Ciencias Informáticas. s.l. : Grupo Editorial Ediciones Futuro, 2010. Vol. 3, 12.
65. **Toll Palma, Yuniet del Carmen , y otros. 2011**. Programa de mejoras en el proyecto repositorios de objetos de aprendizaje RHODA del Centro Tecnologías para la Formación FORTES en la Universidad de las Ciencias Informáticas. La Habana : s.n., 2011. Vol. 4, 11.
66. **Toll Palma, Yuniet del Carmen. 2011**. Guía de evaluación de la calidad de Objetos de Aprendizaje producidos en la Universidad de Ciencias Informáticas. 2011.
67. **Torres Gil, Manuel**. Técnicas de pruebas.
68. **UCI. 2010**. *Proyecto Estratégico 2008-2012*. Habana : s.n., 2010.
69. **Wiley, David A. 2001**. Connecting learning objects to instructional design theory: A definition, a metaphor and a taxonomy, 2001. [En línea] 2001. <http://www.elearning-reviews.org/topics/technology/learning-objects/2001-wiley-learning-objects-instructional-design-theory.pdf>.
70. **Wiley, David A. 2002**. *The Instructional Use of Learning Objects*. s.l. : Agency for Instructional Technology; 1 edition (2002), 2002.
71. **Wiley, David. 1999**. So what do I do with a learning object? 1999.

ANEXOS

Anexo 1: Interfaces para crear ejercicios en el OA-IE de Teoría Combinatoria

Figura 25: Interfaz Crear ejercicio tipo 1.

Teoría Combinatoria

Crear ejercicios Tipo 1

De un total de escoger para

Orden

Importa

No importa

Ej: De cuántas formas se pueden escoger 3 estudiantes de un total de 10 para ir al teatro?

Derechos de autor © 2014 UCL.

Figura 26: Interfaz Crear ejercicio tipo 2.

Teoría Combinatoria

Crear ejercicios Tipo 2

En una caja hay de los tipos y
objetos, bolas, medias, etc..

Hay al menos de cada tipo. ¿De cuántas formas se pueden escoger ?

Si debe haber al menos del tipo y al menos
 exactamente exactamente

del tipo

Derechos de autor © 2014 UCL.

Figura 27: Interfaz Crear ejercicio tipo 3.

Teoría Combinatoria

Contenidos

Teoría Combinatoria

- Experimentación
 - Permutación
 - Combinación
- Principio de la suma
 - Nivel bajo
 - Nivel medio
 - Nivel medio-alto
 - Nivel alto
- Evaluación
 - Ejercicios propuestos
 - Crear ejercicio
 - Tipo 1
 - Tipo 2
 - Tipo 3**
 - Tipo 4
 - Reporte de evaluación
 - Sobre el Applet

Crear ejercicios Tipo 3

¿ Cuántas soluciones enteras no negativas tiene la ecuación

$$X_1 + X_2 + \square X_3 + \square X_4 + \square X_5 = \square ?$$

Donde $X_i, i = 1; 2; 3 \dots n$, son enteros no negativos tales que:

X_1 \geq $=$

X_2 \geq $=$

Derechos de autor © 2014 UCL.

Figura 28: Interfaz Crear ejercicio tipo 4.

Teoría Combinatoria

Contenidos

Teoría Combinatoria

- Experimentación
 - Permutación
 - Combinación
- Principio de la suma
 - Nivel bajo
 - Nivel medio
 - Nivel medio-alto
 - Nivel alto
- Evaluación
 - Ejercicios propuestos
 - Crear ejercicio
 - Tipo 1
 - Tipo 2
 - Tipo 3
 - Tipo 4**
 - Reporte de evaluación
 - Sobre el Applet

Crear ejercicios Tipo 4

¿ Cuántas cadenas de dígitos y dígitos se pueden formar

letras o letras
 vocales vocales
 consonantes consonantes
 bits bits

si los elementos si se pueden repetir ? no

Derechos de autor © 2014 UCL.

Anexo 2: Historias de Usuario

Tabla 20: HU_8 Crear ejercicio de Teoría Combinatoria tipo 1.

| Historia de Usuario | |
|--|---|
| Número: HU_8 | Nombre: Crear ejercicio de Teoría Combinatoria tipo 1. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Media | Tiempo estimado: 0.4 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.4 |
| Descripción: Permite crear nuevos ejercicios para realizar como evaluaciones, del tipo “ <i>De cuántas formas se pueden escoger ‘n1’ elementos de un total de ‘n1’ para...</i> ”. | |
| Observaciones: El usuario deberá llenar y seleccionar todos los campos especificados, sino se mostrará un mensaje advirtiendo el error. | |

Tabla 21: HU_9 Crear ejercicio de Teoría Combinatoria tipo 2.

| Historia de Usuario | |
|---|---|
| Número: HU_9 | Nombre: Crear ejercicio de Teoría Combinatoria tipo 2. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Media | Tiempo estimado: 0.4 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.4 |
| Descripción: Permite crear nuevos ejercicios para realizar como evaluaciones, del tipo “ <i>En una caja hay bolas de los tipos n1, n2 y n3, ¿de cuántas formas se pueden escoger n4, si debe haber al menos/exactamente n5 del tipo ni?</i> ”. | |
| Observaciones: El usuario deberá llenar y seleccionar todos los campos especificados, sino se mostrará un mensaje advirtiendo el error. | |

Tabla 22: HU_10 Crear ejercicio de Teoría Combinatoria tipo 3.

| Historia de Usuario | |
|---------------------------|---|
| Número: HU_10 | Nombre: Crear ejercicio de Teoría Combinatoria tipo 3. |
| Modificación: ---- | |

| | |
|--|------------------------------|
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Media | Tiempo estimado: 0.4 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.4 |
| Descripción: Permite crear nuevos ejercicios para realizar como evaluaciones, del tipo “¿Cuántas soluciones enteras no negativas tiene la ecuación $X_1 + \dots + X_n = 21$?”. | |
| Observaciones: El usuario deberá llenar y seleccionar todos los campos especificados, sino se mostrará un mensaje advirtiendo el error. | |

Tabla 23: HU_11 Crear ejercicio de Teoría Combinatoria tipo 4.

| Historia de Usuario | |
|---|---|
| Número: HU_11 | Nombre: Crear ejercicio de Teoría Combinatoria tipo 4. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 1 |
| Prioridad: Media | Tiempo estimado: 0.4 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.4 |
| Descripción: Permite crear nuevos ejercicios para realizar como evaluaciones. En este caso los ejercicios estarán relacionados con los principios de la Suma, del Producto y de inclusión - exclusión. | |
| Observaciones: El usuario deberá llenar y seleccionar todos los campos especificados, sino se mostrará un mensaje advirtiendo el error. | |

Tabla 24: HU_14 Hallar relación de recurrencia a partir de una sucesión.

| Historia de Usuario | |
|---|---|
| Número: HU_14 | Nombre: Hallar relación de recurrencia a partir de una sucesión. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 2 |
| Prioridad: Alta | Tiempo estimado: 0.5 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.5 |

| |
|---|
| Descripción: Permite hallar una relación de recurrencia a partir de una sucesión. El sistema permite comprobar si la relación de recurrencia entrada por el usuario es correcta o calcular directamente estos valores. |
| Observaciones: |

Tabla 25: HU_15 Clasificar relaciones de recurrencia.

| Historia de Usuario | |
|--|---|
| Número: HU_15 | Nombre: Clasificar relaciones de recurrencia. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 2 |
| Prioridad: Alta | Tiempo estimado: 0.2 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.2 |
| Descripción: Permite clasificar una relación de recurrencia y calcular su orden. Brinda la posibilidad al usuario de crear la relación de recurrencia. El sistema permite comprobar si la respuesta dada por el usuario es correcta o calcular directamente estos valores. | |
| Observaciones: En caso de dejar datos sin entrar se muestra un mensaje de error al hacer clic en los botones “Crear relación”, “Comprobar” o “Mostrar resultado”. Estos últimos aparecen deshabilitados al cargar la vista y se activan al crear la relación de recurrencia, y a su vez se desactiva el botón “Crear relación”. Para deshacer y comenzar otra vez la práctica se debe presionar el botón “Limpiar”. | |

Tabla 26: HU_16 Evaluar identificar sucesiones numéricas y relaciones de recurrencia.

| Historia de Usuario | |
|---|---|
| Número: HU_16 | Nombre: Evaluar identificar sucesiones numéricas y relaciones de recurrencia. |
| Modificación: ---- | |
| Programador responsable: Yasniel Alejandro Carballo Martínez | Iteración Asignada: 2 |
| Prioridad: Alta | Tiempo estimado: 0.5 |
| Riesgo en desarrollo: Medio | Tiempo real: 0.5 |

Descripción: Permite realizar ejercicios sobre el contenido de Relaciones de Recurrencia, para evaluar los conocimientos del usuario. El usuario selecciona el ejercicio que desea realizar, da una respuesta y crea su evaluación.

Observaciones: Al hacer clic en el botón “*Terminar*” se muestra un mensaje que confirma que se ha guardado la evaluación. En caso que el ejercicio a realizar necesita como respuesta los valores de una sucesión y no se llenan los espacios correspondientes a estos valores, se mostrará un mensaje advirtiendo el error.

Anexo 3: Tarjetas CRC

Tabla 27: Tarjeta CRC Panel_Conjunto.

| Panel_Conjunto | |
|---|---|
| Responsabilidades | Colaboraciones |
| Clase encargada de representar cada una de las vistas asociadas a las opciones que aparecen en las unidades definidas para el OA-IE de Teoría de Conjuntos. | Principal, Ventana, Generadora_Conjuntos, Conjunto, Evaluacion_Conj |

Tabla 28: Tarjeta CRC Panel_Grafo.

| Panel_Grafo | |
|--|--|
| Responsabilidades | Colaboraciones |
| Clase encargada de representar cada una de las vistas asociadas a las opciones que aparecen en las unidades definidas para el OA-IE de Teoría de Grafos. | Principal, Ventana, Generadora_Grafos, Grafo, Evaluacion_graf, Arco, Dibujo_Arista, Dibujo_Vertice, Nodo, Nodo_Evaluacion, Pintar_Grafo, Pintar_Grafo_Evaluacion |

Tabla 29: Tarjeta CRC Panel_Turing.

| Panel_Turing | |
|-------------------|----------------|
| Responsabilidades | Colaboraciones |
| | |

| | |
|---|--|
| Clase encargada de representar cada una de las vistas asociadas a las opciones que aparecen en las unidades definidas para el OA-IE de Teoría de la Computabilidad. | Principal, Ventana, Generadora_Cadena, Condicion, Estados, Evaluacion_Turing, Funcion_Transicion, Movimiento |
|---|--|

Tabla 30: Tarjeta CRC Ventana.

| Ventana | |
|--|--|
| Responsabilidades | Colaboraciones |
| Clase encargada de representar la interfaz principal del Sistema de OA-IE. | Panel_Conjunto, Panel_Grafo, Panel_Turing, Panel_Combinatoria, Panel_Recurrencia, Imagen_Contenido |

Tabla 31: Tarjeta CRC Ejercicio_Comb.

| Ejercicio_Comb | |
|--|--------------------|
| Responsabilidades | Colaboraciones |
| Clase modelo encargada de representar ejercicios de Teoría Combinatoria para ser mostrados como parte de evaluaciones. | Panel_Combinatoria |

Tabla 32: Tarjeta CRC Ejercicio_RelacionR.

| Ejercicio_RelacionR | |
|--|-------------------|
| Responsabilidades | Colaboraciones |
| Clase modelo encargada de representar ejercicios de Relaciones de Recurrencia para ser mostrados como parte de evaluaciones. | Panel_Recurrencia |

Tabla 33: Tarjeta CRC Evaluacion_Recurrencia.

| Evaluacion_Recurrencia |
|------------------------|
|------------------------|

| Responsabilidades | Colaboraciones |
|--|-------------------|
| Clase modelo encargada de representar evaluaciones de Teoría Combinatoria. | Panel_Recurrencia |

Anexo 4: Tareas de ingeniería

Tabla 34: TI_1 HU_6 Definir y crear los componentes visuales necesarios que permitan experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 1 | Número de Historia de Usuario: 6 |
| Nombre Tarea: Definir y crear los componentes visuales necesarios que permitan experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.6 |
| Fecha Inicio: 25/2/2014 | Fecha Fin: 03/3/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se realiza el diseño e implementación de la vista que permita experimentar con los principios básicos de Teoría Combinatoria mediante la resolución de ejercicios con una complejidad alta. | |

Tabla 35: TI_2 HU_6 Validación de los datos.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 2 | Número de Historia de Usuario: 6 |
| Nombre Tarea: Validación de los datos. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 04/3/2014 | Fecha Fin: 04/3/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se implementan los métodos necesarios para validar que no se dejen campos vacíos, no se dejen opciones sin seleccionar y que los datos entrados se correspondan con los tipos de datos permitidos. | |

Tabla 36: TI_1 HU_7 Definir y crear los componentes visuales necesarios que permitan evaluar contenido de Teoría Combinatoria.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 1 | Número de Historia de Usuario: 7 |
| Nombre Tarea: Definir y crear los componentes visuales necesarios que permitan evaluar contenido de Teoría Combinatoria. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 05/3/2014 | Fecha Fin: 07/3/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se realiza el diseño e implementación de la vista que permita evaluar el contenido de Teoría Combinatoria. | |

Tabla 37: TI_2 HU_4 Validación de los datos.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 2 | Número de Historia de Usuario: 7 |
| Nombre Tarea: Validación de los datos. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 08/3/2014 | Fecha Fin: 08/3/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |
| Descripción: Se implementan los métodos necesarios para validar que no se dejen campos vacíos, no se dejen opciones sin seleccionar y que los datos entrados se correspondan con los tipos de datos permitidos. | |

Tabla 38: TI_1 HU_ 19 Definir y crear los componentes visuales necesarios para mostrar reporte de evaluaciones de Relaciones de Recurrencia.

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 1 | Número de Historia de Usuario: 19 |
| Nombre Tarea: Definir y crear los componentes visuales necesarios para mostrar reporte de evaluaciones de Relaciones de Recurrencia. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 29/4/2014 | Fecha Fin: 29/4/2014 |
| Programador Responsable: Yasniel Alejandro Carballo Martínez | |

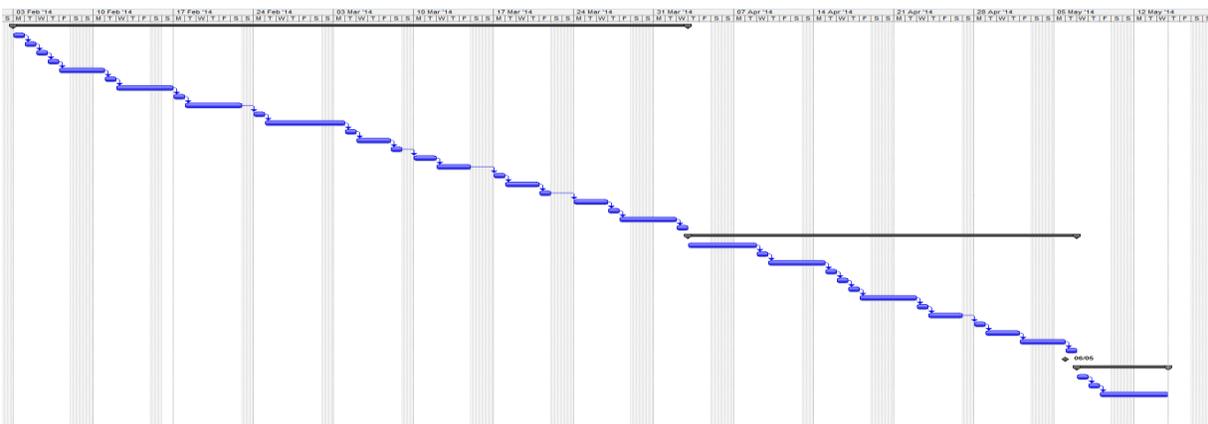
Descripción: Se realiza el diseño e implementación de la vista que permita mostrar un reporte de las evaluaciones realizadas por el usuario.

Anexo 5: Diagrama de Gantt

Figura 29: Tareas de Ingeniería.

| ID | Task Name | Duration | Start | Finish |
|----|--|----------|--------------|--------------|
| 1 | Diseño e implementación del OA de Teoría Combinatoria | 47 days | Mon 03/02/14 | Wed 02/04/14 |
| 2 | Definir y crear los componentes visuales necesarios para realizar la permutación. | 1 day | Mon 03/02/14 | Mon 03/02/14 |
| 3 | Validación de los datos. | 1 day | Tue 04/02/14 | Tue 04/02/14 |
| 4 | Definir y crear los componentes visuales necesarios para realizar la combinación | 1 day | Wed 05/02/14 | Wed 05/02/14 |
| 5 | Validación de los datos. | 1 day | Thu 06/02/14 | Thu 06/02/14 |
| 6 | Definir y crear los componentes visuales necesarios que permitan experimentar con los Principios de la Suma y del Producto con un nivel de complejidad | 2 days | Fri 07/02/14 | Mon 10/02/14 |
| 7 | Validación de los datos. | 1 day | Tue 11/02/14 | Tue 11/02/14 |
| 8 | Definir y crear los componentes visuales necesarios para experimentar con los principios básicos de la teoría combinatoria mediante la resolución de e | 4 days | Wed 12/02/14 | Sun 16/02/14 |
| 9 | Validación de los datos. | 1 day | Mon 17/02/14 | Mon 17/02/14 |
| 10 | Definir y crear los componentes visuales necesarios que permitan experimentar con los Principios de la Suma y del Producto con un nivel de complejidad | 5 days | Tue 18/02/14 | Sat 22/02/14 |
| 11 | Validación de los datos. | 1 day | Mon 24/02/14 | Mon 24/02/14 |
| 12 | Definir y crear los componentes visuales necesarios que permitan experimentar con los principios básicos de la teoría combinatoria mediante la resoluc | 6 days | Tue 25/02/14 | Mon 03/03/14 |
| 13 | Validación de los datos. | 1 day | Tue 04/03/14 | Tue 04/03/14 |
| 14 | Definir y crear los componentes visuales necesarios que permitan evaluar contenido de Teoría Combinatoria. | 3 days | Wed 05/03/14 | Fri 07/03/14 |
| 15 | Validación de los datos. | 1 day | Sat 08/03/14 | Sat 08/03/14 |
| 16 | Definir y crear los componentes visuales necesarios para mostrar reporte de evaluaciones de Teoría Combinatoria. | 2 days | Mon 10/03/14 | Tue 11/03/14 |
| 17 | Definir y crear los componentes visuales necesarios para crear ejercicio de Teoría Combinatoria tipo 1. | 3 days | Wed 12/03/14 | Fri 14/03/14 |
| 18 | Validación de los datos. | 1 day | Mon 17/03/14 | Mon 17/03/14 |
| 19 | Definir y crear los componentes visuales necesarios para crear ejercicio de Teoría Combinatoria tipo 2. | 3 days | Tue 18/03/14 | Thu 20/03/14 |
| 20 | Validación de los datos. | 1 day | Fri 21/03/14 | Fri 21/03/14 |
| 21 | Definir y crear los componentes visuales necesarios para crear ejercicio de Teoría Combinatoria tipo 3. | 3 days | Mon 24/03/14 | Wed 26/03/14 |
| 22 | Validación de los datos. | 1 day | Thu 27/03/14 | Thu 27/03/14 |
| 23 | Definir y crear los componentes visuales necesarios para crear ejercicio de Teoría Combinatoria tipo 4. | 3 days | Fri 28/03/14 | Tue 01/04/14 |
| 24 | Validación de los datos. | 1 day | Wed 02/04/14 | Wed 02/04/14 |
| 25 | Diseño e implementación del OA de Relaciones de Recurrencia | 32 days? | Thu 03/04/14 | Wed 14/05/14 |
| 26 | Definir y crear los componentes visuales necesarios para hallar una sucesión a partir de una relación de recurrencia. | 4 days | Thu 03/04/14 | Tue 08/04/14 |
| 27 | Validación de los datos. | 1 day? | Wed 09/04/14 | Wed 09/04/14 |
| 28 | Definir y crear los componentes visuales necesarios para hallar una relación de recurrencia a partir de una sucesión. | 4 days | Thu 10/04/14 | Mon 14/04/14 |
| 29 | Validación de los datos. | 1 day? | Tue 15/04/14 | Tue 15/04/14 |
| 30 | Definir y crear los componentes visuales necesarios para clasificar relaciones de recurrencia | 1 day | Wed 16/04/14 | Wed 16/04/14 |
| 31 | Validación de los datos | 1 day? | Thu 17/04/14 | Thu 17/04/14 |
| 32 | Definir y crear los componentes visuales necesarios que permitan evaluar identificar sucesiones numéricas y relaciones de recurrencia. | 3 days | Fri 18/04/14 | Tue 22/04/14 |
| 33 | Validación de los datos. | 1 day? | Wed 23/04/14 | Wed 23/04/14 |
| 34 | Definir y crear los componentes visuales necesarios que permitan evaluar clasificación de relaciones de recurrencia. | 3 days | Thu 24/04/14 | Sat 26/04/14 |
| 35 | Validación de los datos. | 1 day? | Mon 28/04/14 | Mon 28/04/14 |
| 36 | Definir y crear los componentes visuales necesarios para mostrar reporte de evaluaciones de Relaciones de Recurrencia. | 3 days | Tue 29/04/14 | Thu 01/05/14 |
| 37 | Definir y crear los componentes visuales necesarios para crear ejercicio de Relaciones de Recurrencia. | 2 days | Fri 02/05/14 | Mon 05/05/14 |
| 38 | Validación de los datos. | 1 day? | Tue 06/05/14 | Tue 06/05/14 |
| 39 | Definir y crear los componentes visuales necesarios que permitan mostrar ayuda para el trabajo con el OA de Relaciones de Recurrencia. | 0 days | Tue 06/05/14 | Tue 06/05/14 |
| 40 | Integración de los OA. | 6 days? | Wed 07/05/14 | Wed 14/05/14 |
| 41 | Estudio del diseño de los OA existentes | 1 day? | Wed 07/05/14 | Wed 07/05/14 |
| 42 | Definición y creación de la interfaz del sistema. | 1 day? | Thu 08/05/14 | Thu 08/05/14 |
| 43 | Implementación de los métodos necesarios para integrar los OA. | 4 days | Fri 09/05/14 | Wed 14/05/14 |

Figura 30: Diagrama de Gantt.



Anexo 6: Pruebas de aceptación

Tabla 39: Caso de prueba de aceptación de la HU_8.

| Caso de prueba de aceptación | | |
|---|---|------------------------|
| Número: 4 | Nombre de historia de Usuario: Crear ejercicio de Teoría Combinatoria tipo 1. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita crear ejercicio de Teoría Combinatoria de tipo 1. | | |
| Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar en el menú “<i>Contenidos</i>” el contenido de Teoría Combinatoria. 2. Seleccionar “<i>Tipo 1</i>” del menú inferior izquierdo, siguiendo la ruta Teoría Combinatoria ->Evaluación-> Crear ejercicio-> Tipo 1. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Ingresar el total de elementos, la cantidad a seleccionar, el tipo de elemento, el propósito de la selección y seleccionar si importa o no el orden. Todos los campos son obligatorios. | Luego de ingresados todos los datos, al presionar “ <i>Crear</i> ” se muestra un mensaje que advierte que el ejercicio ha sido creado exitosamente. Al presionar “ <i>Cancelar</i> ” se limpian todos los datos para volver a ser introducidos. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| No ingresar los datos requeridos al presionar “ <i>Crear</i> ”. Introducir el total de elementos menor que la cantidad a seleccionar. | Se muestra un mensaje de error indicando que no se deben dejar campos vacíos. Se muestra un mensaje advirtiendo que el total de elementos debe ser mayor que la cantidad a seleccionar. | Satisfactorio |
| Evaluación de la prueba: Satisfactorio. | | |

Tabla 40: Caso de prueba de aceptación de la HU_14.

| Caso de prueba de aceptación | | |
|--|--|------------------------|
| Número: 5 | Nombre de historia de Usuario: Hallar relación recurrencia a partir de una sucesión. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita hallar una relación recurrencia a partir de una sucesión. | | |
| Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar en el menú “Contenidos” el contenido de Relaciones de Recurrencia. 2. Seleccionar “Hallar relación de recurrencia” del menú inferior izquierdo, siguiendo la ruta <i>Relaciones de Recurrencia->Experimentación->Hallar relación de recurrencia</i>. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Seleccionar las relaciones que se corresponden con cada una de las sucesiones mostradas al presionar “Comprobar”. No es obligatorio hacer la selección para los casos en que se presiona “Mostrar resultado” o “Más ejercicios”. | Luego de seleccionar la relación de recurrencia en cada caso, al presionar “Comprobar” se muestra un mensaje que advierte si la selección es correcta o no. Al presionar “Mostrar resultado”, se muestran los resultados correctos. Al presionar “Más ejercicios” aparecen otros nuevos. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| No seleccionar las relaciones que se corresponden con cada una de las sucesiones mostradas al presionar “Comprobar”. | Se muestra un mensaje de error indicando que se deben seleccionar las relaciones que se corresponden con cada una de las sucesiones mostradas. | Satisfactorio |
| Evaluación de la prueba: Satisfactorio. | | |

Tabla 41: Caso de prueba de aceptación de la HU_16.

| Caso de prueba de aceptación |
|------------------------------|
|------------------------------|

| | | |
|---|---|-------------------------------|
| Número: 6 | Nombre de historia de Usuario: Evaluar identificar sucesiones numéricas y relaciones de recurrencia. | |
| Nombre de la persona que realiza la prueba: Yasniel Alejandro Carballo Martínez | | |
| Descripción de la prueba: Verificar que permita evaluar identificar sucesiones numéricas y relaciones de recurrencia. | | |
| Pasos de ejecución: | | |
| <ol style="list-style-type: none"> 1. Seleccionar en el menú “<i>Contenidos</i>” el contenido de Relaciones de Recurrencia. 2. Seleccionar “<i>Identificar sucesiones y relaciones</i>” del menú inferior izquierdo, siguiendo la ruta <i>Relaciones de Recurrencia->Evaluación-> Identificar sucesiones y relaciones</i>. | | |
| Casos válidos | Resultado esperado | Resultado de la prueba |
| Seleccionar la evaluación que desea realizar. Si se pide identificar una relación a partir de una sucesión, entrar los coeficientes de los términos y seleccionar los operadores entre los términos, presionar “ <i>Terminar</i> ”. Si se pide identificar una sucesión a partir de una relación, se deben entrar los valores de los términos de la sucesión y presionar “ <i>Terminar</i> ”, en este caso los campos son obligatorios. | Luego de seleccionar la evaluación e introducir los datos que se requieren, al presionar “ <i>Terminar</i> ” se debe mostrar un mensaje que advierte que la evaluación ha sido guardada exitosamente. | Satisfactorio |
| Casos no válidos | Resultado esperado | Resultado de la prueba |
| En caso de que el ejercicio pida identificar una sucesión a partir de una relación, no entrar los valores de los términos de la misma. | Se debe mostrar un mensaje de error indicando que se deben entrar los valores de los términos de la sucesión. | Satisfactorio |
| Evaluación de la prueba: Satisfactorio. | | |

Anexo 7: Guía para la validación de los OA-IE como software pedagógico

Tabla 42: Indicadores del aspecto formativo.

| No | Aspecto formativo | Evaluador | | | | | |
|----|--|-----------|----|----|----|----|----|
| | | E1 | E2 | E3 | E4 | E5 | E6 |
| 1 | Presentación y explicación del tema a tratar. | | | | | | |
| 2 | Estructuración lógica de los contenidos. | | | | | | |
| 3 | Exhortación del desarrollo de habilidades y competencias al estudiante. | | | | | | |
| 4 | Reflexión sobre lo aprendido. | | | | | | |
| 5 | Autoevaluación sobre el contenido mostrado en el OA-IE. | | | | | | |
| 6 | Calidad de los contenidos. | | | | | | |
| 7 | Adecuación de los objetivos de aprendizaje. | | | | | | |
| 8 | Retroalimentación que proporciona el contenido mostrado. | | | | | | |
| 9 | Motivación. | | | | | | |
| 10 | Organización del trabajo individual y/o colaborativo de los estudiantes. | | | | | | |
| 11 | Nivel de presentación de recursos audiovisuales. | | | | | | |
| 12 | Participación activa durante el aprendizaje mediante actividades interactivas. | | | | | | |
| 13 | Verificación de las fuentes de información utilizadas. | | | | | | |
| 14 | Aporte según el contenido de los recursos audiovisuales a los estudiantes. | | | | | | |
| | Total: | | | | | | |

Tabla 43: Indicadores del aspecto de diseño y presentación.

| No | Aspecto de diseño y presentación | Evaluador | | | | | |
|----|---|-----------|----|----|----|----|----|
| | | E1 | E2 | E3 | E4 | E5 | E6 |
| 1 | Correspondencia entre los recursos audiovisuales y el contenido mostrado. | | | | | | |
| 2 | Visibilidad del texto. | | | | | | |
| 3 | Rapidez para cargar recursos audiovisuales. | | | | | | |

| | | | | | | | |
|---------------|---|--|--|--|--|--|--|
| 4 | Proporción del texto respecto a la distribución de los contenidos dentro del OA-IE. | | | | | | |
| 5 | El uso de colores para los contenidos. | | | | | | |
| 6 | Manejo de formatos uniformes dentro de los OA-IE. | | | | | | |
| 7 | Diversidad en la representación del contenido mostrado. | | | | | | |
| 8 | El diseño de la información audiovisual. | | | | | | |
| 9 | Visibilidad de las imágenes. | | | | | | |
| 10 | Evaluación del nivel de organización de las imágenes y texto. | | | | | | |
| Total: | | | | | | | |

Tabla 44: Indicadores del aspecto tecnológico.

| No | Aspecto tecnológico | Evaluador | | | | | |
|---------------|---|-----------|----|----|----|----|----|
| | | E1 | E2 | E3 | E4 | E5 | E6 |
| 1 | Usabilidad | | | | | | |
| 2 | Accesibilidad | | | | | | |
| 3 | Reusabilidad | | | | | | |
| 4 | Facilidad de indexado del OA-IE dentro de un repositorio | | | | | | |
| 5 | Compatibilidad con distintos navegadores | | | | | | |
| 6 | Nivel de organización de la estructura de archivos | | | | | | |
| 7 | Calidad de las imágenes | | | | | | |
| 8 | Integridad de los enlaces de navegación por la estructura didáctica | | | | | | |
| 9 | Correspondencia con la estructura didáctica | | | | | | |
| 10 | Calidad de la redacción y ortografía en la exposición del contenido | | | | | | |
| Total: | | | | | | | |