



Centro de Identificación y Seguridad Digital

Facultad 1

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

**Conjunto de
Objetos de Aprendizaje
para la
Matemática Discreta**

Autor: Isael Herrera Chica

**Tutores: Dr. C. Edistio Yoel Verdecia Martínez, P. Aux.
Ing. Alién García Hernández, Instructor.**

La Habana

2012

Declaración de Autoría

Declaramos ser autores 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 _____.

Isael Herrera Chica

Firma del Autor

Dr. C. Edistio Yoel Verdecia Martínez

Firma del Tutor

Ing. Alién García Hernández

Firma del Tutor

Datos de contacto

Nombre y apellidos: Isael Herrera Chica.

Correo: ichica@estudiantes.uci.cu

Situación laboral: Estudiante.

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Nombre y Apellidos: Dr. C. Edistio Yoel Verdecia Martínez, Profesor Auxiliar.

Correo: edistioyoel@uci.cu

Situación laboral: Profesor.

Años de graduado: 17 años.

Especialidad de graduación: Licenciado en Cibernética - Matemática. Máster en Gestión de Proyectos Informáticos. Doctor en Ciencias Pedagógicas

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Nombre y apellidos: Ing. Alién García Hernández, Instructor.

Correo: agarciah@uci.cu

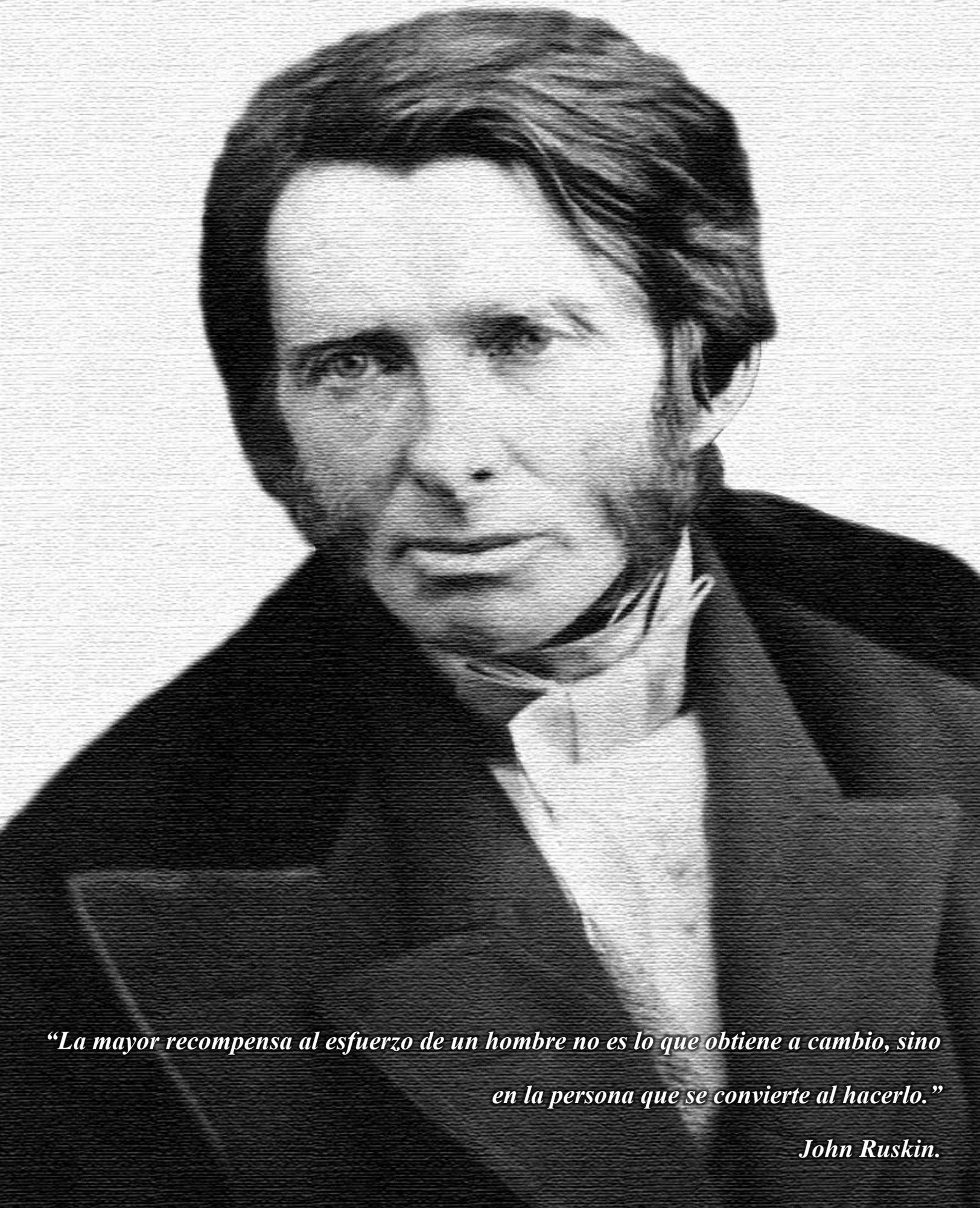
Situación laboral: Profesor Instructor.

Años de graduado: 3 años

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.



“La mayor recompensa al esfuerzo de un hombre no es lo que obtiene a cambio, sino en la persona que se convierte al hacerlo.”

John Ruskin.

Agradecimientos

Tan suficiente no sería entonces mi felicidad, de no haberme escoltado los esfuerzos y la dedicación todo este ministerio del amor, que de una forma u otra me motivaron a culminar satisfactoriamente mis estudios, sería un acto imperdonable no mencionarlos, ellos son:

A mi madre, abuelos y hermana

Mi principal fuente de energía, el alma de todos mis pasos, por obsequiarme siempre sin temor a desilusiones eses inagotable organismo que es el amor, por toda la confianza, el apoyo que me brindan, por todo esto y más he decidido que mi reputación destile hasta la última gota, por vestir de orgullo sus días y colmarles de honores hasta la eternidad.

A mis compañeros de Universidad

Por hacerme más amena la estancia en este cuartel general de los ceros y unos y por su apoyo y complicidad en este acto sagrado de consagración.

A mis profesores

La matriz de mis conocimientos, el antivirus de la mediocridad, el sinónimo más apropiado del profesionalismo, por el noble acto de educarnos y educarme.

A mis tutores

Edistio y Alién, kamikazes del oficio de guiarme hacia la cumbre de mis metas, heraldos de todos mis aciertos. Por el tiempo reclutado desinteresadamente de todos sus consejos, por la consideración exacta de sus preocupaciones por mis resultados, por todo el honor que ha sido para mí su colaboración.

En general

A todos los que de una forma u otra han sido cómplices de esta contienda...Gracias.

Dedicatoria

Imperecederamente:

A mi madre, comandante de todas las revoluciones de mi cuerpo.

A mi abuelo, padre de mi alma, de todas las ausencias y hasta de mis zapatos.

A mi abuela, tan llena de madres para mí y de arcángeles para mi boca.

A mi hermana, la de más victorias en todas mis batallas contra ella, por quererme tanto.

A mi tío Pablo, tutor de un tanto por ciento de los alivios de mis huesos y mi bolsillo.

A Rosy, mi novia y amiga, colega en esta travesía tan desbordada de esfuerzos y software.

A mi primo Reinier, el más ebrio de todos mis consejos.

A mis tutores, inquilinos de la sabiduría, encargados para guillotinar todos mis absurdos y poner en orden estos cinco años de mi vida estudiantil.

Resumen

En los últimos años con el desarrollo de la Informática la enseñanza virtual ha ocupado un lugar significativo, esto ha traído consigo que los educadores busquen ideas de cómo mejorar sus asignaturas para que sean más interactivas y garanticen el aprendizaje de los estudiantes. En la Universidad de las Ciencias Informáticas (UCI) se usa un Entorno Virtual de Aprendizaje (EVA) donde se pueden encontrar recursos de cada asignatura. Entre las asignaturas que se pueden encontrar en el EVA se encuentra la Matemática Discreta (MD) que ha identificado como recursos muy importantes y de los cuales no se disponían: los Objetos de Aprendizaje (OA); por lo que se plantea, como objetivo de la presente investigación desarrollar un grupo de OA que permitan una mayor interactividad y experimentación con determinados contenidos en el proceso de enseñanza-aprendizaje. Se realizó una valoración de las metodologías, herramientas y lenguajes que se ajusten al desarrollo de OA, en este caso se seleccionó XP e ISD-MeLO como metodologías, como herramienta el entorno de desarrollo integrado Netbeans y como lenguaje Java para la creación de los OA como aplicaciones tipo applet. Se creó un grupo de OA interactivos y experimentales con un alto componente visual y grado de interactividad para apoyar el proceso de enseñanza-aprendizaje de la MD a partir de los contenidos que los profesores determinaron eran más necesarios y que mostraban mayores posibilidades de ser graficados e interactuar con ellos. Para validar la calidad de los OA se seleccionó la guía propuesta por Toll Palma. En el documento se reflejan detalles específicos del uso y la concepción de los OA, tanto para usuarios finales como para futuros desarrolladores. 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.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO I. LOS OBJETOS DE APRENDIZAJE Y SU UTILIZACIÓN EN EL PROCESO DE ENSEÑANZA APRENDIZAJE DE LAS MATEMÁTICAS DISCRETAS.....	8
1.1 IMPORTANCIA DE LA MATEMÁTICA DISCRETA EN LA FORMACIÓN DE LOS PROFESIONALES DE LAS TIC	8
1.2 CARACTERIZACIÓN DEL PROCESO DE ENSEÑANZA APRENDIZAJE DE LA MATEMÁTICA DISCRETA EN LA UCI	10
1.3 LA UTILIZACIÓN DE OBJETOS DE APRENDIZAJE EN EL PROCESO DE ENSEÑANZA APRENDIZAJE	15
1.4 DIAGNÓSTICO DE LA UTILIZACIÓN OBJETOS DE APRENDIZAJE EN EL PROCESO DE ENSEÑANZA APRENDIZAJE DE LA MATEMÁTICA DISCRETA EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.....	35
CONSIDERACIONES FINALES DEL CAPÍTULO.....	36
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE OBJETOS DE APRENDIZAJE INTERACTIVOS Y EXPERIMENTALES (OA-IE) PARA APOYAR EL PROCESO DE ENSEÑANZA DE LAS MATEMÁTICAS DISCRETAS	38
2.1 SELECCIÓN DE CONTENIDOS PARA DESARROLLAR OBJETOS DE APRENDIZAJE	38
2.2 SELECCIÓN DE LA METODOLOGÍA, HERRAMIENTAS Y LENGUAJES A UTILIZAR	40
2.3 DISEÑO DE LOS OBJETOS DE APRENDIZAJE	43
CONCLUSIONES PARCIALES	57
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LOS OBJETOS DE APRENDIZAJE	59
3.1 IMPLEMENTACIÓN DE LOS OA	59
3.2 VALIDACIÓN DE LOS OA.....	66
CONCLUSIONES PARCIALES	75
CONCLUSIONES.....	76
RECOMENDACIONES.....	78
BIBLIOGRAFÍA.....	79
ANEXOS	82

Introducción

En la actualidad se vive en un mundo dominado por el avance de las nuevas tecnologías, donde la informática juega un papel fundamental en casi todos los ámbitos de la sociedad. La informática es una de las áreas de conocimiento que se ha desarrollado aceleradamente en los últimos años dentro de las vinculadas a las Tecnologías de la Información y las Comunicaciones (TIC), una gran cantidad de disciplinas científicas dependen de ella ya que propicia un conjunto de técnicas y conocimientos que hacen posible el tratamiento automatizado de la información por medio de computadoras (Prieto, et al., 2004).

Las TIC se pueden definir como el conjunto de tecnologías que permiten a las personas la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de información de manera instantánea a través de medios tecnológicos de última generación, ya sea en formato de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética (Rosario, 2005). Abarcan todos los ámbitos de la experiencia cotidiana: el trabajo, las modalidades para comprar y vender, los trámites, el acceso a la salud y los procesos pedagógicos, que al incorporarlas se ha visto en la necesidad de replantear las maneras tradicionales de los procesos de Enseñanza-Aprendizaje (PEA).

Actualmente las TIC son un elemento clave dentro del sistema educativo. La incorporación de las mismas permite nuevas formas de acceder, generar y transmitir información y conocimientos, a la vez que permite flexibilizar el tiempo y el espacio en el que se desarrolla la acción educativa. También conlleva al uso de nuevas estrategias y metodologías docentes para lograr una enseñanza activa, participativa y constructiva (Moya Martínez, 2009). Esto por supuesto demuestra lo acertado de utilizar las TIC como apoyo al desarrollo del PEA, un ejemplo de ello es la Educación a distancia, o los modelos semi - presenciales, donde no es necesaria la presencia física del profesor para el desarrollo del proceso. En estos modelos se les brinda mayor flexibilidad a los estudiantes para gestionar su tiempo.

Uno de los elementos que vinculan a las TIC y la educación y que se ha impuesto en la última década son los denominados Objetos de Aprendizajes (OA), que según (Wiley, 2000) son recursos informativos digitales que pueden ser reusados para apoyar el aprendizaje.

Los OA son un nuevo tipo de elemento instruccional computarizado que surge del paradigma de modelación orientado a objetos utilizado en Ciencias de la Computación (CC) y que ayudan a los usuarios en la realización de tareas y, por ende, al logro de las competencias planteadas. Existe una innumerable cantidad de estos recursos que traen beneficios como: motivar el auto-aprendizaje, posibilitar el acceso remoto a la información (en línea), desarrollar el análisis y la reflexión y proponen además mecanismos

para la aclaración de dudas. A pesar de estas ventajas, muchos de los OA con que se cuenta hoy en día poseen dos grandes limitantes, una de ellas es que poseen un bajo grado de interactividad, lo que impide dar el control de navegación a los usuarios para que exploren a voluntad el contenido, privándolos de facilidades como la navegación, retroalimentación, evaluación y la colaboración. La segunda gran limitante es que estos OA no permiten experimentación, o sea, no brindan herramientas para la transferencia y aplicación de lo aprendido, no permiten el diálogo simulado y no contienen mecanismos de control. Estas desventajas traen consigo que el efecto de realismo en el contenido que se presenta sea bajo y que se dificulte la comprensión de información didáctica abstracta o compleja.

Cuba no ha quedado exenta de estos cambios, por lo que en las universidades cubanas es cada vez mayor el número de medios informáticos que se usan para apoyar la enseñanza. Dentro de esta gama se pueden mencionar los Laboratorios Virtuales (LT), Entornos Virtuales de Enseñanza-Aprendizaje (EVE/A), libros electrónicos, sistemas multimedia y sitios web entre otros. Un laboratorio virtual (LV) es un sistema computacional que pretende aproximar el ambiente de un Laboratorio Tradicional, y tiene como ventajas que acerca y facilita a un mayor número de alumnos la realización de experiencias, aunque alumno y laboratorio no coincidan en el espacio. Es una herramienta de auto-aprendizaje, donde el alumno altera las variables de entrada, configura nuevos experimentos, aprende el manejo de instrumentos, personaliza el experimento, etc. (Rosado, et al., 2002). Por otro lado los EVE/A son un conjunto de facilidades informáticas y telemáticas para la comunicación y el intercambio de información en el que se desarrollan procesos de enseñanza – aprendizaje. En un EVE/A interactúan, fundamentalmente, profesores y estudiantes (Mestre Gómez, et al., 2007).

En la Universidad de las Ciencias Informáticas (UCI) los estudiantes cuentan entorno tecnológico desarrollado, donde el estudiante tiene posibilidad de realizar una auto preparación con el apoyo de las TIC, la mayoría de los apartamentos de la residencia estudiantil poseen una computadora, además de las existentes en los laboratorios docentes y en el centro de información. Un elemento importante es el Entorno Virtual de Enseñanza / Aprendizaje (EVE/A), este se implementa sobre Moodle (Modular Object - Oriented Dynamic Learning Environment), herramienta para producir y gestionar cursos basados en internet, páginas web y procedimientos que permitan fácilmente la comunicación a través de Internet y el trabajo colaborativo (de la Torre, 2006).

Al Moodle que se utiliza en la UCI se le referirá como EVA-UCI, a partir de que se ha personalizado para satisfacer las necesidades de la organización, un ejemplo de ello es la creación automática de grupos. En

el EVA-UCI se pueden encontrar contenidos, orientaciones y recursos de cada asignatura que conforma el plan de estudio de la Ingeniería en Ciencias Informáticas (ICI), así como bibliografía digital que ayuda en la asimilación de los contenidos que se reciben.

La Matemática Discreta (MD) es una de las áreas de conocimiento necesarias en los estudios de Ingeniería Informática y de CC. La MD es la base de una gran parte de lo relacionado con los procesos digitales, proporciona las bases matemáticas para diferentes aspectos de la informática como son: estructuras de datos, algoritmia, bases de datos, teoría de autómatas, sistemas operativos, investigación operativa y ayuda además al desarrollo de ciertas capacidades fundamentales para un ingeniero: capacidad de formalizar, de razonar rigurosamente y de representar adecuadamente algunos conceptos (Carrazana Cordero, 2009).

La introducción de la enseñanza de la MD en la UCI comienza en el curso 2003-2004, actualmente los contenidos de la MD se encuentran divididos en dos asignaturas, Matemática Discreta 1 (MD1) y Matemática Discreta 2 (MD2) donde se imparten los siguientes contenidos: teoría de conjuntos, relaciones binarias, lógica clásica, circuitos lógicos, técnicas de demostraciones, teoría de los números, teoría de la computabilidad, relaciones de recurrencia, combinatoria y teoría de grafos.

Los contenidos de las MD presentan variada complejidad, la teoría de conjuntos, las relaciones binarias y la teoría combinatoria presuponen conocimientos con los que de alguna manera el estudiante ha interactuado desde la enseñanza media superior. Según encuestas realizadas a profesores que imparten MD y que colaboraron con la investigación se determinó que los contenidos más complejos los constituyen los de lógica, teoría de la computabilidad, relaciones de recurrencia y teoría de grafos. El contenido de más lenta asimilación por los estudiantes es el de teoría de la computabilidad, por las habilidades de pensamiento creativo y algorítmico que requiere. Por otra parte algunos contenidos como el de teoría de conjuntos se pueden representar de manera gráfica, lo que puede facilitar la asimilación por parte de los estudiantes.

En el ámbito internacional Se pueden encontrar diferentes applets relacionados con contenidos de la Matemática Discreta, como son: Máquinas de Turing (MT), Teoría de conjuntos (TC), relaciones binarias, teoría de Grafos (TG), etc. Un applet es una mini-aplicación escrita en Java que se ejecuta en un navegador web al cargar una página HTML que incluye información sobre el applet a ejecutar por medio de los tags <APPLET>...</APPLET> (García de Jalón, et al., 2000).

Varios ejemplos de estos applets son:

- Turing Machine Applet de Whartman Associates que permite sumar y multiplicar números. La aplicación muestra el flujo de eventos del cálculo (la lectura u escritura de los símbolos en la cinta) y las reglas de decisión usadas para procesar cada paso en el cálculo. Este applet se puede encontrar en la dirección (<http://www.warthman.com/ex-turing.htm>).
- Turing Machine Applet de la Universidad de Canterbury. Esta aplicación simula una Máquina de Turing que está diseñado para copiar una cadena que contiene n números 1. Se detiene dejando $2n$ números 1 en la cinta. Este applet se puede encontrar en la dirección (<http://www.cosc.canterbury.ac.nz/mukundan/thco/TM2.html>).
- Simulador de Grafos de la Facultad de Informática de la Universidad Politécnica de Madrid. Este applet consta de dos fases, en la primera de ellas hay que construir el grafo al que se quiera aplicar el algoritmo, una vez que se ha construido el grafo se pasa a la segunda fase que es la de resolución del algoritmo. Este applet se puede encontrar en la dirección (http://www.dma.fi.upm.es/java/matematicadiscreta/busqueda/pag_applet.htm#applet).
- Simulador de diagramas de Venn. Posibilita realizar operaciones entre conjuntos. Este applet se puede encontrar en la dirección (<http://www.cs.kent.ac.uk/people/staff/pjr/EulerVennCircles/EulerVennApplet.html>).

De manera general estos applets permiten ver o simular de manera gráfica algunos de los contenidos mencionados. Sin embargo el nivel de interacción y de experimentación con estos applets es pobre y en muchos casos las simulaciones están prediseñadas, estando limitadas las posibilidades en cuanto al diseño de nuevas simulaciones. El principal inconveniente de estos recursos es que no permiten el envío de información de manera bidireccional entre el usuario y la aplicación, es la aplicación quien brinda información invariable que ya está definida por defecto. Dada la importancia de la MD se han realizado algunas investigaciones como la de (García Hernández, et al., 2011) presentada en el Congreso Internacional Pedagogía 2011 para lograr una mayor comprensión y motivación por parte de los estudiantes hacia la MD. Otros trabajos relacionados con la MD son los de: (Morales Peña, 2011), (García Hernández, et al., 2011) y (García Hernández, et al., 2011).

Realizando un análisis de los recursos de las asignaturas MD1 y MD2 en el EVA-UCI se percibe que en dichos espacios solo se encuentran documentos, la mayoría de ellos en formato “.pdf”, lo que desaprovecha las ventajas que brindan las TIC en cuanto a la utilización, por ejemplo, de OA como parte

de los elementos que pudieran ayudar en la asimilación de los contenidos con un nivel mayor de dificultad.

De lo anteriormente expuesto los **problemas detectados** se pueden resumir en:

- Los materiales que se encuentran en el espacio de la asignatura en el EVA-UCI solo proporcionan orientación sin permitir la experimentación en los contenidos de la asignatura.
- Existe dentro del programa de las asignaturas de MD una pobre variedad de ejercicios y tareas que limitan la asimilación adecuada de los contenidos.
- El aprovechamiento de las potencialidades TIC en el PEA de la MD es insuficiente.
- Es insuficiente la existencia de recursos 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 tener un grado de interactividad elevado que permita la experimentación.

En base a lo antes expuesto se plantea como **problema científico** de la presente investigación el insuficiente aprovechamiento de las TIC en el proceso de enseñanza aprendizaje de la Matemática Discreta.

Para esta investigación se define como **objeto de estudio** el proceso de enseñanza aprendizaje de la Matemática Discreta en la Universidad de las Ciencias Informáticas apoyado en las TIC y el **campo de acción** lo constituye el desarrollo de objetos de aprendizaje que sean interactivos y experimentales.

El **objetivo general** que se traza para la investigación es desarrollar un conjunto de objetos de aprendizajes interactivos y experimentales para apoyar el PEA de la Matemática Discreta.

Para darle solución a este objetivo se definieron las siguientes **tareas**:

1. Caracterización del PEA de la Matemática Discreta a nivel internacional, nacional y en la UCI.
2. Caracterización de los OA, las metodologías y herramientas para su desarrollo.
3. Determinación del estado del uso de OA en el PEA de la MD en la UCI.
4. Selección de los contenidos para desarrollar OA.
5. Diseño de los OA para los contenidos seleccionados.
6. Implementación y verificación de los OA.
7. Valoración de la calidad como software y como recurso pedagógico de los OA elaborados.

Como **idea a defender** en la investigación se plantea: "Si se desarrollan objetos de aprendizajes, con un alto grado de interactividad y donde el estudiante pueda experimentar interactuando con algunos parámetros y visualizando su efecto, estos se pueden utilizar en el proceso de enseñanza aprendizaje de

la matemática discreta garantizando que sean correctos desde el punto de vista tecnológico, de presentación, formativos y de software”.

Para dar cumplimiento al objetivo propuesto se utilizaron métodos teóricos y empíricos y técnicas de investigación. Como **métodos teóricos** se emplearon:

- **Histórico lógico:** Con el objetivo de conocer los antecedentes del problema, la evolución del mismo y las investigaciones que se han llevado a cabo con anterioridad. Permite caracterizar el objeto de estudio.
- **Analítico sintético:** Se realiza un análisis de la bibliografía referente al proceso de desarrollo de OA para definir sus características más importantes desde el punto de vista de considerarlo un producto de software y un producto pedagógico.
- **Modelación:** Para la modelación de cada uno de los OA. Permite la abstracción de la realidad al modelo informático y de ahí buscar la manera de implementarlo, concretando las características de cada uno de los contenidos seleccionados.

Los **métodos empíricos** utilizados fueron:

- **Entrevista y entrevista focalizada:** Se utiliza para obtener la situación real del uso de los OA en la MD, los criterios de los profesores de la conveniencia o no de usarlos y entender sus especificidades para poder llegar a plantear una solución. Además en el caso de la entrevista focalizada para determinar las posibilidades de interacción de los contenidos seleccionados que pudieran favorecer la asimilación de los mismos.
- **Revisión documental:** con el objetivo de analizar los documentos publicados en el espacio creado por el EVA-UCI para la MD y determinar las características que deben tener los OA.

Los **métodos estadísticos – matemáticos** empleados fueron la estadística descriptiva en MS Excel, que permitió el procesamiento y la interpretación de los datos asociados a la validación por el criterio de expertos.

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

En el primer capítulo denominado **“Los objetos de aprendizaje y su utilización en el proceso de enseñanza aprendizaje de las Matemáticas Discretas”**, se caracteriza el PEA de la MD en la UCI y se

realiza una valoración de la importancia de la MD para la formación de futuros profesionales de las TIC. Se caracterizan además los OA, y las diferentes metodologías y herramientas para su desarrollo. Finalmente se realiza un diagnóstico del estado del uso de OA en el PEA de la MD en la UCI. El Capítulo 2: ***“Análisis y diseño de objetos de aprendizaje interactivos y experimentales para apoyar el proceso de enseñanza de las matemáticas discretas”***, describe detalladamente el proceso de selección de los contenidos más convenientes para la realización de los OA, dando paso al Análisis y diseño de los OA para los contenidos seleccionados garantizando que estos tengan un nivel de interacción y experimentación adecuadas. Este capítulo contiene la selección de las herramientas a utilizar. En tercer y último capítulo titulado ***“Implementación de los objetos de aprendizaje su validación desde el punto de vista pedagógico y como software”***, se implementan los OA propuestos y se realizan las pruebas pertinentes a cada OA para verificar que estos cumplan con el objetivo planteado como software y desde el punto de vista pedagógico.

Capítulo I. Los objetos de aprendizaje y su utilización en el proceso de enseñanza aprendizaje de las Matemáticas Discretas

En el presente capítulo se precisan los elementos teóricos que sustentan la investigación. Se comienza realizando una caracterización del proceso de enseñanza aprendizaje de la Matemática Discreta en la UCI a partir de la importancia de sus contenidos para la formación de los especialistas vinculados a las TIC. Se analizan además los principales conceptos relacionados con los OA, su concepción, diseño y creación. El capítulo termina con diagnóstico del estado de la utilización de OA en el PEA de la MD en la UCI.

1.1 Importancia de la Matemática Discreta en la formación de los profesionales de las TIC

El estudio de conceptos como la continuidad y el cambio continuo es el objetivo de la matemática continua, sus procesos por lo general no son contables. Sus definiciones fundamentales son las de límites y suavidad en las curvas; al trabajar con número reales es imposible numerar por ejemplo los elementos que se encuentran entre el número 2 y el 3. 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).

Hablando sobre la importancia de la combinatoria (área fundamental de la MD) para la informática, Björne y R. P. Stanley plantearon: “El desarrollo reciente de la combinatoria es en cierto modo la historia de la cenicienta: los matemáticos ortodoxos la miraban por encima del hombro, considerándola menos respetable que otras áreas a pesar de sus muchos servicios tanto a la matemática pura como aplicada. Pero entonces llegó el príncipe de la informática con todos sus problemas y necesidades matemáticas, y la combinatoria fue a quien mejor le entraba el zapatito de cristal “ (Björne, y otros, 1999).

Desde el punto de vista de la Enseñanza de la Matemática 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 (Canavelli, y otros, 2007).

Tal y como expresa (Xiuguo, y otros, 2009) un curso de MD se debe centrar en la corrección, la lógica y los algoritmos, y la mayoría de los estudiantes tienen dificultades para aprender los contenidos de esta parte esencial de las matemáticas; por tal motivo los profesores deben mejorar la calidad del PEA de la MD.

En la informática y las telecomunicaciones se hace presente la MD: la información se manipula y almacena en las computadoras de forma discreta (palabras formadas por ceros y unos), se necesitan contar objetos (unidades de memorias, unidades de tiempo), se precisa estudiar relaciones entre conjuntos finitos (búsquedas en bases de datos), es necesario analizar procesos que incluyan un número finito de pasos (algoritmos) (Rosen, y otros, 2004). La MD ha pasado de ser una mera “colección de problemas sueltos y trucos de resolución” a tener una estructura definida y bien fundamentada, a convertirse en una asignatura estudiada en varias carreras universitarias.

El SIGCSE (Special Interest Group on Computer Science Education) es un grupo conformado por científicos de la ACM (Association for Computing Machinery) cuyo principal objetivo es investigar el desarrollo de la enseñanza de las CC a nivel mundial. En un reporte creado por un comité de dicho grupo en abril del 2007, Bill Marion y Doug Baldwin (Marion, et al., 2007) codirectores del mismo plantearon que *“...la matemática discreta es parte esencial en las ciencias de la computación, y su enseñanza debe mejorar cada día.”*

La ACM divide las carreras de la denominada Computing Curricula (CCU) en cinco (5) carreras: Ciencias de la Computación (CC), Ingeniería de Software (SE), Tecnologías de la Información (TI), Sistemas de Información (SI) e Ingeniería Electrónica (EI).

Al realizar un análisis de los currículos de las carreras de CC, SE, SI e Ingeniería Informática en ellas se estudian las estructuras discretas impartiendo los siguientes temas: funciones, relaciones binarias, teoría de conjuntos, lógica básica, técnicas de demostraciones, combinatoria, relaciones de recurrencia, grafos, árboles y probabilidad discreta. Además en la asignatura Teoría de Autómatas se imparten las Máquinas de Turing.

En Cuba se estudian tres (3) carreras asociadas a los temas computacionales: Ciencias de la Computación, Ingeniería Informática e Ingeniería en Ciencias Informáticas, vale aclarar que las ingenierías

son muy similares en contenido y objetivos; aunque las dos primeras tienen bien diferenciado su objeto de la profesión, se diferencian fundamentalmente en el peso del componente laboral. En las carreras que se estudian en el país también están presentes las estructuras discretas, los temas estudiados son: Lógica, Álgebra, Combinatoria, Grafos, Teoría de Números, Computabilidad, Diseño y Análisis de Algoritmos.

1.2 Caracterización del proceso de enseñanza aprendizaje de la Matemática Discreta en la UCI

Como expresa (Morales Peña, 2011) en su Tesis de Maestría en Enseñanza de las Matemáticas, el avance en el desarrollo de las ciencias y la tecnología involucra cada vez más teorías matemáticas. La MD se crea dada la necesidad de dominar nuevos contenidos y habilidades que surgen a partir del desarrollo de las CC. Por lo tanto es necesario hacer un estudio del PEA de MD, resaltando aquellos contenidos básicos que no deben faltar en el currículo de un especialista de los campos de las TIC. A través de los años con los avances de las CC, en busca de una MD centrada cada vez más en la corrección, la lógica y los algoritmos se le han añadido contenidos cada vez más modernos que complementan a la MD como una asignatura de vital importancia para la formación de profesionales de las TIC.

Como ya se mencionó en la introducción, la UCI centra su proceso educativo en el aprendizaje. Pretendiendo que el estudiante adquiriera un papel más activo y una mayor responsabilidad en el proceso de formación; el papel del profesor se enfoca a la planeación, orientación y la evaluación de las experiencias de aprendizaje de manera que los estudiantes construyen el conocimiento y su relación con el campo profesional. Los contenidos son pertinentes, más formativos que informativos, con un énfasis en contenidos significativos, el desarrollo de habilidades de aprendizaje a través de la actividad productiva y la investigación, así como la enseñanza de valores y el desarrollo de actitudes. La transferencia, análisis y manejo de la información se realiza haciendo un uso intensivo y racional de los medios tecnológicos modernos, incorporando en el plan de estudio diversas experiencias de aprendizaje que permiten al estudiante desarrollar los conocimientos, competencias, valores y actitudes que contribuyan a su pleno desarrollo individual y social (UCI, 2009).

La enseñanza de la MD en la UCI ha sufrido cambios, a lo largo del propio desarrollo de la ICI. En el primer curso solo se impartían los temas: lógica proposicional, lógica de predicado y diseño de algoritmos en la asignatura Lógica y Algoritmos, que desaparece en el siguiente curso, donde los contenidos vinculados a esta área desaparecen del currículo. La asignatura MD aparece por primera vez en el curso 2003-2004 con noventa (90) horas clases, distribuidas entre los temas de teoría de conjuntos, relaciones

binarias, lógica básica, técnicas de demostraciones, relaciones de recurrencia, combinatoria y teoría de grafos. Esta estructura se mantuvo hasta el curso 2008-2009, donde se comienza un rediseño, surgiendo así la MD2.

Durante este rediseño se introdujeron los temas circuitos lógicos, teoría de números y teoría de la Computabilidad; además de aumentar los contenidos de técnicas de demostraciones, relaciones de recurrencia, combinatoria y teoría de grafos; debido en gran medida a que los contenidos existentes hasta el momento no bastaban para que el estudiante fuera capaz de desarrollar su capacidad de análisis mediante el uso de la abstracción o idealización de condiciones para investigar estructuras y procesos informáticos y su capacidad para representar y razonar sobre sus propiedades y relaciones.

Teniendo en cuenta el informe semestral a nivel UCI de la asignatura MD1 en el curso 2009-2010, elementos que son distintivos además de los cursos anteriores (UCI, 2011) es preciso señalar:

- 1) Existen problemas con la aplicación, en muchos casos aún incierta, de métodos de evaluación efectiva que permitan una retroalimentación apropiada en el proceso de aprendizaje y una comunicación efectiva entre los diferentes actores de la clase.
- 2) Se pone de manifiesto una inadecuada planificación de las clases donde el tiempo no se distribuye satisfactoriamente para lograr el total cumplimiento de los objetivos, esto también se encuentra afectado por el gran cúmulo de contenidos a impartir en las actividades y el grado de complejidad de los mismos.
- 3) Uso deficiente de las TIC en el PEA de MD, el cual se limita al uso de presentaciones y de documentos publicados en el EVA-UCI.
- 4) Existen deficiencias en la impartición y el dominio de la teoría de la Computabilidad y la teoría de números, determinadas en gran medida por la inexperiencia en dichos temas y la formación de los profesores que la imparten.

Los contenidos de las MD presentan una variada complejidad, existen diferentes temas con los que de alguna manera el estudiante ha interactuado desde la enseñanza media superior. Por otra parte existen contenidos con un grado mucho mayor de complejidad. Asimismo ocurre desde el punto de vista del profesor donde varía el grado de complejidad de impartición de los contenidos, donde vale resaltar la complejidad del tema de teoría de la computabilidad por las habilidades de pensamiento creativo y

algoritmización que requiere; de igual manera es el contenido de más lenta asimilación por parte de los estudiantes.

A este punto de la investigación se puede concluir de manera parcial que el estudio de la asignatura MD es un elemento esencial en la formación de futuros profesionales de las TIC puesto que ayuda al desarrollo de habilidades que son necesarias para comprender muchos conocimientos relacionados con los procesos informáticos. Por otra parte se analizó la estructura del plan de estudios de la MD en la UCI así como los contenidos que se imparten en la misma, la cual se divide en dos asignaturas, MD1 y MD2 donde los temas estudiados son: Lógica, Álgebra, Combinatoria, Grafos, Teoría de Números, Computabilidad, Diseño y Análisis de Algoritmos, llegando a la conclusión de que existen deficiencias en el PEA de dichos contenidos. Uno de los elementos que distingue al PEA de la UCI es su EVA, por ello se hace necesario caracterizarlo y tener en cuenta sus posibilidades.

1.2.1 El entorno virtual de enseñanza aprendizaje de la Universidad de las Ciencias Informáticas.

Como alegan (López García, et al., s/f) Moodle está basado en un modelo pedagógico de construccionismo social. Su facilidad de uso y flexibilidad a la hora de diseñar diferentes tipos de cursos permite atender a diferentes demandas de los profesores, dependiendo de los objetivos previamente fijados. Moodle no delimita casi en absoluto las funciones propias de administración con las de profesor. Un profesor tiene casi todo el control sobre el curso que administra; incluso la gestión de los alumnos y del resto de profesores del mismo (de la Torre, 2006).

Así pues, se pueden establecer cursos según diferentes enfoques utilizando para ello los diferentes recursos y herramientas que se ofrecen: **Enfoque “tradicional:** Lugar donde “colgar” contenidos y recursos, basado en contenidos estáticos en Moodle: Páginas Web (HTML), Páginas de texto (sin formato), Enlaces web, Cualquier fichero “referenciable” mediante una URL (imágenes, audio, vídeo, etc.). **Enfoque interactivo/evaluador:** Basado en la interacción y evaluación de los alumnos mediante actividades de Moodle: Tarea (entregar un trabajo), Cuestionario (preguntas de diversos tipos), Consulta (pregunta a la clase), Encuesta (encuesta educativa preestablecida), Lección (contenidos interactivos). **Enfoque social:** Lugar de aprendizaje en común, haciendo énfasis en tareas cooperativas a través de las siguientes actividades de Moodle: Chat (charla en tiempo real), Foros (debates en la web), Glosario (vocabulario creado en común), Wiki (construcción de una web en común), Taller (cada alumno es evaluado por todos los demás).

Moodle hace una distinción entre los elementos que pueden ser usados: Recursos y Actividades. En principio, los recursos serían los elementos que permitirían a los alumnos acceder a los contenidos. Por otro lado las actividades serían las herramientas de trabajo para los mismos. No obstante, el carácter abierto de la mayoría de los elementos de Moodle hace que en ocasiones cueste trabajo distinguir entre ambas categorías. Los **recursos** son elementos que contienen información que puede ser leída, vista, bajada de la red o usada de alguna forma para extraer información de ella mientras que las **actividades** son elementos que se piden que se hagan basado en los recursos que se han utilizado (de la Torre, 2006). El objetivo de las actividades en Moodle se enfoca a la gestión del aprendizaje del estudiante, están pensadas para que el alumno trabaje y ejecute acciones, dando la posibilidad al profesor de seguir su progreso en el proceso de estudio y aprendizaje.

Desde el punto de vista técnico Moodle está construido sobre una base tecnológica basada en estándares, modular y extensible. Es una aplicación web basada en el lenguaje de programación PHP y una base de datos relacional (MySQL), que forman una pareja tecnológica muy difundida, robusta y ampliamente probada, de amplio uso en los programas Open Source (figura 1) (López García, et al., s/f).

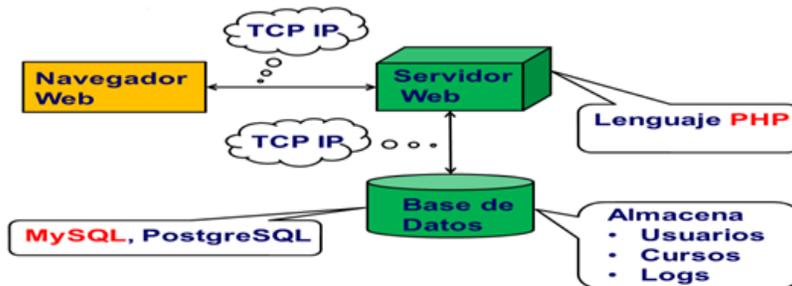


Figura 1. Fundamentos tecnológicos de Moodle (López García, et al., s/f).

Moodle presenta variadas características como son: Entorno de aprendizaje modular y dinámico orientado a objetos, sencillo de mantener y actualizar; excepto el proceso de instalación, no necesita prácticamente de mantenimiento por parte del administrador; dispone de una interfaz que permite crear y gestionar cursos fácilmente; los recursos creados en los cursos se pueden reutilizar; la inscripción y autenticación de los estudiantes es sencilla y segura; resulta muy fácil trabajar con él, tanto para el profesorado como el alumnado; detrás de él hay una gran comunidad que lo mejora, documenta y apoya en la resolución de problemas y está basado en los principios pedagógicos constructivistas: el aprendizaje es especialmente efectivo cuando se realiza compartiéndolo con otros (Soto García, et al., 2010).

Desde la perspectiva del programador, la, M de Moodle contiene un concepto muy importante, ya que Moodle, en su totalidad, está diseñado de forma modular. Esto permite al desarrollador realizar importantes modificaciones a la plataforma sin necesidad de cambiar el código fuente. La modularidad se convierte en una importante característica que permite reducir la cantidad de tiempo empleada en realizar modificaciones cuando se lanzan nuevas versiones de Moodle. (García Alba, 2010).

Los módulos son componentes auto-controlados que extienden las funcionalidades de una aplicación. Éstos deben ser fáciles de instalar y mantener, asegurando aislar el impacto que pueda tener un fallo sobre el resto del programa. La lógica del funcionamiento de Moodle se encierra en tres grandes grupos de módulos: los de comunicación, de recursos y de actividades. **Módulos de comunicación:** Estos módulos permiten que los alumnos puedan comunicarse con el profesor (hacer preguntas, plantear dudas, etc.) y entre ellos construir su propia comunidad de aprendizaje. **Módulos de recursos:** Estos módulos representan los contenidos y materiales del curso. Son todo tipo de textos, libros, apuntes, presentaciones de diapositivas, enlaces a páginas web externas etc., pensados para que los estudiantes los lean y estudien sobre ellos. **Módulos de actividades:** Estos módulos son la parte activa y colaborativa donde el estudiante tiene que hacer algo más que leer un texto. Debates y discusiones, resolución de problemas propuestos, redacción de trabajos, talleres, cuestionarios en línea, etc.

Existe una gran variedad de módulos para Moodle que constituyen el eje principal de este sistema de gestión de aprendizaje. Estos módulos dan la posibilidad al usuario de enriquecer el diseño de cursos, permitiendo calificar de forma automática o manual, trabajar de forma colaborativa, además de brindar la posibilidad de debates y consultas. Por defecto al terminar la instalación de la aplicación existen diecisiete módulos estándares, y otros, no estándares, pueden ser añadidos al sistema aprovechando la flexibilidad de su arquitectura. Estos últimos pueden ser descargados desde la base de datos "Modules and Plugins" del sitio oficial de Moodle¹ e instalados descomprimiendo el paquete dentro del directorio /mod en la raíz del sistema. Algunos de estos módulos son: Libro, Tarea, Foro, Chat, Consulta, Hot Potatoe, Etiqueta, Encuesta, Taller, Cuestionario, Recurso, Glosario, Wiki y Lecciones (Soto García, et al., 2010).

Un elemento importante dentro de los entornos virtuales son los repositorios de objetos de aprendizaje (ROA). Cuando se habla sobre OA se insiste en que la reutilización debe ser una de sus principales bondades, es entonces casi intrínseco que se tenga un lugar destinado para su almacenamiento y clasificación para facilitar posteriormente su mantenimiento, localización y, posiblemente, también

¹www.moodle.org

compartir ese OA con otros sistemas para aplicaciones diversas. Este almacén es lo que se conoce como Repositorio de Objetos de Aprendizaje. Los sistemas de repositorios son la infraestructura clave para el desarrollo, almacenamiento, administración, localización y recuperación de todo tipo de contenido digital.

De manera general se puede decir que los ROA son un tipo de bibliotecas digitales especializadas en recursos educativos que utilizan los estándares de metadatos que han desarrollado los organismos encargados de la estandarización del e-learning, preparadas tecnológicamente para interoperar con otros repositorios y con otras aplicaciones de los entornos e-learning. La interacción de los ROA debe ser posible con todos los sistemas, herramientas y usuarios que hagan uso de contenidos, así como con aquellos otros repositorios o recursos que puedan agregar a su catálogo o con los que puedan comunicarse para hacer posibles las búsquedas federadas.

Por la forma en la que se concentran los recursos, principalmente se identifican dos tipos de ROA: Los que contienen los objetos de aprendizaje y sus metadatos, en éstos los objetos y sus descriptores se encuentran dentro de un mismo sistema e incluso dentro de un mismo servidor y los que contienen sólo los metadatos, en este caso el repositorio contiene sólo los descriptores y se accede al objeto a través de una referencia a su ubicación física que se encuentra en otro sistema o repositorio de objetos. (López Guzmán, 2005)

A partir de lo esbozado en la introducción se hace necesario realizar un análisis de los principales conceptos relacionados con los OA y su utilización en el PEA, a partir del auge que ha tomado la utilización de las TIC en la educación, independientemente del nivel de enseñanza.

1.3 La utilización de objetos de aprendizaje en el proceso de enseñanza aprendizaje

En la actualidad el uso de las TIC se ha vuelto cotidiano en muchas esferas de la sociedad, un ejemplo de ello es la esfera de la educación donde se ha abierto paso un nuevo modelo de aprendizaje para elevar el nivel educativo y la capacitación de la población a nivel mundial. Una de las alternativas más prometedoras en este sentido es la utilización de OA.

El concepto de OA ha sido y es tema de investigación de muchas Instituciones, debido a ello su definición varía de acuerdo a las necesidades de cada organismo. Según plantea el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), los OA son *“una entidad, digital o no digital, que puede ser utilizada, reutilizada y referenciada durante el aprendizaje soportado con tecnología”* (IEEE, 2011). Mason y otros los definen como *“una pieza digital de material de aprendizaje que direcciona a un tema claramente identificable o*

salida de aprendizaje y que tiene el potencial de ser reutilizado en diferentes contextos” (Mason, y otros, 2003).

Otros dentro de los cuales se encuentra (Morales, 2005) señalan a los OA como unidades de aprendizaje independiente y autónomo que están predispuestos a su reutilización en diversos contextos instruccionales. Por otra parte existe una definición que parte de una diferenciación de conceptos un poco más elaborada definiendo primero un objeto mediático como un conjunto de bits de texto, gráficos, video o audio. Si a un objeto mediático se le asigna un valor como conocimiento para un sujeto o sujetos, entonces se le puede considerar como objeto de conocimiento y cuando a un objeto de conocimiento se le integra una estrategia instruccional, se obtiene un objeto de aprendizaje (Merrill, 1999).

La Corporación Universitaria para el Desarrollo de Internet 2, de México considera a un OA como: *“una entidad informativa digital desarrollada para la generación del conocimiento, habilidades y actitudes requeridas en el desempeño de una tarea que tiene sentido en función de las necesidades del sujeto que lo usa y que representa, y se corresponde con una realidad concreta susceptible de ser intervenida”* (Chan, 2011).

Se define además un objeto de aprendizaje (OA) *“como una pieza digital de material educativo, cuyo tema y contenido puede ser claramente identificable y direccional, y cuyo principal potencial es la reutilización dentro de distintos contextos aplicables a la educación virtual”* (Ruiz, 2006). Por otro lado se conoce un objeto de aprendizaje *“como un recurso digital modular identificado de manera única y etiquetado con metadatos, que puede ser utilizado para el apoyo de la enseñanza”* (Beck, 2008).

Teniendo en cuenta la variedad de las definiciones, así como la diversidad de recursos que pueden considerarse como OA se puede concluir para términos de esta investigación que se considerará como OA la definición planteada por (Ruiz, 2006) al considerarlos *“como una pieza digital de material educativo, cuyo tema y contenido pueden ser claramente identificable y direccional, y cuyo principal potencial es la reutilización dentro de distintos contextos aplicables a la educación virtual”*. Esta es una propuesta conceptual más sólida y menos abierta a discusión y la que más se ajusta al objetivo que se persigue en el marco de la presente investigación.

Según (Fuentes, et al., 2009) los OA pueden ser clasificados atendiendo al tipo de contenido pedagógico y al formato.

Según el contenido pedagógico 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.

Según el formato: Imagen, Texto, Sonido y Multimedia.

Las principales características que presentan los OA, son las siguientes (López, 2005):

- **Reutilizables:** El recurso debe ser modular para servir como base o componente de otro recurso. También debe tener una tecnología, una estructura y los componentes necesarios para ser incluido en diversas aplicaciones.
- **Accesibles:** Pueden ser indexados para una localización y recuperación más eficiente, utilizando esquemas estándares de metadatos.
- **Interoperables:** Pueden operar entre diferentes plataformas de hardware y software.
- **Portables:** Pueden moverse y albergarse en diferentes plataformas de manera transparente, sin cambio alguno en estructura o contenido.
- **Durables:** Deben permanecer intactos a las actualizaciones de software y hardware.

La clasificación de los Objetos de Aprendizaje según Wiley de acuerdo a sus características puede ser de diferentes formas, en (Wiley, 2000) se explica que los Objetos de aprendizaje se pueden dividir en cinco (5) grupos a través de la siguiente taxonomía.

Características de los OA	OA fundamentales	OA combinado-cerrado	OA combinado-abierto	OA generativo-presentación	OA generativo-educacional
Número de elementos combinados	Uno	Pocos	Varios	Poco-Varios	Pocos-Varios
Componentes reutilizables del Objeto	(No aplica)	No	Si	Si/No	Si/No
Funciones comunes	Exhibir	Educacional o	Educacional	Exhibir	Educacional y/o

	Mostrar	practica prediseñada	y/o practica prediseñada	Mostrar	practica generada por computadora
Potencial para la reutilización intercontextual	Alto	Medio	Bajo	Alto	Alto
Potencial para la reutilización intracontextual	Bajo	Bajo	Medio	Alto	Alto

Tabla 1. Taxonomía de objetos de aprendizaje elaborada por (Wiley, 2000).

En (Toll Palma, 2011) se afirma que a la hora de diseñar los OA deben considerarse una serie de características como son:

- *Formato digital*: tiene capacidad de actualización y/o modificación constante; es decir, es utilizable desde Internet y accesible a muchas personas simultáneamente y desde distintos lugares.
- *Propósito pedagógico*: el objetivo es asegurar un proceso de enseñanza-aprendizaje satisfactorio. Por tanto, el OA incluye no sólo contenidos sino que también guía el propio proceso de aprendizaje del estudiante.
- *Contenido interactivo*: implica la participación activa de cada individuo (profesor-estudiante/s) en el intercambio de información. Para ello es útil que el objeto incluya actividades (ejercicios, simulaciones, cuestionarios, diagramas, gráficos, diapositivas, tablas, exámenes, experimentos, entre otros) que permitan facilitar el proceso de asimilación y el seguimiento del progreso de cada estudiante.

Los OA son recursos válidos para desarrollar competencias lo que, a su vez, implica considerar aspectos de contenidos y procesos de aprendizaje que atienden a la calidad de la enseñanza y, por ende la optimización de los aprendizajes, en el mundo tecnológico. Desde esta perspectiva el diseño de un OA resulta un desafío para el docente, quien debe elegir el contenido, crear las formas de presentación, apoyándose en las características de los usuarios o destinatarios y la ubicación de éste en el espacio. El diseño viene a fragmentar el proceso de aprendizaje en una serie de actividades, lo que permite tener un gran número de combinaciones que pueden generarse a partir de todos los OA que constituyen un curso. De esta forma facilita la flexibilización de las estrategias de aprendizaje.

Por otra parte se hace necesario señalar que, según el diseño de un OA debe contener al menos tres características básicas: a) ser referenciable, para lo cual debe ser rotulado o etiquetado, para garantizar su

acceso; b) ser reutilizable, lo que implica que debe adaptarse a diferentes contextos de aprendizaje y c) finalmente, ser independiente del medio en el que se los proporciona y del sistema que los utilizará, garantizando su interoperabilidad (Ossadán Núñez, 2006).

Una vez descrito el OA, mediante metadatos se catalogan en repositorios de OA para quedar disponibles al público, y ser incorporados a diferentes experiencias del aprendizaje. La finalidad de promover la creación y disposición de recursos para el aprendizaje, supone una visión de acumulación del capital académico que se da por ejercicio de la enseñanza, y que puede trascender el uso individual en el aula al ponerlo a disposición en red. Los recursos se comparten y como resultado de esta gestión se pueden abatir costos y multiplicar los beneficios o impactos de un objeto al será provechado por un mayor número de usuarios (Chan, 2011).

1.3.1 Metodologías para el desarrollo de Objetos de Aprendizaje

Aunque la base de la metodología para el diseño de OA es el modelo pedagógico, estos se pueden considerar 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).

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software” (Piattini, 1996). La finalidad del uso de metodologías es lograr el desarrollo de un software de calidad.

Dentro de las clasificaciones existentes para dichas metodologías sobresalen dos: las metodologías tradicionales y las ágiles. Las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, son generalmente aplicadas a grandes proyectos e indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben detener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. Por otra parte se encuentran las llamadas metodologías ágiles que se centran más en la obtención del sistema sin importar cuán documentado esté el mismo, generalmente propone

que los involucrados participen en el proceso de desarrollo para ir mejorando las funcionalidades del sistema en producción, así como el trabajo en parejas.

1.2.1.1. Extreme Programming (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. XP se basa en realimentación 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Las características esenciales de XP organizadas en los tres grupos: historias de usuario, roles, proceso y prácticas.

La metodología se basa en: Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Programación en 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 -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata. Frecuente integración 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 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 (H Canós, et al., s/f).

1.2.1.2. Rational Unified Process (RUP)

RUP es un proceso de Ingeniería cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Cubre el ciclo de vida de desarrollo de software. RUP presenta 3 características esenciales: *Dirigidos por casos de uso*: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, estos representan los requisitos funcionales. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso. *Centrado en la Arquitectura*: La arquitectura muestra la visión común del sistema completo en la que el equipo de trabajo y los usuarios deben de estar de acuerdo, por lo que describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. *Iterativo e incremental*: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

RUP divide el proceso de desarrollo en 4 fases: *Inicio*: El Objetivo en esta etapa es determinar la visión del proyecto. *Elaboración*: En esta etapa el objetivo es determinar la arquitectura óptima. *Construcción*: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial. *Transmisión*: El objetivo es llegar a obtener el release del proyecto.

La visión de la estructura del ciclo de vida RUP se basa en un desarrollo iterativo, concretado por hitos para revisar el avance y planear la continuidad o los posibles cambios de rumbo. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo: Ingeniería de Negocios: Se entiende las necesidades del negocio. *Requerimientos*: Se traslada las necesidades del negocio a un sistema automatizado. *Análisis y Diseño*: Se trasladan los requerimientos dentro de la arquitectura de software. *Implementación*: Se crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado. *Pruebas*: Se asegura que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte: Configuración y administración del cambio: Se guardan todas las versiones del proyecto. *Administrando el proyecto*: Se administran horarios y recursos. *Ambiente*: Se administra el ambiente de desarrollo. *Distribución*: Se hacen todo lo necesario para la salida del proyecto Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que

cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son: *Actividades* (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos. *Trabajadores* (“quién”): Vienen hacer las personas o entes involucrados en cada proceso. *Artefactos* (“qué”): Un artefacto puede ser un documento, un modelo, o un elemento de modelo. *Flujo de actividades* (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable. Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software (Martínez, et al., s/f).

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

ISD-MeLO fue propuesta 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. Esta metodología está enfocada en el 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. Esta metodología propone cinco fases: Análisis, Diseño, Desarrollo, Implementación y Evaluación (Pernalet Chirinos, et al., 2009).

Fases	Actividades
Análisis	Análisis del perfil del aprendiz. Análisis del problema. Búsqueda de objetos de aprendizaje que se ajusten 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. Categorizar los objetos de aprendizaje. Especificar los objetos de aprendizaje. Mantener la metadata 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 <u>interfaz</u> . Desarrollar prototipo y evaluar. Mantener la metadata en el diseño de la interfaz.

Fases	Actividades
Desarrollo	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. Mantener la metadata.
Implementación	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	Llevar a cabo una evaluación formativa. Llevar a cabo una evaluación sumativa.

Tabla 2 Metodología ISD-MeLO (Fases y Actividades).

Luego de un análisis de las metodologías anteriormente descritas se llegó a las siguientes consideraciones. RUP garantiza la elaboración de todas las fases de un producto software orientado a objeto. Permite realizar la documentación del software durante todo el proceso de desarrollo, pero posee como inconveniente que es para desarrollar proyectos de gran envergadura y por lo general los OA no son grandes proyectos.

Por su parte XP es una metodología ágil que sirve para proyectos de corto plazos además la programación extrema es una forma ligera, eficiente, flexible, predecible y científica de generar software. Esta metodología ha surgido desde la experiencia, como una forma de resolver los problemas encontrados en los procesos de desarrollo software en los que se han visto involucrados sus autores. XP es mucho más fácil de implementar y de aprender. Mientras que en RUP la entrega final debe de ser algo mucho más definido en XP se realizan entregas continuas y discretas que permiten evaluar el sistema por lo que se puede concluir que XP presenta muchos elementos que pueden ser útiles para el desarrollo de OA.

La metodología ISD-MeLO es usada en el desarrollo de objetos de aprendizaje. Permite el desarrollo de éstos de una manera organizada de acuerdo a las tareas propuestas en cada fase, logrando así que queden con la calidad requerida, 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. Tiene un enfoque ecléctico desde la perspectiva de las Teorías de Aprendizaje, lo cual le proporciona flexibilidad para aplicarse en una amplia diversidad de situaciones de aprendizaje. El gran inconveniente que presenta esta metodología es que no contempla al OA como una pieza de software por lo que no es del todo útil para cumplir el objetivo de la presente investigación.

1.3.2 Valoración de herramientas y lenguajes utilizados en la creación de objetos de aprendizaje

Para el desarrollo de los OA es necesario el uso de diferentes herramientas, necesarias para el cumplimiento del objetivo planteado en la presente investigación. A continuación se realiza una valoración de algunas de estas herramientas así como los lenguajes de programación que pueden ser usados para la creación de los OA. Se hace necesario además el análisis de diferentes parámetros e indicadores que tasan la calidad del producto resultante, por lo que se realiza una valoración de los procesos de evaluación de la calidad en el proceso de la creación de los OA.

1.3.2.1 Análisis de algunas herramientas utilizadas para el desarrollo de objetos de aprendizaje

Macromedia Flash 8.0.

Macromedia Flash es una herramienta orientada a crear aplicaciones y contenidos dinámicos, es decir, utilidades interactivas y multimedia. Este programa posibilita la utilización tanto de gráficos vectoriales como imágenes de mapa de bits y un lenguaje de programación propio llamado Action Script con el que se pueden crear aplicaciones más complejas, además cuenta con una gran variedad de métodos y funciones ya implementados que hacen más ágil y cómodo el trabajo. Flash permite incluir audio comprimido en diversos formatos como el mp3, importar gráficas creadas con otros programas, formularios y algo de programación. Todo esto definido al igual que los vectores por un conjunto de instrucciones que mueven los objetos de posición y forma, y que dan como resultado archivos muy pequeños que se cargan en poco tiempo. Flash lleva un sistema de ayuda muy completo y fácil de entender y no resulta difícil manejarlo correctamente. Para determinados efectos, es rápido y cómodo, mucho más conveniente que la alternativa de un gif estático o animado (Fuentes, et al., 2009).

ToolBook

Es una herramienta de autor que permite la creación de aplicaciones multimedia combinando textos, imágenes, sonidos, animaciones, vínculos a sitios Web y autoevaluaciones en distintos formatos. Posee el lenguaje de programación Open Script orientado a objetos que enriquece extraordinariamente sus posibilidades en la generación de aplicaciones multimedia. Existen dos formas de trabajar con ToolBook: el nivel autor (es el que permite diseñar y modificar la aplicación multimedia) y el nivel lector (que es donde se ejecuta la misma). Para facilitar su trabajo dispone de numerosas plantillas que agilizan la creación de multimedia, incluyen botones predefinidos de navegación y un asistente de script de programación, asociado a las acciones más comunes que realiza un usuario en una aplicación. Permite compartir guiones

entre distintos objetos, sin necesidad de tener que reescribir el código fuente.

Algunas de las Características de ToolBook son: interfaces amigables y fáciles de usar, posee asistentes, plantillas y un catálogo de objetos reusables que ayudan a crear una aplicación en horas no semanas, soporta un amplio rango de elementos multimedia (Microsoft PowerPoint, PDF, audio, video, Macromedia Flash, Java applets etc.) haciéndolo fácil de incorporar en contenidos de aprendizaje online, se puede usar en diferentes medios: HTML, CD-ROM, red con o sin conocimiento de sistemas de manejo de contenidos y se pueden incorporar scripts, simulaciones, software de entrenamiento y más (Fuentes, et al., 2009).

Macromedia Authorware.

Es un programa que permite crear y presentar información interactiva, de acuerdo a las necesidades de cada persona o empresa; se usa principalmente para el área de la educación presencial o virtual. Muchas de estas aplicaciones se ponen a disposición en las páginas Web o solamente son de carácter privado, aunque también se usan para fines comerciales, por ejemplo, anuncios de publicidad, y para presentaciones hechas en Power Point, es decir, transformarlas para que tengan más efectos multimedia, de ese modo tienen más animaciones y son mucho más interactivas, pero la mayoría de veces este programa es usado para el campo del aprendizaje.

Macromedia Authorware combina imágenes, audio, texto, video, animación, en fin tiene muchas posibilidades, y además permite importar archivos de Macromedia Flash, Internet, etc., e inclusive exportar aplicaciones de Authorware como XML (Lenguaje de Marcado Extensible). Como se ve tiene muchas ventajas y se puede trabajar en formato de CD, DVD (películas), MP3, etc. Es una alternativa a la hora de desarrollar aplicativos de enseñanza para colegios, universidades y programas de capacitación empresarial, muy fáciles de manejar, integrando diferentes herramientas multimedia y creando una interacción entre el programa y el usuario (Fuentes, et al., 2009).

NeoBook

NeoBook es un Software de autor de gran difusión en el ámbito educativo, que goza de mucha popularidad debido a su facilidad de uso y bajo costo. En el campo informático se entiende como herramienta de autor, a todo software que permite crear aplicaciones independientes del software que lo generó. Estas aplicaciones son programas o archivos ejecutables (del tipo *.EXE). Hoy día la definición es más restrictiva, puesto que se sobreentiende que una herramienta de autor puede manejar elementos multimedia (texto, imagen estática, imagen dinámica, sonidos y vídeos) y enlaces hipertextuales

(hipertextos e hipervínculos). De esta forma, un documento de Word, de Word Perfect o una imagen, no son el resultado de utilizar una herramienta de autor. En resumen, el elemento común a las herramientas de autor es el hecho de crear ejecutables que corren independientes del software que los generó, habiendo un proceso de compilado de por medio (Fuentes, et al., 2009).

ExeLearning.

El editor eLearning XHTML (eXe) es un programa de Autor para el desarrollo de contenidos. Esta herramienta está especialmente indicada para profesionales de la educación (profesores y diseñadores instruccionales), dedicada al desarrollo y publicación de materiales de enseñanza y aprendizaje a través de la web. Al ser una herramienta de autor evita al usuario que quiere crear contenidos basados en la web el tener que tener conocimientos previos de los lenguajes de programación necesarios para desarrollar contenidos en Internet (HTML, XHTML, XML, JavaScript, Ajax, etc.). Actualmente gracias a la implementación de nuevas herramientas tecnológicas aplicadas a la educación podemos usar internet como un instrumento más para el desempeño de nuestra actividad académica. Ofreciendo a los profesores y estudiantes oportunidades para que simultáneamente presenten contenidos y a su vez medios para interactuar con dichos contenidos (Fuentes, et al., 2009).

Entornos de desarrollo integrados

Las herramientas analizadas anteriormente, son las más usadas para el desarrollo de los OA clásicos, que se limitan al desarrollo de multimedia, animaciones flash, libros animados, presentaciones animadas etc. Cuando se considera al OA como una pieza de software, es necesario para su desarrollo el uso de otras herramientas mucho más robustas como son los Entornos de desarrollo integrados.

Un entorno de desarrollo integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos.

Netbeans.

Es un entorno de desarrollo integrado 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: Mac OS, 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, etc. (Oracle Corporation).

Microsoft Visual Studio Ultimate 2010

Entorno integrado de desarrollo (IDE) privativo. Permite programar en los diferentes lenguajes .NET. Dispone de herramientas y utilidades para editar y manipular distintos tipos de ficheros. Microsoft Visual Studio 2010 Ultimate² incluye potentes herramientas que simplifican todo el proceso de desarrollo de aplicaciones de principio a fin. Los equipos pueden observar una mayor productividad y ahorro de costes al utilizar características de colaboración avanzadas, así como herramientas de pruebas y depuración integradas que ayudarán a crear siempre un código de gran calidad. Visual Studio 2010 Professional es un entorno integrado que simplifica la creación, depuración e implementación de aplicaciones (Microsoft, 2010).

1.3.2.2 Valoración de los lenguajes para el desarrollo de objetos de aprendizaje

C++

Es un lenguaje imperativo y orientado a objetos derivado del C. En realidad un superconjunto de C, que nació para añadirle cualidades y características de las que carecía. Se puede decir que C++ es un lenguaje híbrido, ya que permite programar tanto en estilo procedimental (como si fuese C), como en estilo orientado a objetos, como en ambos a la vez. Además, también se puede emplear mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario. Las principales ventajas que presenta el lenguaje C++ son: difusión: al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, páginas web, etc. Algunas de las características de C++ son: Versatilidad: C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema, portabilidad: el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas, Eficiencia: C++ es uno de los lenguajes más rápidos en cuanto ejecución, herramientas: existe una gran cantidad de compiladores, depuradores, librerías, etc. (Luján Mora, s/f).

Java

²<http://www.microsoft.com/visualstudio/es-es>

Java es un lenguaje de programación orientado a objetos, creado por Sun Microsystems³ 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, etc. 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 Java Virtual Machine, que se encarga de interpretar el código (archivos compilados .class) y convertirlo a código particular de la CPU que se esté utilizando (siempre que se soporte dicha máquina 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).

C#

C# ("C Sharp") es un lenguaje de programación orientado a objetos de propósito general diseñado, desarrollado y estandarizado por Microsoft⁴ como parte de su plataforma .NET. Es un lenguaje con una sintaxis muy similar a la de C++ y Java. Combina la alta productividad de Visual Basic con el poder y la flexibilidad de C++. Algunas de las características de C# son: la misma aplicación que se ejecuta bajo Windows podría funcionar en un dispositivo móvil de tipo PDA. Con C#.NET uno no se ata a ninguna plataforma en particular, se puede crear una gran variedad de aplicaciones en C#: aplicaciones de consola, aplicaciones para Windows con ventanas y controles, aplicaciones para la Web, etc. C# gestiona automáticamente la memoria, y de este modo evita problemas de programación tan típicos en lenguajes como C o C++. Mediante la plataforma .NET es posible interactuar con otros componentes realizados en otros lenguajes .NET de manera muy sencilla, también es posible interactuar con componentes no gestionados fuera de la plataforma .NET. Por ello, puede ser integrado con facilidad en sistemas ya creados, desde C# se podrá acceder a una librería de clases muy completa y muy bien diseñada, que permitirá disminuir en gran medida los tiempos de desarrollo (s/f).

PHP

PHP (acrónimo de: Hypertext Preprocessor). Es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales (PHP, 2011). Con PHP se puede procesar la información de formularios y generar páginas

³<http://www.oracle.com/us/sun/index.htm>

⁴www.microsoft.com

con contenidos dinámicos. PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y Open BSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. De modo que, con PHP tiene la libertad de elegir el sistema operativo y el servidor web de su gusto. Además, tiene la posibilidad de utilizar programación por procedimientos, programación orientada a objetos (POO), o una mezcla de ambas. PHP también cuenta con soporte para comunicarse con otros servicios usando una gran variedad de protocolos dentro de los que se pueden mencionar a: LDAP, IMAP, SNMP, NNTP, POP3, HTTP y COM entre otros (PHP, 2011).

1.3.3 Procesos de evaluación de la calidad de los objetos de aprendizaje

En su investigación (Toll Palma, 2011) alega que el enfoque principal en la evaluación de los OA se centra en la funcionalidad, basada en la estética y facilidad de uso de los recursos, mientras que en la presente investigación se aboga por que la atención esté dirigida a tener en cuenta tanto la parte formativa, la de diseño y presentación como la tecnológica de los OA, así como al cumplimiento de los estándares que dirijan el sentido de la evaluación. De igual manera, todas las propuestas definen la evaluación en un momento previo a la interacción del objeto con los estudiantes, lo que permite crear hipótesis respecto al grado de impacto que tendrán éstos en el proceso de aprendizaje.

Para cada uno de los componentes de los OA existen criterios de evaluación afines, entre los que se encuentran:

- Aspectos de los contenidos temáticos. Congruencia y veracidad de los contenidos, calidad explicativa, caducidad de los contenidos.
- Aspectos del diseño instruccional. Facilidad de navegación, buen uso de recursos audiovisuales, aseguramiento de metas pedagógicas mediante la aplicación de competencias taxonómicas.
- Aspectos del metadato. Validez y congruencia de la información descrita, uso de estándares, utilidad y relevancia de los campos descritos.

En los últimos años, los objetos de aprendizaje han cobrado un auge increíble debido al impulso de las instituciones de educación superior sobre la educación a distancia como una solución al obstáculo representado en el factor espacio-tiempo dentro de los procesos de enseñanza-aprendizaje tradicionales. Unido a esto, debe recalcar el hecho de que los objetos de aprendizaje, como tecnología educativa, son

relativamente nuevos y se encuentran aún en una intensa etapa de estudio, lo cual permite desarrollar y proponer metodologías para su desarrollo enfocadas a diferentes cualidades.

Esta situación permite el establecimiento de criterios para garantizar la calidad de los OA, pues el ajustarse a diferentes convenciones de desarrollo, propician diversas definiciones de calidad, para ello deben establecerse criterios que permitan garantizar la calidad de los recursos creados sin importar la metodología empleada en su desarrollo, el contexto de uso en el cual se aplicarán, ni la temática tratada en sus contenidos.

(Vidal Castro, 2008) Sugiere una adaptación de la norma ISO 9126 sobre la evaluación de la calidad, para evaluar la calidad de los objetos de aprendizaje. La misma está definida por características y sub-características como: Funcionalidad (Adecuación, Exactitud, Interoperabilidad, Conformidad y Recuperabilidad). Usabilidad (Comprensibilidad, Facilidad de aprendizaje, Operatividad y Conformidad). Eficiencia (Tiempo de respuesta y Utilización de recurso). Reutilización (Reutilización de contenido y Diseño reutilizable). Portabilidad (Facilidad de Instalación, Facilidad de Adaptación y Coexistencia).

Se analizan propuestas de instrumentos de evaluación de la calidad para ser utilizados en los OA, uno de ellos es el dado por el autor (Vidal Castro, 2008).

Instrumento de Revisión de Objeto de Aprendizaje (LORI) proporciona un marco de evaluación de los OA basado en el análisis de nueve dimensiones. Cada dimensión se evalúa mediante una escala de cinco niveles. Se investigó la fiabilidad de este instrumento. Las nueve dimensiones de los OA evaluadas por este instrumento son:

1. **Calidad del Contenido:** se refiere a la exactitud y veracidad en la presentación de las ideas y un apropiado nivel de detalle.
2. **Alineamiento de los objetivos de aprendizaje:** relacionado con el alineamiento entre objetivos de aprendizaje, apoyos, evaluación y características del estudiante.
3. **Retroalimentación y adaptación:** es la adaptación del contenido y retroalimentación dada por estudiantes.
4. **Motivación:** se refiere al potencial para motivar e interesar a la comunidad de estudiantes o usuarios.
5. **Diseño de Presentación:** es el diseño de información visual y auditiva que faciliten el aprendizaje y un eficiente procesamiento mental.

6. **Usabilidad en la interacción:** relacionado con la facilidad de la navegación, predictibilidad y calidad de las interfaces de usuario.
7. **Accesibilidad:** referido al diseño de los formatos de control y presentación para facilitar el acceso a todo tipo de estudiantes.
8. **Reusabilidad:** es el grado en que el objeto puede ser usado en diferentes contextos de aprendizaje y con estudiantes de diferente formación.
9. **Cumplimiento de Estándares:** es el cumplimiento con los estándares y especificaciones internacionales.

Este instrumento es factible para evaluar algunos aspectos de los OA, fundamentalmente desde el punto de vista formativo y de diseño, pero sería ideal profundizar en los aspectos tecnológicos, para poder obtener OA completos y que cumplan con los indicadores de calidad que requiere.

Otro instrumento es el que propone (Morales, 2005) en él se detallan criterios para evaluar los OA. Estos criterios se agrupan en cuatro dimensiones:

1. **Aspectos psicopedagógicos** (capacidad de motivación y adecuación a los destinatarios)
2. **Aspectos didáctico-curriculares** (promueve habilidades meta cognitivas, participación activa, plantea actividades problemáticas, aprendizaje por descubrimiento, promueve trabajo colaborativo y cooperativo y es reutilizable)
3. **Aspectos técnicos-estéticos** (colores, tamaño, resolución y diseño de interfaz)
4. **Aspectos funcionales** (facilidad de uso, accesibilidad, eficacia, publicidad, navegación, velocidad, etc.)

Mediante estos criterios de valoración los autores anteriormente mencionados proponen cómo realizar una evaluación integral de los OA desde la perspectiva pedagógica, curricular, técnica y funcional. Están bastante completos, pues incluye aspectos formativos, tecnológicos y de diseño y presentación.

A partir del análisis crítico sobre la evaluación de la calidad de los OA, se concluye que algunos indicadores incluyen aspectos formativos, tecnológicos y de diseño para evaluar los OA producidos, pero a todos de una manera u otra le faltan parámetros que permitan evaluar completamente todos los elementos que lo conforman. A raíz de esta profundización teórica se analizaron algunos de los beneficios propuestos por (Velázquez, 2005) sobre el manejo adecuado de la calidad que pueden contribuir a la creación de los

OA y al mejor desarrollo del proceso de enseñanza-aprendizaje, entre ellos se puede mencionar los siguientes:

- Creación de metadatos más robustos y descriptivos desde el punto de vista de la calidad de contenido.
- Contar con bases para definir un estándar de calidad en la creación de OA en las instituciones educativas.
- El desarrollo de OA con una calidad de contenido controlada puede impactar positivamente en el proceso de enseñanza-aprendizaje tanto en cursos presenciales como en educación a distancia.
- Definir los aspectos que determinan la calidad del contenido de objetos de aprendizaje facilitará la búsqueda, creación y la utilización de información por medios electrónicos.

Por su parte (Toll Palma, 2011) propone una guía para la evaluación de la calidad que está dada por tres apartados: el primer apartado está conformado por los indicadores de evaluación agrupados en cuatro aspectos: el aspecto general, el aspecto formativo, el aspecto de diseño y presentación y el aspecto técnico. Definiendo en conjunto 40 indicadores de evaluación. Se ha considerado que todos los indicadores deben ser de cumplimiento básico para cualquier OA. Para los 40 indicadores agrupados en los tres aspectos se utiliza la escala de Excelente, Bien, Regular y Mal, donde se le otorga una puntuación de 3, 2, 1 y 0 respectivamente. El objetivo fundamental es poder a partir de estas puntuaciones evaluar cada uno de los indicadores de manera independiente y luego con el uso de las métricas obtener una evaluación final que permita evaluar el OA 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.

Apartado 1. Indicadores de evaluación

Aspecto formativo. En el aspecto formativo se tienen en cuenta un grupo de indicadores que describen el comportamiento de los OA desde el punto de vista pedagógico. Al evaluarlos, debe considerarse el aporte de los mismos en el desarrollo del aprendizaje

Aspecto de Diseño y Presentación. Se muestra el aspecto de diseño y presentación donde se evalúan los indicadores, fundamentalmente enfocados al diseño estético de la interfaz que visualiza el estudiante al interactuar con el OA, sus funcionalidades, la organización del contenido y la calidad de los recursos empleados (audio, imágenes y textos). Es válido resaltar que la calidad de estos indicadores tributa

directamente en el nivel de aceptación que puede llegar a tener el OA, de ahí la importancia que reviste tenerlos presente durante la revisión.

Aspecto tecnológico. Se relacionan los indicadores del aspecto técnico, cuyo nivel de importancia es significativo pues responden directamente a las funcionalidades del OA para un correcto funcionamiento y transmisión de los contenidos a los estudiantes. Conjuntamente con el resto de los indicadores de los aspectos formativo y de diseño y presentación, permiten dictaminar el nivel de calidad alcanzado por el OA antes de ser presentado a los estudiantes.

Apartado 2. Métricas. El apartado lo conforman las métricas utilizadas para medir la calidad de los OA, las mismas están dadas por los indicadores relacionados a cada uno de los aspectos anteriormente expuestos. Cada uno de los aspectos (formativos, diseño y tecnológicos) son evaluados en Excelente, Bien, Regular y Mal, que llevados a una representación numérica, le correspondería los valores 3, 2, 1 y 0 respectivamente.

Apartado 3. Nivel de calidad alcanzado

Rangos de escala	Nivel de Calidad alcanzado por el OA
$E \geq 36$	Muy Adecuado
$31 \leq E < 36$	Adecuado
$24 \leq E < 31$	Poco Adecuado
$E < 24$	No Adecuado

Tabla 3. Escala de nivel de calidad

Una vez analizada la calidad del OA teniendo en cuenta los diferentes indicadores analizados previamente (aspecto general, el aspecto formativo, el aspecto de diseño y presentación y el aspecto técnico), se hace necesario analizar otros aspectos que permitan verificar la calidad de los OA como producto de software para garantizar su correcto funcionamiento, es por ello que como complemento de la evaluación de la calidad del OA se realizaran pruebas de software.

En (Universidad Carlos III de Madrid, s/f) se plantea que las pruebas son prácticas a realizar en diversos momentos de la vida del sistema de información para verificar:

- El correcto funcionamiento de los componentes del sistema.
- El correcto ensamblaje entre los distintos componentes.

- El funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.
- El funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación.
- Que el sistema cumple con el funcionamiento esperado y permite al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.
- Que los cambios sobre un componente de un sistema de información, no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados.

Niveles de pruebas

- Pruebas Unitarias: Las pruebas unitarias constituyen la prueba inicial de un sistema y las demás pruebas deben apoyarse sobre ellas.
- Pruebas de Integración. El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.
- Pruebas del Sistema. Las pruebas del sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.
- Pruebas de Implantación. El objetivo es comprobar el funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación, y permitir al usuario que, desde el punto de vista de operación, revise el sistema en base al cumplimiento de los requisitos no funcionales especificados.
- Pruebas de Aceptación. El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.
- Pruebas de Regresión. El objetivo de las pruebas de regresión es eliminar el efecto onda, es decir, comprobar que los cambios sobre un componente de un sistema de información, no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados.

Enfoques de diseño de pruebas

Enfoque estructural o de caja blanca. Se verifica la estructura interna del componente con independencia de la funcionalidad establecida para el mismo. Por tanto, no se comprueba la corrección de los resultados si éstos se producen.

Objetivo: Probar el funcionamiento de la estructura de control de las unidades de programación.

- Garantizan que se ejecutan una vez por lo menos todos los caminos independientes de cada módulo.
- Prueban todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecutan todos los bucles.
- Ejecutan todas las estructuras internas.

Enfoque funcional o de caja negra. Se comprueba el correcto funcionamiento de los componentes del sistema de información, analizando las entradas y salidas y verificando que el resultado es el esperado.

- Las pruebas de caja negra se centran en los requisitos funcionales del software
- Comprobar que la funcionalidad del sistema es completamente operativa.
- Que la entrada se acepta de forma adecuada y la salida es correcta.
- Verificar que la integridad de la información interna se mantiene.

1.4 Diagnóstico de la utilización objetos de aprendizaje en el proceso de enseñanza aprendizaje de la Matemática Discreta en la Universidad de las Ciencias Informáticas

La UCI cuenta con un repositorio de objetos de aprendizaje, es un espacio creado para apoyar a la comunidad universitaria en la gestión de recursos didácticos, utilizando la tecnología de OA. Es un lugar para el trabajo metodológico colaborativo, orientado a elevar la calidad de los recursos didácticos. Haciendo un análisis de los OA dedicados a la MD publicados en dicho espacio, se puede notar la deficiente cantidad de los mismos y su bajo grado de orientación. El grado de interacción con los estudiantes de estos OA es casi nulo permitiendo solo la presentación de contenido estático mayormente, donde no se proporciona al usuario facilidades como la navegación, retroalimentación y evaluación. En resumen no posibilitan el cumplimiento de las Base Orientadora de la Acción, al no tener acciones de orientación, ejecución y control.

En una encuesta realizada a los profesores de MD, el 58% de los mismos considera que todavía falta la utilización correcta de la motivación, como elemento impulsor de la clase. Aunque el 100% considera que

no se cumplen totalmente los objetivos de la clase si no existe dicha motivación. El 89% nunca ha utilizado objetos de aprendizaje (OA) en sus clases y de ellos un 63% expresa que no los utilizan puesto que no poseen los OA adecuados y un 28% porque resulta difícil adaptarlos a las clases.

En base a lo anteriormente expuesto se puede concluir parcialmente que existe una gran necesidad de encontrar nuevas vías y elementos que ayuden al profesor a lograr una mayor motivación en el estudiante para con la asignatura de manera que se cumplan los objetivos de la clase. La creación de OA pudiera ser un elemento de apoyo para aumentar esta motivación. El bajo uso de estos recursos se debe en gran medida a que los pocos que existen son poco factibles por su pobre desempeño. Por tanto la creación de nuevos y renovadores OA, que permitan ser adaptados fácilmente a las clases, que motiven la creatividad y el pensamiento en el estudiante y que faciliten a los profesores nuevos mecanismos de evaluación sería un paso de avance en el perfeccionamiento del PEA de la MD.

Consideraciones finales del capítulo

A lo largo del capítulo se han expuesto los principales conceptos referentes al tema y se puede arribar a las siguientes consideraciones:

1. La MD posee una gran importancia en la formación de los futuros profesionales de las TIC puesto que proporciona las bases matemáticas para diferentes aspectos de la informática como son: estructuras de datos, algoritmia, bases de datos, teoría de autómatas, sistemas operativos, investigación operativa y ayuda al desarrollo de habilidades como el análisis mediante el uso de la abstracción o idealización de condiciones para investigar estructuras y procesos informáticos, desarrolla la capacidad para representar y razonar sobre sus propiedades y relaciones.
2. Después de realizar un análisis y caracterización de PEA de la MD en la UCI se analizó la estructura de dicha asignatura así como su contenido. La MD se divide en dos asignaturas MD1 y MD2, donde se estudian contenidos como: Lógica, Álgebra, Combinatoria, Grafos, Teoría de Números, Computabilidad, Diseño y Análisis de Algoritmos. Se detectaron problemas como: aplicación de métodos de evaluación efectiva, existe una inadecuada planificación de las clases donde el tiempo no se distribuye satisfactoriamente para lograr el total cumplimiento de los objetivos, uso deficiente de las TIC en el PEA de MD.
3. Se definieron los OA como recursos digitales que pueden ser reusados y cuyo principal objetivo es el de apoyar el proceso educacional. Los OA motivan el aprendizaje, posibilitar el acceso remoto a

la información, desarrollan el análisis y la reflexión. En la actualidad la mayoría de los OA presentan bajo grado de interactividad y no permiten experimentación lo que trae consigo que se dificulte la comprensión de información didáctica abstracta o compleja. Son accedidos mediante repositorios de OA, donde están catalogados, y quedan disponibles para compartir con el público.

4. Después de un diagnóstico del uso de los OA en el PEA de MD se evidenció el poco uso de estos recursos en la UCI debido al bajo grado de interactividad y la deficiente cantidad de estos recursos. En la universidad se cuenta con una pobre cantidad de OA, que se limitan a la presentación de contenido estático, impidiendo la experimentación e interacción con la aplicación.

Capítulo 2: Análisis y diseño de objetos de aprendizaje interactivos y experimentales (OA-IE) para apoyar el proceso de enseñanza de las Matemáticas Discretas

El presente capítulo tiene como objetivos presentar los resultados de los diferentes contenidos de la MD seleccionados; para el desarrollo de los OA-IE y su justificación; la selección de las metodologías, herramientas y lenguajes que serán utilizados y presentar el diseño de los diferentes OA-IE, de acuerdo a la metodología de desarrollo seleccionada y a las características que desde el punto de vista pedagógico deben tener.

2.1 Selección de Contenidos para desarrollar objetos de aprendizaje

Para realizar la selección de los contenidos para el desarrollo de OA-IE se realizó un proceso evaluación de los contenidos que se imparten en la MD, atendiendo a un conjunto de aspectos de acuerdo al criterio de diferentes profesores que imparten la asignatura. Este proceso de evaluación partió de una encuesta que se muestra en el anexo 1, los aspectos contenidos en la encuesta fueron consolidados y sus resultados quedan reflejados en la siguiente tabla.

Contenido	Indicador					
	Grado de complejidad	Posibilidad de representar gráficamente	Posibilidad de Experimentar	Grado de dificultad Asimilación	Grado de dificultad Impartición	Resultado en evaluaciones
Teoría de conjuntos	Medio	SI	SI	Bajo	Medio	Alto
Relaciones binarias	Medio	SI	SI	Medio	Medio	Alto
Lógica	Muy alto	NO	SI	Alto	Muy alto	Medio
Teoría de números	Alto	SI	SI	Muy alto	Muy alto	Bajo
Máquinas de Turing	Muy alto	SI	SI	Muy alto	Muy alto	Bajo
Teoría combinatoria	Alto	SI	SI	Alto	Alto	Medio
Teoría de Grafos	Alto	Si	SI	Medio	Medio	Medio
Relaciones de recurrencia	Alto	SI	SI	Muy alto	Muy alto	Medio

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

A partir de este proceso se definieron los temas siguientes:

- **Máquinas de Turing:** Teniendo en cuenta la alta complejidad de este contenido, así como la dificultad a la que se enfrentan los profesores para impartir el mismo. Además de lo anterior, los

resultados en diferentes exámenes, incluida la prueba final de MD1, pueden ser clasificados de pobres. Los estudiantes muestran serias dificultades en su asimilación. Esto se une a la poca diversidad de los ejercicios que se encuentran en el EVA y la inexperiencia didáctica de los profesores en el tema. Implementando un OA que permita interactuar con las Máquinas de Turing y resolver ejercicios se puede incidir favorablemente en la asimilación de este contenido.

- **Teoría de Conjuntos:** Es un contenido, que aunque no presenta alta complejidad ni grandes problemas de asimilación, si presenta dificultades a la hora de ser visualizado, lo que conlleva a que se complejice en gran medida su impartición. Los profesores en ocasiones no utilizan toda la potencia gráfica que puede tener este contenido para facilitar su asimilación. Los resultados asociados a él no son malos, aunque pudieran mejorarse si se lograra una mejor comprensión y ejercitación de los contenidos, presentando ejercicios más variados en forma y complejidad. Al poseer grandes posibilidades de visualización, permite además experimentar de forma dinámica y ver los resultados de aplicar las diferentes operaciones sobre conjuntos definidos de manera dinámica por los estudiantes y profesores. Es el primer contenido dentro de la Matemática Discreta lo que posibilita una interacción temprana con la visualización gráfica.
- **Teoría de Grafos:** Es un contenido complejo e importante para los especialistas de la informática. Presenta algunas dificultades a la hora de ser impartido por los profesores lo que dificulta en varios aspectos la asimilación por parte de los estudiantes. Los resultados de este contenido en exámenes no son los deseados por el colectivo de profesores. Se llega a la conclusión por unanimidad en la encuesta de que es un contenido que presenta grandes posibilidades de ser graficado, lo que permitiría en gran medida aumentar la motivación de los estudiantes. Permite experimentar con algunos parámetros y sobre todo con la ejecución de diferentes algoritmos sobre los grafos. Es un contenido de vital importancia para asignaturas de gran peso en la carrera de ICI como Programación 2. Es importante destacar que el proceso de dibujar un grafo es complejo.

De manera general la construcción de OA para los temas analizados anteriormente permitiría una mejoría en el proceso de asimilación de los estudiantes. Se puede aumentar la cantidad de ejercicios y ajustarlos a las características de cada estudiante. Todo lo anterior permitiría una mejoría en la auto-preparación de los estudiantes y de los profesores.

2.2 Selección de la metodología, herramientas y lenguajes a utilizar

Para la selección de la metodología, herramientas y lenguajes, se tuvieron en cuenta diversos criterios, que se analizaron en función de los requerimientos necesarios para la creación de OA-IE como producto de software para apoyar el PEA. Se tuvieron presentes requisitos que permitan la distribución de los OA-IE resultantes en el espacio dedicado a la MD en el EVA-UCI. La selección que se realiza en este epígrafe tiene como principales elementos los analizados en el epígrafe 1.2 del Capítulo I.

2.2.1 Metodologías de desarrollo

Una vez analizadas las metodologías se llegó a la conclusión de que las metodologías por separado presentan elementos que pudieran ser útiles para el proceso de desarrollo de OA-IE, pero por si solas ninguna cumple con los requerimientos necesarios para su desarrollo como una pieza de software. Por tal razón, en la presente investigación se considera el uso de una mezcla de las metodologías XP e ISD-MeLO, a continuación se explican los principales elementos que llevaron a tomar esa decisión.

Al ser la creación de los OA-IE relacionados con los contenidos seleccionados pequeños proyectos de desarrollo de software auto contenidos, que deben tenerse en un corto plazo es factible la utilización de XP como metodología para guiar el proceso de desarrollo, es fácil de implementar, su documentación es ligera, su curva de aprendizaje es suave, todo ello la hace eficiente y flexible en el caso de la presente investigación. Por otro parte la metodología ISD-MeLO es usada específicamente para la concepción de los OA-IE. Esta metodología puede ser muy útil para guiar el proceso de creación de los OA-IE desde el punto de vista pedagógico lo cual le proporciona flexibilidad para aplicarse en una amplia diversidad de situaciones. En la investigación se dispondrá de docentes que conciban las características pedagógicas del OA, por lo que desde el punto de vista informático XP tendrá un peso mayor, sin demeritar el uso de ISD-MeLO.

2.2.2 Lenguaje

Luego del análisis de los lenguajes de programación realizado en el capítulo anterior, se pudo contrastar las distintas características y facilidades de lenguajes como Java, C++ y C#. Como resultado de este análisis se decidió el uso de Java para la creación de los OA-IE. La promesa inicial de Gosling, el creador de JAVA, se resume en Write Once, Run Anywhere (escríbelo una vez, ejecútalo en cualquier lugar), proporcionando un lenguaje independiente de la plataforma y un entorno de ejecución ligero y gratuito para las plataformas más populares de forma que los binarios (bytecode) de las aplicaciones Java pudiesen ejecutarse en cualquier plataforma. Java es un lenguaje Open Source y Orientado a Objetos que permite

la programación multi-hilo y está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows NT y Mac, sobre arquitecturas distintas y con sistemas operativos diversos. Algunas de las ventajas que se pueden lograr con Java a diferencia de los otros lenguajes analizados es la creación de applets que pueden ser incrustados fácilmente en documentos HTML.

2.2.3 Herramientas para el desarrollo

En el capítulo anterior se analizó un conjunto de herramientas que son utilizadas con el fin de crear OA-IE de diferentes tipos y formatos. De manera general estas herramientas permiten desarrollar multimedia, aplicaciones flash, libros animados, presentaciones animadas etc. Cuando se quiere desarrollar un OA-IE como un producto de software, estas herramientas pudieran presentar grandes limitantes en cuanto al diseño e implementación. Por el motivo anterior se define el uso de un IDE para la creación de OA-IE, además se tuvo en cuenta la necesidad de crear OA-IE que puedan ser puestos a disposición de los estudiantes a través del EVA-UCI, incrustándolo en él, y que puedan ser ejecutados en cualquier plataforma. Por esta razón para esta investigación se decidió usar el Netbeans como IDE al proporcionar herramientas cómodas y fáciles de utilizar, permitir el desarrollo de aplicaciones multiplataforma, y especialmente Java applets que pueden ser incrustados en una página web. Netbeans, como ya se analizó en el capítulo anterior es un entorno de desarrollo integrado Open Source y de distribución gratuita y permite además la creación de aplicaciones en una gran cantidad de lenguajes como PHP, Java, C/C++ y Groovy entre otros.

2.2.4 Herramientas Case

La realización de un nuevo software requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software. Se puede definir a las herramientas CASE (Computer Aided Software Engineering) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. Como es sabido, los estados en el ciclo de vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación (Murillo Alfaro, 1999).

Suite Visual Paradigm

Suite de productos para desarrollar software de manera eficiente, rápida y de forma colaborativa. Permite realizar diagramas de procesos de negocios, modelado UML, modelos de casos de usos, modelos de actividad, de interacción, de bases de datos, de entidad-relación. Además posee integración para varios IDE, puede realizar ingeniería de código y también generar documentación, entre otras cosas; todo bajo un modelo colaborativo. Visual Paradigm Suite soporta todas las necesidades de diseño y modelado a lo largo del ciclo de vida de desarrollo de software, es una herramienta que ayuda a construir aplicaciones de calidad, de manera más rápida, óptima y barata. Uno de los productos de Visual Paradigm Suite es Visual Paradigm for UML (VP-UML) que permite generar diagramas de varios tipos, incluyendo UML, modelado de procesos de negocios, mapas mentales, Object-Relational Mapping etc. (Grupo Satellite s.a de c.v, 2011).

Considerando todas las facilidades brindadas por el Visual Paradigm, que es un producto de calidad, muy fácil de instalar y actualizar, que está disponible en varias ediciones entre las que se encuentran Enterprise, Professional, Community, Standard, Modeler y Personal, ofreciéndoles a los usuarios la posibilidad de escoger según sus necesidades; y que además, es una herramienta de software libre, y que se cuenta con una licencia en la universidad, permite concluir, que constituye la mejor opción para realizar el modelado de diagramas de este trabajo según la metodología propuesta.

2.2.5 Evaluación de la calidad

La evaluación de la calidad es un proceso necesario e importante, sobre todo cuando se trata de OA-IE que serán utilizados como apoyo del 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 ver los OA en su dualidad como productos pedagógicos y de software. Después del análisis de los instrumentos e indicadores utilizados para la evaluación de la calidad analizados, se evidencia la necesidad de una evaluación desde la perspectiva pedagógica, curricular, técnica y funcional. Los instrumentos analizados se pueden considerar bastante completos, incluyen los aspectos formativos, tecnológicos y de diseño y presentación. Pero son muy generales, estos deberían incluir otros criterios que permitan completar la evaluación de los OA-IE, como la vigencia de los contenidos, compatibilidad, equilibrio en la distribución de los contenidos, nivel de organización de la estructura de archivos y directorios, asociación de archivos con la estructura didáctica, entre otros. Se decide adoptar como estrategia de evaluación de la calidad utilizar la Guía propuesta por (Toll Palma, 2011) al contener los elementos que permiten evaluar la calidad del

OA-IE, sin embargo se aplicarán las técnicas de prueba de software para probar las funcionalidades del software de acuerdo a la metodología de desarrollo seleccionada, de manera que se garantice su correcto funcionamiento.

2.3 Diseño de los objetos de aprendizaje

Luego de haber sido seleccionados los elementos necesarios para la realización de los OA-IE, es preciso realizar el diseño de cada uno. A la hora de conformar los OA-IE existen aspectos a tener en cuenta para lograr la calidad requerida de tal forma que se cumplan los objetivos que se persiguen con su realización. Al considerar los OA-IE como recursos amplios, que además de abarcar contenidos y contener información, considera los elementos del proceso de asimilación que son necesarios para el óptimo desarrollo del PEA, se definen tres unidades invariantes: **la unidad de información, la unidad didáctica y la unidad de evaluación y retroalimentación.**

La unidad de información: la conforman los elementos teóricos (definiciones, conceptos, ejemplos resueltos, etc.) y es la fuente desde donde el usuario del OA-IE puede informarse y tomar sus apuntes sobre el contenido tratado. Desde el punto de vista de diseño es el resultado la elección del contenido, la creación de las formas de presentación, apoyándose en las características de los usuarios o destinatarios y la ubicación de éste en el espacio. El OA-IE viene a constituir la interface entre el sujeto y el tipo de acción por lo tanto, debe cumplir con los criterios de accesibilidad para el usuario. Dicha unidad está conformada por contenidos multimedia individuales (texto, imágenes y audio) en la que el estudiante tiene la posibilidad de acceder a los conocimientos relacionados con el contenido sobre el cual está desarrollado el OA-IE. 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 los OA-IE, este elemento potencia la reutilización y disminuye el tiempo de desarrollo.

Por otra parte el diseño de los OA-IE viene a fragmentar el proceso de aprendizaje en una serie de actividades, lo que permite tener un gran número de combinaciones que pueden generarse a partir de todos los OA-IE que constituyen un curso. De esta forma se facilita la flexibilización de las estrategias de aprendizaje a través de **la unidad didáctica.** Esta unidad está orientada a experimentar y reflexionar acerca de los elementos que se adquirió en la teoría y, por tanto, estará orientada a un estilo de aprendizaje concreto. Abarca cada uno de los elementos que permiten la interacción y experimentación del estudiante con el contenido y está estrechamente relacionada con las otras unidades.

La **unidad de evaluación y retroalimentación** permite utilizar diferentes estrategias para comprobar lo que se ha aprendido. En esta unidad se utilizan diferentes criterios para la evaluación y contiene actividades de enseñanza- aprendizaje, es muy importante porque permite comprobar los conocimientos adquiridos.

El contenido específico de las unidades didácticas varía para cada uno de los OA-IE a desarrollar, de acuerdo a las características de cada uno de ellos, a pesar de ello existen elementos desde el punto de vista del diseño que son comunes. Luego del análisis realizado anteriormente y con el objetivo de simplificar el proceso de desarrollo de los OA-IE se decidió realizar un **prototipo genérico** que contenga los embriones de las diferentes unidades y que pueda ser reusado para el diseño de cada uno de los OA-IE propuestos y de otros que se puedan concebir en el futuro.

2.3.1 Prototipo del OA Experimental e Interactivo.

El prototipo está conformado por un applet que servirá como plantilla base para el desarrollo del resto de los OA-IE, ella tendrá implementada la forma en que se muestran diferentes recursos como son imágenes, texto y audio, donde solo será necesario agregar la información referente a los contenidos específicos que abordan los OA-IE mediante pequeñas modificaciones del código en algunos casos. La plantilla propuesta contiene en su diseño las tres unidades fundamentales que conforman los OA-IE como se muestra en la figura 2.

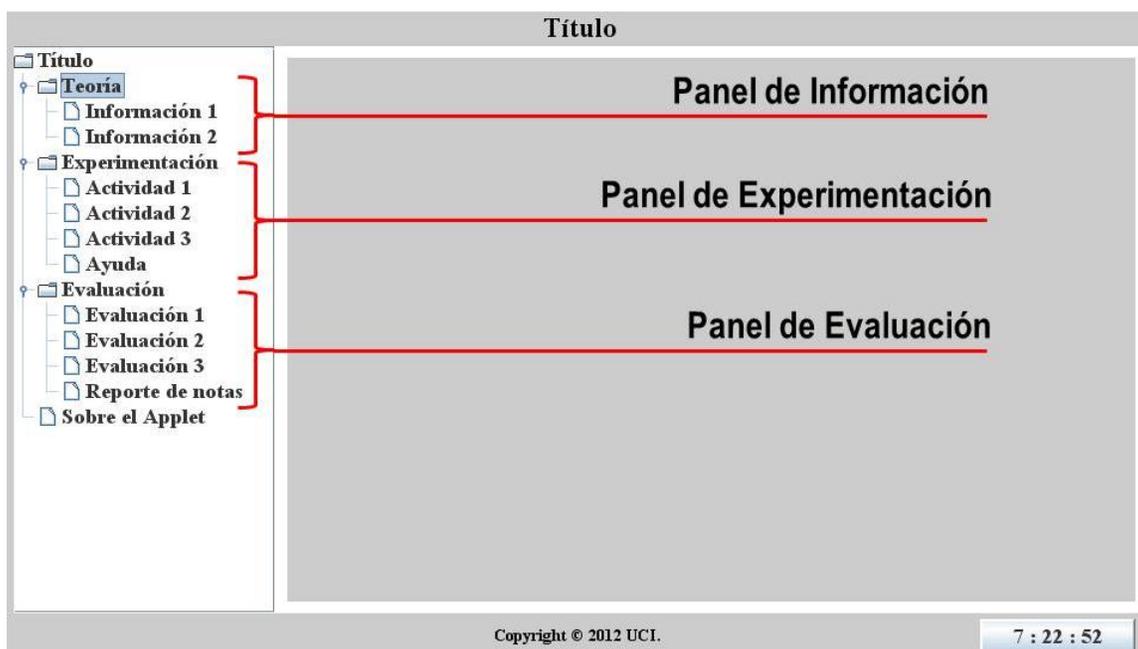


Figura 2. Prototipo del OA.

El diseño básico de la plantilla consiste en un applet que en su parte izquierda contiene un menú que muestra la unidad de información, la unidad de experimentación y simulaciones y la unidad de evaluación y retroalimentación, se cuenta además con una opción auxiliar donde se puede visualizar información referente a la aplicación. En la parte derecha se muestra el área de trabajo que está conformada por diferentes paneles anidados que muestra diferentes interfaces visuales y que están asociados a cada una de las opciones del menú mencionado anteriormente.

Unidad de información (UI): el menú destinado a la UI (panel de información) controla lo referente a la información que se muestra acerca de los contenidos que trata el OA-IE. Para cada una de las opciones en este panel se carga en el área de trabajo a manera de diapositivas la información definida para esa opción. Se cuenta con una barra de controles que permitirán al usuario controlar la presentación de dicha información. Dicha barra cuenta además con controles que permite la reproducción de audio en caso de que algún contenido tenga asociados recursos de este tipo. Para modificar esta sección es necesario modificar el código para cada una de las opciones del menú, indicando las direcciones de la galería de imágenes de acuerdo al contenido asociado a cada opción así como los archivos de audio en caso de ser usados.

Unidad didáctica (UD): mediante el menú destinado a la UD (panel de experimentación y simulaciones) se puede acceder a las diferentes simulaciones y actividades propuestas en el OA-IE. En esta sección se permite la reflexión, interacción y experimentación con los contenidos mediante la práctica. Para cada uno de los elementos del menú se definen diferentes tipos de actividades y simulaciones que complementan los conocimientos teóricos adquiridos. Para modificar esta sección se define la actividad deseada en el menú, el cual cuenta con un panel asociado donde se podrá implementar la actividad correspondiente.

Unidad de evaluación y retroalimentación (UER): En el menú destinado a ella (panel evaluación y retroalimentación) se realiza el control de los conocimientos adquiridos mediante diferentes criterios de evaluación. Para cada una de las opciones en este menú se pueden encontrar diferentes actividades de carácter evaluativo. Para realizar modificaciones sobre la UER será necesario especificar en el menú el ejercicio que se desea diseñar el cual tiene asociado un panel donde se podrá implementar dicha tarea. Aquí se encontrara una opción que permite generar un resumen de la actividad realizada por el estudiante. Finalmente se cuenta con un menú auxiliar donde se podrá encontrar información referente a la aplicación, dicho menú tendrá asociado un panel donde se podrá adicionar información como: autor, versión de la

máquina virtual de java a utilizar, versión de la aplicación, datos de contacto etc.

Cada uno de los OA-IE fue desarrollado siguiendo la estructura explicada, a continuación se realiza una explicación del diseño cada uno de los OA-IE.

2.3.2 OA Teoría de Conjuntos

En la figura 3 se muestra un ejemplo de la UI en el OA-IE de Teoría de Conjuntos (OA-TC), donde se pueden observar los controles que permiten manipular el contenido que se muestra, el funcionamiento de esta unidad fue explicado en el epígrafe anterior.

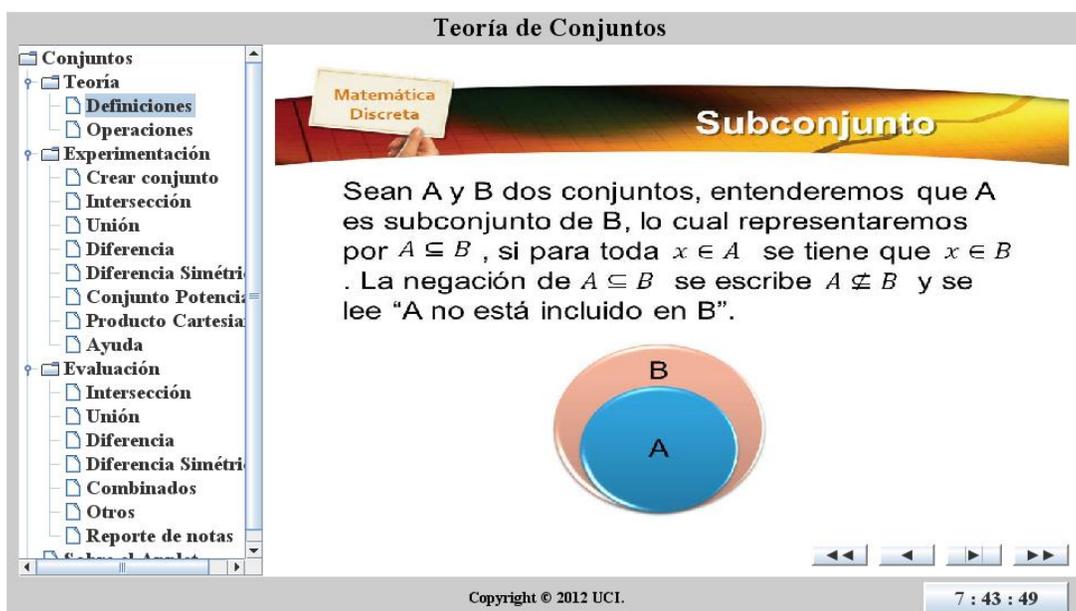


Figura 3. Unidad Informativa.

Uno de los elementos que se pueden encontrar dentro de la UD es una interfaz que permite al usuario la gestión de los conjuntos como se muestra en la figura 4. En esta interfaz se brinda la posibilidad de creación de nuevos conjuntos especificando el nombre, el cual debe ser una letra mayúscula. Los conjuntos creados se añaden vacíos a la lista de conjuntos y a un combo box que permite su selección. Una vez creado los conjuntos es posible seleccionar uno de ellos o bien en la lista de conjuntos o en una lista desplegable donde se van adicionando en el momento de ser creados. Cuando un conjunto es seleccionado se brinda la posibilidad de añadirle elementos que pertenezcan al universo con que se trabaja, en este caso definido por todas las letras minúsculas del alfabeto español, el OA-IE cuenta con una pequeña ayuda donde se explican estos aspectos. La opción “Mostrar conjunto” permite la visualización del conjunto seleccionado mediante los Diagramas de Venn (DV).

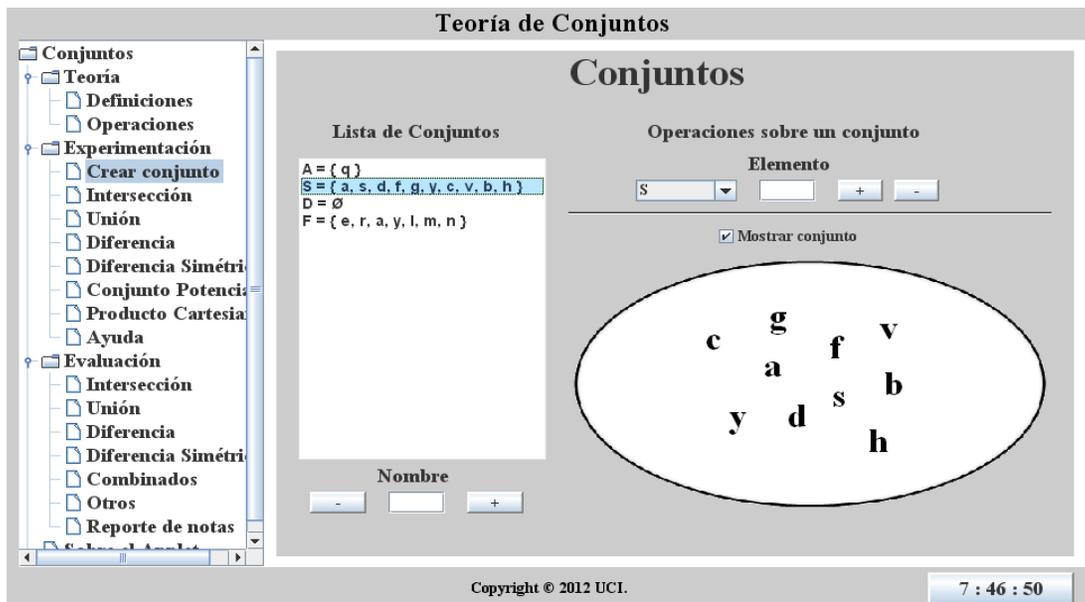


Figura 4. Creación de Conjuntos.

En la sección de experimentación del OA-IE se permite realizar distintas operaciones sobre los conjuntos como son: Unión, Intersección, Diferencia, Diferencia Simétrica y calcular el Conjunto Potencia y el Producto Cartesiano entre dos conjuntos. Cada una de estas operaciones se encuentra disponible en el menú de experimentación, a las cuales se puede acceder después de haber creado al menos un conjunto. De manera general en las operaciones, se le brinda al usuario la posibilidad de elegir los conjuntos sobre los cuales desea realizar la operación mediante listas desplegables que contienen los nombres de los conjuntos creados. Una vez seleccionado el conjunto se mostrará mediante DV en el cual estará identificado con un color específico (rojo o azul). Una vez seleccionados los conjuntos se brinda la posibilidad de que el usuario resuelva el ejercicio o de visualizar automáticamente la respuesta que ofrece la aplicación, en los dos casos se procede a mostrar el resultado mediante DV.

La **intersección** de dos conjuntos es una operación que resulta en otro conjunto que contiene los elementos comunes a los conjuntos seleccionados para la operación. Por ejemplo, dado el conjunto $F = \{j, g, f, a, h, d, s, t, k\}$ y el conjunto $H = \{j, d, f, s, b, n, c, k, l\}$, su intersección es un conjunto $G = \{j, f, d, s, k\}$. La intersección entre conjuntos se denota por el símbolo \cap por lo que $G = F \cap H$. El resultado de la operación es mostrado mediante DV, donde se muestran los elementos en común en color negro mientras que los elementos diferentes se muestran en conjuntos diferentes donde los elementos son identificados con el color de su conjunto de partida como se muestra en la figura 5.

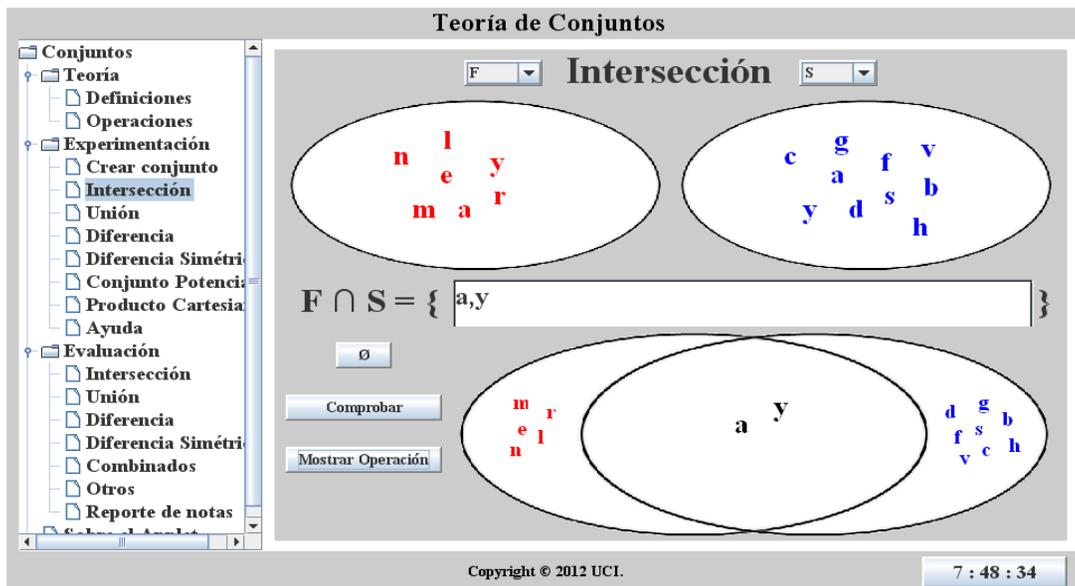


Figura 5. Intersección

La **unión** de dos conjuntos es una operación que resulta en otro conjunto cuyos elementos son los elementos de los conjuntos seleccionados para la operación. Por ejemplo, dado el conjunto $F = \{j, g, f, a, h, d, s, t, k\}$ y el conjunto $H = \{j, d, f, s, b, n, c, k, l\}$, su unión es un conjunto $G = \{j, g, f, a, h, d, s, t, k, b, n, c, l\}$. La unión entre conjuntos se denota por el símbolo \cup por lo que $G = F \cup H$. El resultado de la operación es mostrado mediante DV en un único conjunto, en el que los elementos en común son mostrados en color negro mientras que los elementos diferentes son identificados con el color de su conjunto de partida como se muestra en la figura 6.

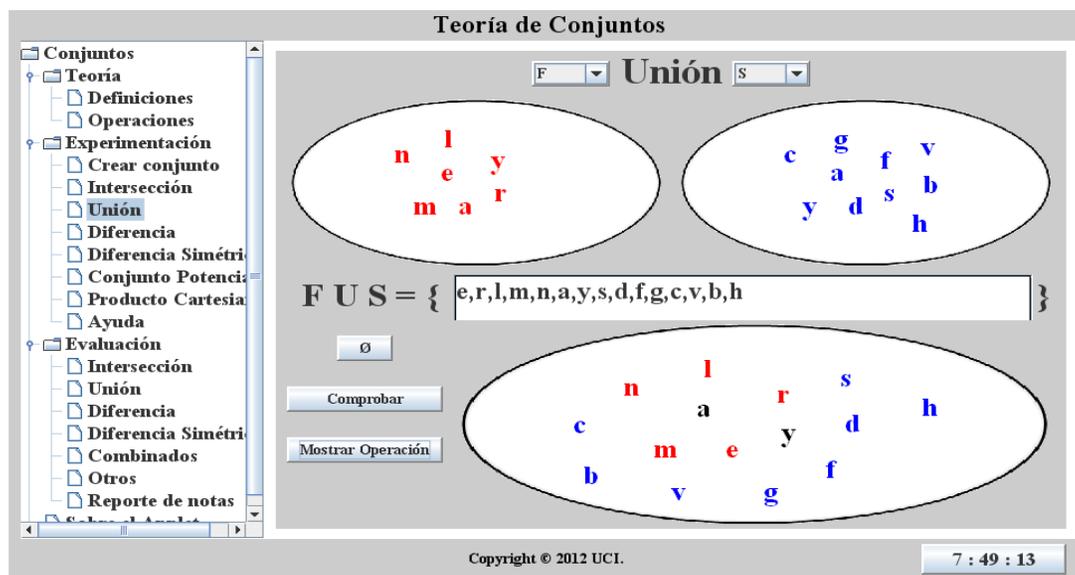


Figura 6. Unión

La **diferencia** entre dos conjuntos es una operación que resulta en otro conjunto, cuyos elementos son todos aquellos que pertenezcan al primero de los conjuntos iniciales que no estén en el segundo. Por ejemplo, dado el conjunto $F = \{j, g, f, a, h, d, s, t, k\}$ y el conjunto $H = \{j, d, f, s, b, n, c, k, l\}$, la diferencia de F con respecto a H es un conjunto $G = \{g, a, h, t\}$. La diferencia entre conjuntos se denota por el símbolo \setminus , por lo que: $F \setminus H = G$. El resultado de la operación es mostrado mediante DV en un único conjunto, en el que los elementos son identificados con el color de su conjunto de partida como se muestra en la figura 7.

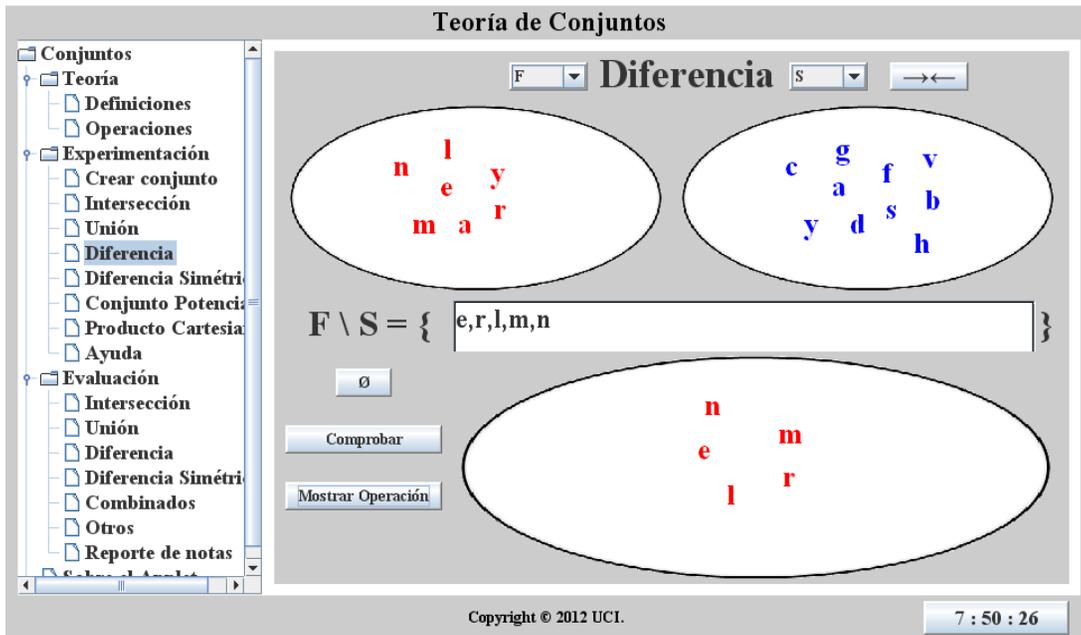


Figura 7. Diferencia

La **diferencia simétrica** de dos conjuntos es una operación que resulta en otro conjunto cuyos elementos son aquellos que pertenecen a alguno de los conjuntos iniciales, sin pertenecer a ambos a la vez. Por ejemplo, dado el conjunto $F = \{j, g, f, a, h, d, s, t, k\}$ y el conjunto $A = \{z, f, b, c\}$, la diferencia simétrica de F y A es un conjunto $G = \{j, g, a, h, d, s, t, k, z, b, c\}$. La diferencia simétrica entre conjuntos se denota por el símbolo Δ , por lo que: $F \Delta A = G$. El resultado de la operación es mostrado mediante DV, donde los elementos son mostrados en un único conjunto identificados con el color de su conjunto de partida como se muestra en la figura 8.

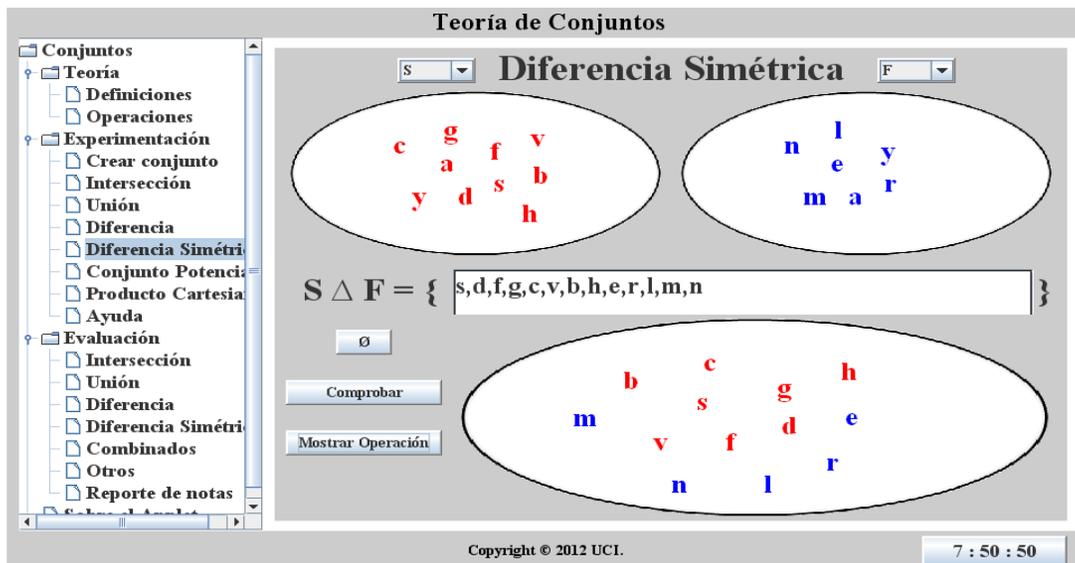


Figura 8. Diferencia Simétrica

El **producto cartesiano** de dos conjuntos es una operación que resulta en otro conjunto cuyos elementos son todos los pares ordenados que pueden formarse tomando el primer elemento del par del primer conjunto, y el segundo elemento del segundo conjunto. Por ejemplo de ser seleccionados los conjuntos $F = \{1, 2, 3, 4\}$ y $B = \{a, b\}$, su producto cartesiano es: $A \times B = \{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b), (4, a), (4, b)\}$. El resultado de la operación es mostrado en un cuadro de texto mediante su notación general. Se cuenta además con un listado donde se guardan las operaciones realizadas como se muestra en la figura 9.

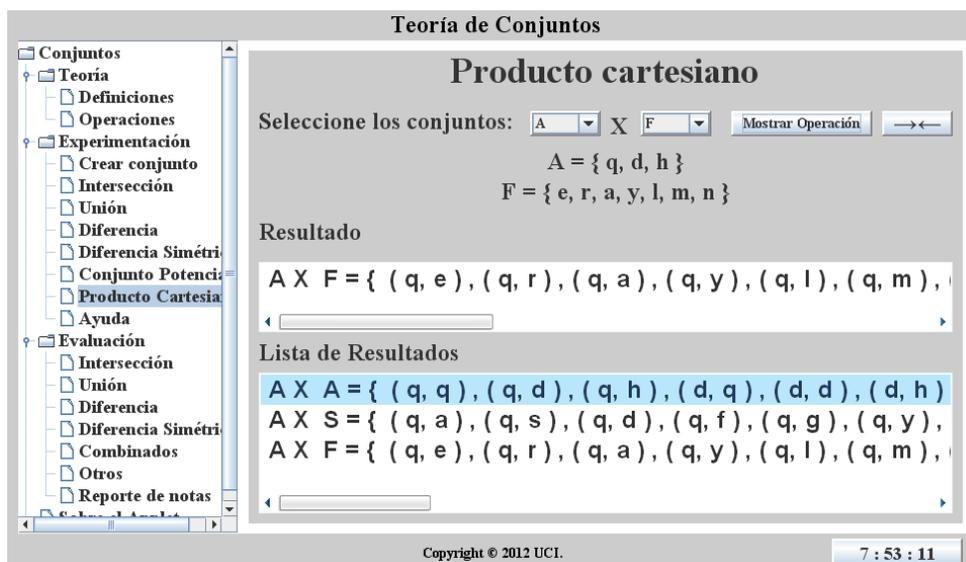


Figura 9. Producto Cartesiano

El **conjunto potencia** es el conjunto formado por todos los subconjuntos posibles del conjunto seleccionado para realizar la operación. Por ejemplo, si es seleccionado el conjunto $A = \{s, d, f\}$ entonces el conjunto potencia de A se denota como $P(A)$ de manera que $P(A) = \{\emptyset, \{s\}, \{d\}, \{f\}, \{d, f\}, \{s, d\}, \{s, f\}, \{s, d, f\}\}$. El resultado de la operación es mostrado en un cuadro de texto mediante su notación general. Se cuenta además con un listado donde se guardan las operaciones realizadas como se muestra en la figura 10.

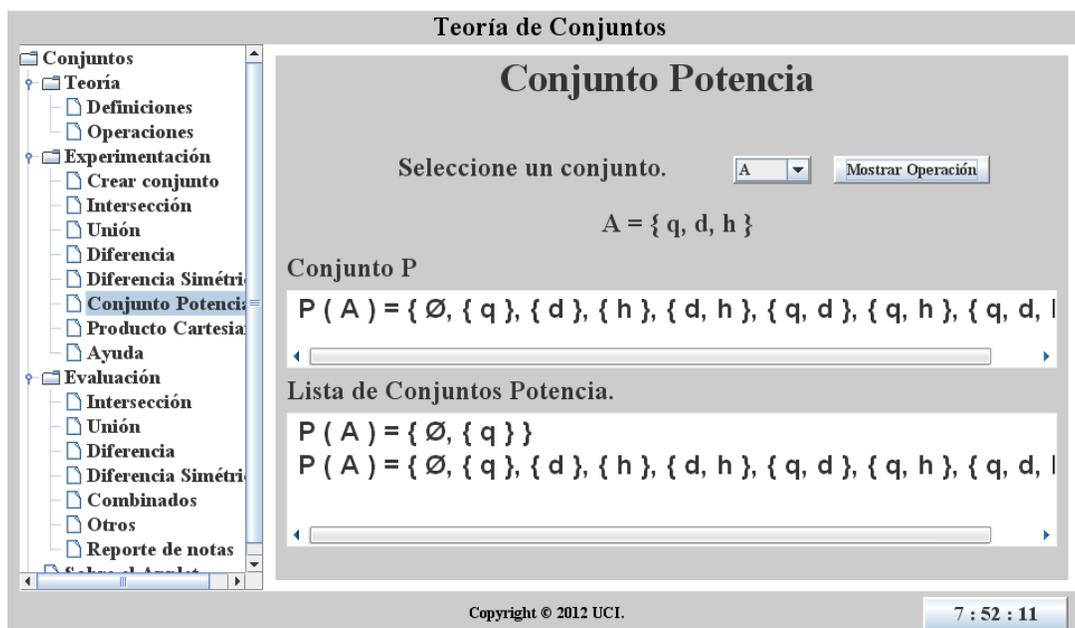


Figura 10. Conjunto Potencia.

En la UER se brindan una serie de ejercicios comprobatorios acerca de los contenidos con los que el estudiante interactúa en el OA (figura 11). Una de las ventajas que brinda esta sección es la variabilidad de ejercicios. Las operaciones presentan variados niveles de complejidad y se realizan sobre conjuntos que son generados aleatoriamente por la aplicación. Cada su usuario que haga uso del OA se enfrentará a ejercicios con datos completamente diferentes, y el OA-IE será capaz de evaluar su respuesta.

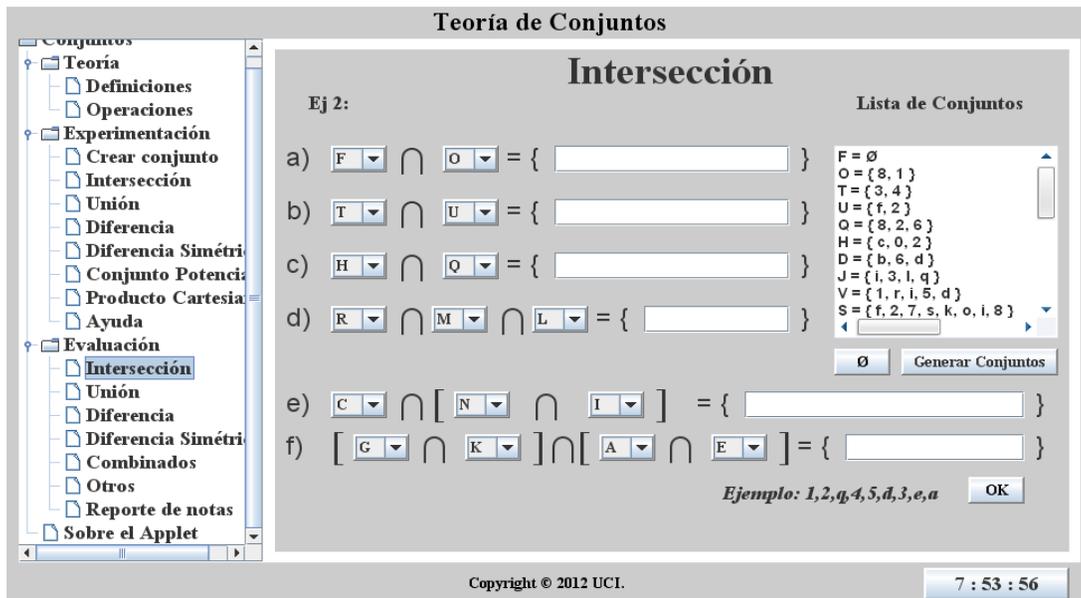


Figura 11. Ejercicio evaluativo de Intersección.

En el anexo 2 se muestran las figuras relativas al resto de los ejercicios evaluativos vinculados con las restantes operaciones definidas para los conjuntos.

2.3.3 OA Máquina de Turing

El OA Máquina de Turing (OA-MT) cuenta con una UI similar a las ya descritas. Al igual que el OA-IE anterior cuenta con su UD, en ella hay definidos varios ejemplos que permiten experimentar con diferentes máquinas prediseñadas (figura 12), esto se hizo de esta manera por la dificultad que conlleva generar diferentes máquinas de Turing de manera aleatoria garantizando su consistencia.

En la interfaz de ejemplos prácticos se da la posibilidad de seleccionar el ejemplo con el que se desea trabajar mediante una lista desplegable identificada como "Ejemplos". Una vez seleccionado el ejemplo se mostrará la descripción del mismo en un cuadro de texto. Se cuenta con una lista donde se muestran a decisión del usuario las funciones de transición que representan una posible solución al ejemplo en cuestión. Esta lista puede ser mostrada mediante la selección de la opción "Mostrar función". Aparece además una lista donde se muestran los estados finales definidos para la respuesta al ejemplo de acuerdo a las funciones definidas.

En esta interfaz se cuenta con un campo de texto donde se pueden experimentar con varias cadenas sobre el ejemplo seleccionado apareciendo en una lista cada uno de los estados por los que pasa el cabezal para cada uno de los símbolos analizados. De las cadenas analizadas pueden ser obtenidos los

pasos que permiten decir si pertenece o no al lenguaje que reconoce la MT, mediante un grupo de controles que permiten ir moviendo el cabezal paso a paso por cada uno de los símbolos contenidos en la cinta.

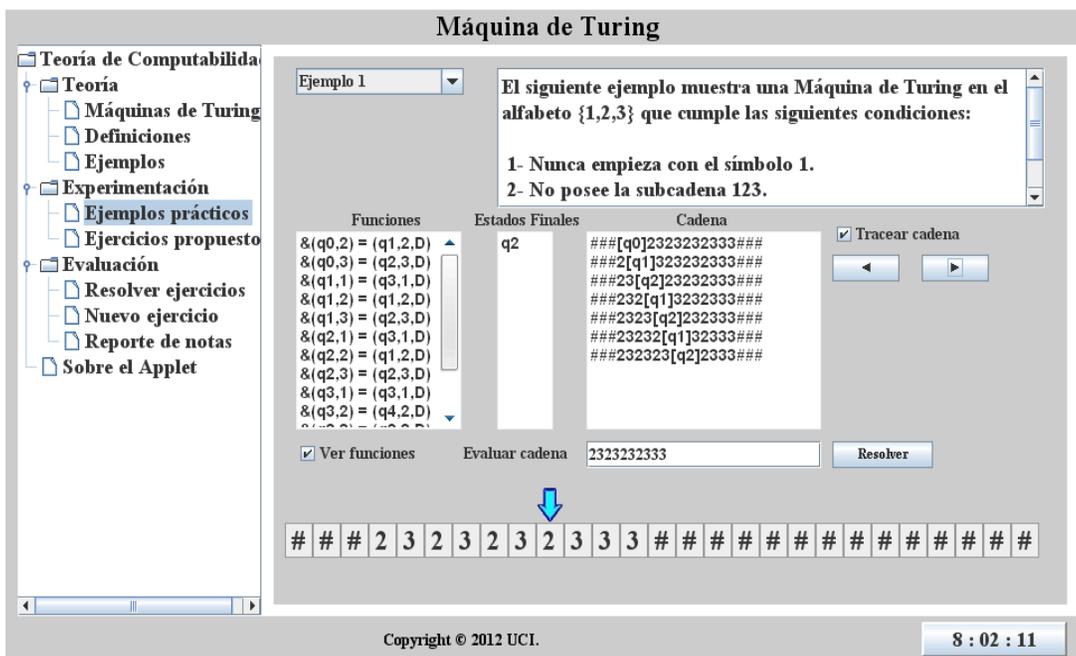


Figura 12. Ejemplo de Máquina de Turing.

En la figura 13 se muestra una interfaz donde se permite la resolución de ejercicios de forma experimental, la forma de resolver dichos ejercicios es muy similar a la resolución de los ejemplos explicados anteriormente. La diferencia es que en este caso las funciones y los estados no están definidos, es el usuario quien debe entrar estos valores, el usuario debe definir la función de transición de la Máquina de Turing. Una vez entrados los valores de las funciones de transición así como el estado inicial y los estados finales, se puede contar con funcionalidades similares a las de la sección de ejemplos, y se cuenta además con la opción “Comprobar Máquina”, la cual determina si la solución del usuario es una solución correcta o no.

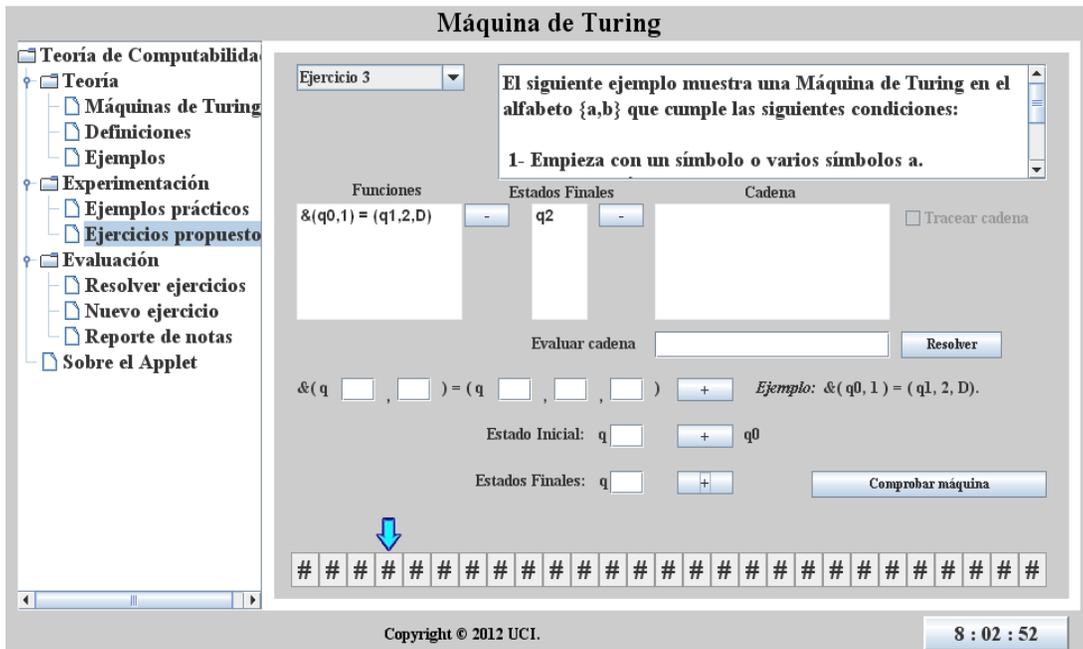


Figura 13. Interfaz de los Ejercicios prácticos.

Una de las mayores ventajas que brinda el OA-IE, es la posibilidad de crear nuevos ejercicios como se muestra en la figura 14. El profesor tiene la posibilidad de definir el alfabeto de entrada así como una serie de condiciones que la Máquina debe cumplir. Luego de definidos el alfabeto y las condiciones, el ejercicio es añadido a la lista de ejercicios evaluativos donde es posible realizar su resolución.

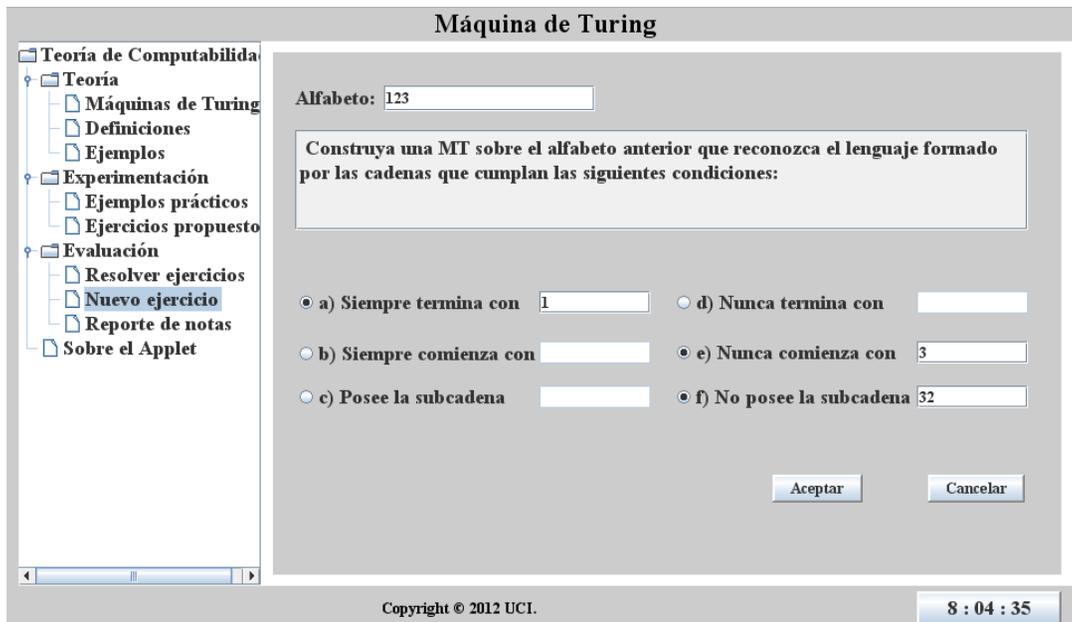


Figura 14. Nuevo Ejercicio

La UER del OA – MT es muy similar a su UD donde se permite la resolución de ejercicios propuestos, con la diferencia de que en este caso los ejercicios son diferentes y de carácter evaluativo, es una unidad que complementa el proceso de aprendizaje ya que permite comprobar los conocimientos adquiridos luego de que el estudiante interactúa con elementos teóricos y prácticos en el OA.

2.3.4 OA Teoría de Grafos.

Al igual que los OA descritos anteriormente el OA–TG cuenta con las tres unidades mismas que conforman los OA propuestos. En el caso de la UD se cuenta con tres marcos de trabajo fundamentales orientados a la gestión de los grafos con los que se experimenta en el OA.

Como se muestra en la figura 15 se cuenta con una interfaz donde se permite la creación de nuevos grafos, especificando su nombre con una cadena de no más de tres caracteres y con letra inicial mayúscula. Los grafos creados se añaden a una lista de grafos donde pueden ser visualizados y seleccionados. Una vez creados los grafos es posible seleccionarlos en la lista de grafos y añadirle vértices especificando su identificador, el cual debe ser una letra minúscula. Los vértices creados se adicionan a una lista de vértices donde pueden ser visualizados y seleccionados. Luego de creado un grafo y de haberle adicionados vértices al mismo se brinda la posibilidad de añadirle aristas. Luego de seleccionar un grafo en la lista de grafos sus vértices serán mostrados en dos listas desplegables donde se podrán seleccionar los dos vértices entre los cuales se desea crear una arista. Las aristas creadas se adicionan a una lista de aristas donde pueden ser visualizadas y seleccionadas.

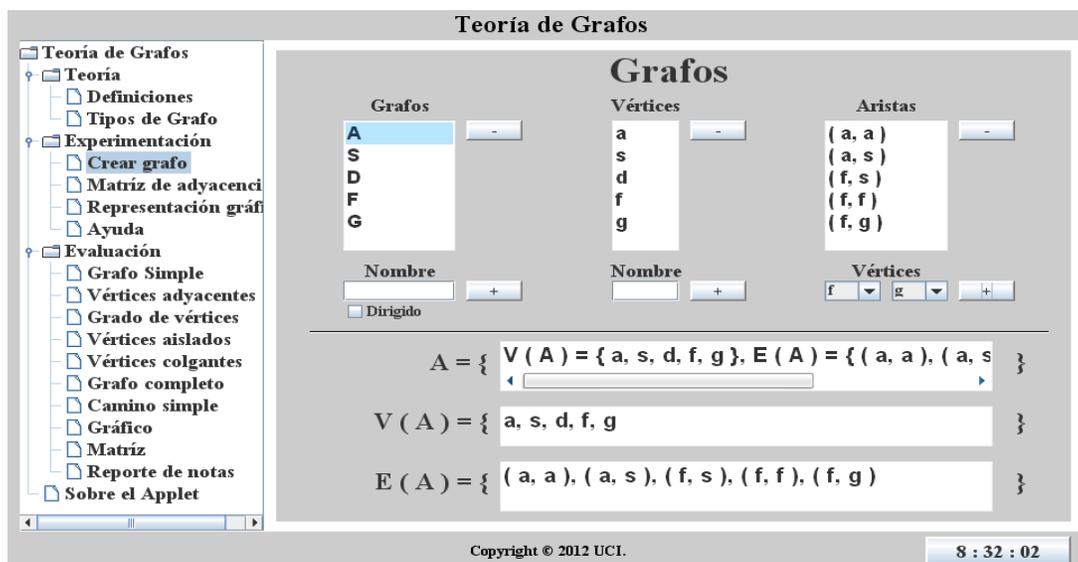


Figura 15. Crear Grafo

En la figura 16 se muestra otra de las interfaces con las que se cuenta en la UD donde se permite la visualización de los grafos mediante matrices de adyacencia. Se cuenta con una lista desplegable donde se puede seleccionar el grafo que se desee visualizar.



Figura 16. Matriz de adyacencia

La representación gráfica brinda un alto grado de comprensión de la Teoría de Grafos y cierta facilidad para la realización de operaciones sobre los mismos, por tal motivo es una de las características que se potencian en el OA-TG, es por ello que en la UD se cuenta con un marco de trabajo donde se permite realizar la representación gráfica de los grafos (figura 17). El grafo que se desee graficar puede ser seleccionado en una lista desplegable, una vez seleccionado se mostrará automáticamente el grafo brindando la posibilidad al usuario de redistribuir los vértices del grafo en el área de dibujo a su antojo.

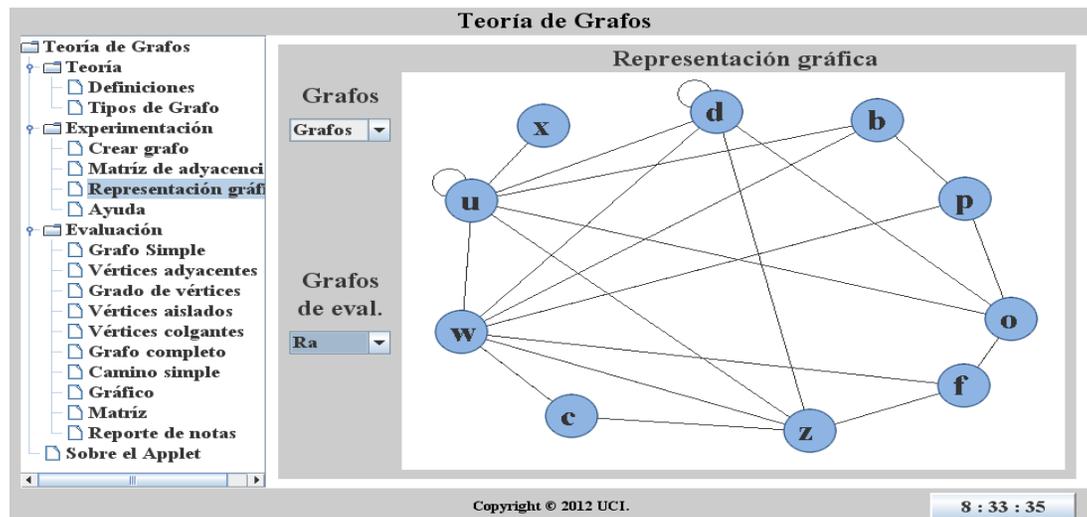


Figura 17. Representación grafica

En la UER del AT-TG al igual que en los OA descritos anteriormente se brindan una variedad de ejercicios de carácter evaluativo sobre algunos de los contenidos con los que se interactúa en el OA (figura 18). La variabilidad de ejercicios es una de las principales ventajas con las que cuenta el OA ya que los grafos son generados aleatoriamente por la aplicación, lo que supone que cada usuario que haga uso del OA se enfrentará a ejercicios con datos completamente diferentes, y el OA-IE será capaz de evaluar la respuesta y brindar una nota al usuario de acuerdo a su desempeño.

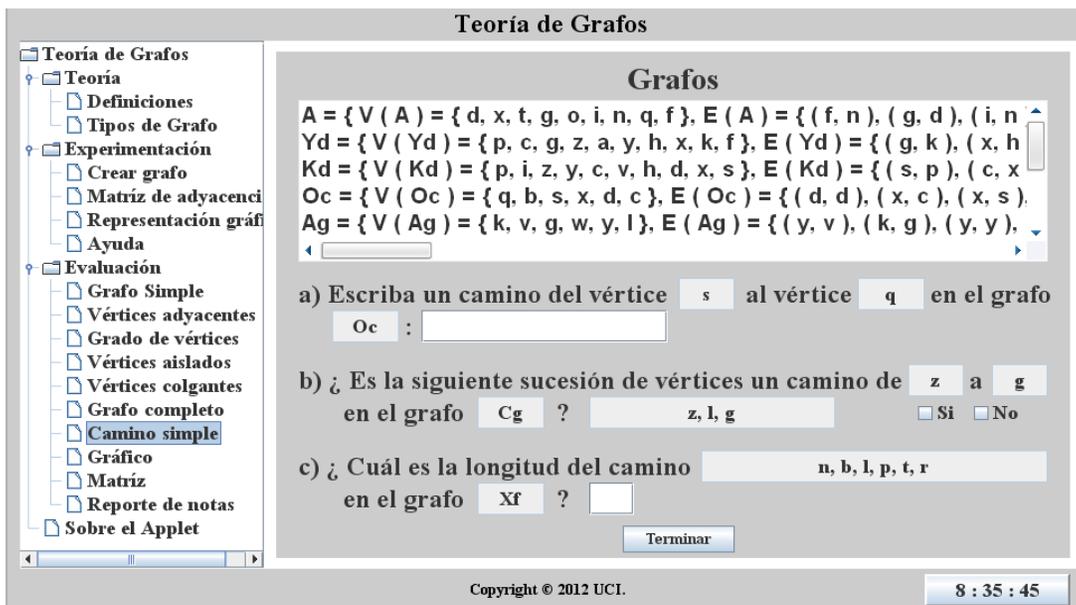


Figura 18. Ejercicio evaluativo de Camino Simple

En el anexo 3 se muestran las figuras relativas al resto de los ejercicios evaluativos vinculados con el resto de las operaciones definidas para los grafos.

Conclusiones parciales

A lo largo del capítulo se analizaron elementos referentes al contenido y diseño de los OA-IE por lo que se puede llegar a las siguientes conclusiones:

1. Después de un análisis de diferentes aspectos de los contenidos impartidos en el programa de la asignatura MD se definieron para el desarrollo de OA-IE los contenidos de Teoría de Conjuntos, Máquinas de Turing y Teoría de grafos.
2. Se definieron las herramientas y metodología para el desarrollo de los OA-IE, quedando definida XP e ISD-MeLO como metodologías de desarrollo, ambas se complementan para el desarrollo de OA-IE como producto de software, XP guiando el proceso como software e ISD-MeLO desde el

punto de vista del PEA. Como lenguaje se seleccionó Java, y se opta por el uso del Netbeans como entorno de desarrollo integrado a partir de las ventajas que brindan para el desarrollo de aplicaciones fiables y para la creación de applets.

3. Con el objetivo de simplificar el proceso de desarrollo y potenciar la reutilización, se diseñó un prototipo genérico que permitió implementar las funcionalidades y unidades que son comunes para todos los objetos de aprendizaje.
4. Se realizó el diseño del Objeto de aprendizaje de Teoría de Conjuntos potenciando las posibilidades de representación gráfica que posee, en su unidad didáctica brinda la posibilidad de gestionar conjuntos viéndolos de manera gráfica, así como la realización de distintas operaciones de manera demostrativa. Se diseñó además una forma de evaluación dinámica que brinda al profesor una gran variabilidad en los ejercicios permitiendo una evaluación diferenciada cada vez que se use el objeto de aprendizaje.
5. En el diseño del Objeto de aprendizaje de Máquinas de Turing se logró un alto grado de experimentación e interacción, brinda la posibilidad de realizar simulaciones sobre Máquinas de Turing previamente definidas y experimentar con ellas. Posibilita además la creación de nuevos ejercicios y facilita la evaluación del contenido con un alto grado de fiabilidad.
6. En el diseño del OA-TG se logró un alto componente visual donde se brinda la posibilidad de gestionar los grafos visualizándolos de manera gráfica. Al igual que en el OA-TC se diseñó una forma de evaluación dinámica que brinda al profesor una gran variabilidad en los ejercicios permitiendo una evaluación diferenciada cada vez que se use el objeto de aprendizaje.

Capítulo 3: Implementación y validación de los objetos de aprendizaje

El presente capítulo tiene como objetivos traducir los requerimientos identificados en las funcionalidades que permitan describir las diferentes historias de usuario a implementar en cada OA-IE, permitiendo estimar tiempo y riesgo de desarrollo. A lo largo del capítulo se presentan algunos de los artefactos generados en el proceso descrito anteriormente. Finalmente se diseñan las pruebas aplicadas, para verificar las funcionalidades de cada OA-IE como producto de software para dar paso al proceso de la evaluación como producto pedagógico usando la guía de evaluación propuesta.

3.1 Implementación de los OA

Para los OA-IE desarrollados existen una serie de Funcionalidades (F) y Requisitos no funcionales (RNF) elaboradas a partir del criterio de los profesores que colaboraron con la investigación, definiendo así las funcionalidades que se espera estén implementadas en cada uno de los OA-IE.

3.1.1 Lista de funcionalidades de los OA.

Seguidamente se muestra una lista con las funcionalidades identificadas para el prototipo genérico, las cuales serán heredadas por cada uno de los OA – IE propuestos.

Lista de funcionalidades de la Plantilla (FP)

1. FP 1 Navegar por la información
2. FP 2 Acceder a los marcos de trabajo de las diferentes unidades del OA
3. FP 3 Mostrar reporte de evaluación (nota)
4. FP 4 Mostrar información sobre el applet
5. FP 5 Mostrar ayuda

A continuación se muestran un listado de las funcionalidades que serán añadidas a la plantilla para conformar cada uno de los OA – IE diseñados, estas funcionalidades son con las que contará cada OA-IE.

Lista de funcionalidades del OA - TC (FOAC)

1. FOAC 1 Realizar operación con conjuntos
2. FOAC 2 Evaluar el contenido de la TC
3. FOAC 3 Graficar conjuntos
4. FOAC 4 Gestionar conjuntos

Lista de funcionalidades del OA - MT (FOAMT)

1. FOAMT 1 Experimentar con ejemplos de MT
2. FOAMT 2 Experimentar con ejercicios propuestos de MT
3. FOAMT 3 Evaluar el contenido de MT
4. FOAMT 4 Crear un nuevo ejercicio de MT

Lista de funcionalidades del OA - TG (FOATG)

1. FOATG 1 Mostrar la matriz de adyacencia
2. FOATG 2 Representar gráficamente el grafo
3. FOATG 3 Evaluar contenido de la TG
4. FOATG 4 Gestionar los grafos

3.1.2 Requisitos no funcionales de los OA

Como se menciona anteriormente además de las funcionalidades que representan las necesidades reales que tiene el cliente, también se cuenta con los RNF. Los RNF definen las propiedades y restricciones del entorno donde estarán disponibles los OA una vez finalizados. Los RNF tienen que ver con características que de una u otra forma puedan limitar los OA, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares. A continuación se muestran los requisitos no funcionales identificados para los OA propuestos.

Usabilidad: Los OA pueden ser usados por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Portabilidad: Los OA deben ser multiplataforma y disponibles en una plataforma web. Son de fácil mantenimiento.

Disponibilidad: Se garantiza a los usuarios el acceso a los OA en todo momento. Los OA están diseñados para su funcionamiento constante, la disponibilidad de los mismos está condicionada a los recursos y tipo de despliegue en los servidores del EVA / UCI, así como los ciclos de mantenimiento determinados.

Legales: El producto es un conjunto de OA para apoyar el PEA de MD en la UCI, se rige por normas internacionales y cumple con las normas y leyes establecidas en el país y en la universidad. Las tecnologías escogidas para el desarrollo de la aplicación, están basadas en la licencia GNU/GPL y en el

caso de la herramienta Visual Paradigm, la universidad cuenta con una licencia libre.

Restricciones: Se utiliza Metodología Ágil XP, puesto que responde a los cambios rápidamente. Como herramienta de apoyo el Visual Paradigm para el modelado de los diagramas UML.

3.1.3 Historias de usuario.

Uno de los artefactos generados por la metodología XP son las historias de usuarios (HU). Las HU son creadas a partir de las funcionalidades definidas. En las siguientes tablas se mostrarán algunas de las HU de los diferentes OA, que presentan prioridad alta conformando parte de las funcionalidades principales de los OA. Las HU mostradas a continuación se corresponden con las funcionalidades “Realizar operaciones entre conjuntos”, “Experimentar con ejercicios propuestos” y “Mostrar Matriz de adyacencia” pertenecientes a los OA-TC y, OA-MT y OA-TG respectivamente. Las HU correspondientes se muestran en una plantilla modificada por el autor, las restantes pueden encontrarse en el anexo 4.

Historia de usuario “Calcular intersección” del OA-TC

Historia de Usuario	
Número: HUTC1	Usuario: Estudiante, Profesor
Nombre Historia: Calcular intersección	Referencia: FOAC1
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite calcular la intersección entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Intersección” en el menú del área de experimentación y se seleccionan los conjuntos para realizar la operación. Para proceder a mostrar el resultado de la operación existen dos variantes. Una primera variante es luego de haber seleccionado los conjuntos proceder a mostrar directamente el resultado que ofrece el OA. En la segunda variante es el usuario quien introduce la respuesta al ejercicio. En ambas variantes el resultado es mostrado mediante diagramas de Venn.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Para la segunda variante en caso de que la respuesta introducida por el usuario no tenga el formato correcto o deje el campo en blanco se muestra un mensaje “Formato de la respuesta incorrecto”. 	

Tabla 5. HU Calcular intersección

Historia de usuario “Resolver ejercicios propuestos” del OA-MT

Historia de Usuario	
Número: HUMT1	Usuario: Estudiante, Profesor
Nombre Historia: Resolver ejercicios propuestos	Referencia: FOAMT2
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite resolver ejercicios de Máquinas de Turing propuestas. Para realizar la operación el usuario selecciona la opción “Ejercicios propuestos” en el menú del área de experimentación, se selecciona el ejercicio a resolver en una lista de ejercicios y se muestra la orden del mismo. Una vez mostrado el ejercicio el usuario entra los datos de las funciones, del estado inicial y los estados finales. Las funciones y estados finales se adicionarán a una lista de funciones y de estados respectivamente, donde podrán ser visualizadas por el usuario. Luego de finalizar el ejercicio se comprueba la respuesta donde la aplicación informa al usuario si la solución está correcto o no.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. En caso de dejar algunos de los datos de las funciones en blanco se muestra un mensaje “Debe llenar todos los campos”. 2. En caso de dejar el campo de estados finales en blanco se muestra un mensaje “Debe llenar el campo”. 	

Tabla 6. HU Resolver ejercicios propuestos

Historia de usuario “Generar matriz de adyacencia” del OA-TG

Historia de Usuario	
Número: HUTG1	Usuario: Estudiante, Profesor
Nombre Historia: Generar matriz de adyacencia	
Referencia: FOATG1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite a los usuarios generar la matriz de adyacencia de un determinado grafo. Para realizar la operación el usuario selecciona la opción “Matriz de adyacencia” en el menú referente al panel de experimentación. Seguidamente aparecerá una interfaz donde se podrá seleccionar el grafo a visualizar tanto de la lista de grafos creados por el usuario como de la lista de grafos generados para las evaluaciones. Una vez seleccionado el grafo se muestra su matriz de adyacencia correspondiente.</p>	
Observaciones:	

Tabla 7. HU Generar matriz de adyacencia

A partir de la identificación de las funcionalidades y de la descripción de las HU por cada característica principal de los OA, fue posible realizar estimaciones de tiempo para la implementación y el plan de pruebas. Quedaron definidas siete (7) iteraciones para la realización de los OA. Las iteraciones se estructuraron de la siguiente manera: una (1) iteración para la implementación y prueba de la Plantilla genérica y dos (2) iteraciones para el resto de los OA (OA-TC, OA-MT, OA-TG), en la primera iteración se implementarán y probarán las funcionalidades correspondientes a las HU de prioridad alta, y en la segunda iteración se implementarán y probarán las funcionalidades correspondientes a las HU de prioridad media y baja.

3.1.3.1 Planificación de Historias de Usuario

No.	Nombre de Historia de usuario	Prioridad	Iteración
1	Navegar por la información	Media	1
2	Acceder a los marcos de trabajo	Media	1
3	Mostrar reporte de evaluación	Media	1
4	Mostrar información sobre el applet	Baja	1
5	Mostrar ayuda	Baja	1
6	Realizar operación de intersección	Alta	2
7	Realizar Operación de Unión	Alta	2
8	Realizar Operación de Diferencia	Alta	2
9	Realizar Operación de Diferencia Simétrica	Alta	2
10	Realizar Operación de Producto cartesiano	Alta	2
11	Realizar Operación de Conjunto potencia	Alta	2
12	Evaluar contenido de TC	Alta	2
13	Graficar Conjuntos	Media	3
14	Crear conjunto	Media	3
15	Eliminar conjunto	Media	3
16	Adicionar elemento a un conjunto	Media	3
17	Eliminar elemento a un conjunto	Media	3
18	Experimentar con cadenas	Alta	4

No.	Nombre de Historia de usuario	Prioridad	Iteración
19	Tracear cadenas	Alta	4
20	Resolver ejercicios propuestos	Alta	4
21	Evaluar contenido de MT	Alta	4
22	Crear nuevo ejercicio de MT	Media	5
23	Generar matriz de adyacencia del grafo	Alta	6
24	Generar representación gráfica del grafo	Alta	6
25	Evaluar contenido de la TG	Alta	6
26	Crear grafo	Media	7
27	Eliminar grafo	Media	7
28	Adicionar vértices al grafo	Media	7
29	Eliminar vértices del grafo	Media	7
30	Adicionar aristas al grafo	Media	7
31	Eliminar aristas al grafo	Media	7

Tabla 8. Planificación de Historias de Usuario

En el anexo 5 se muestra un diagrama de Gantt que responde al plan de iteraciones. El cronograma contempla las tareas a llevar a cabo para dar cumplimiento al objetivo de la investigación y fue elaborado a partir de la planificación de las HU.

3.1.4 Patrón de Diseño

Modelo Vista Controlador (MVC)

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:** Representa la información con la que trabaja la aplicación, es decir, su lógica del negocio. En el caso de la presente investigación los datos son almacenados en clases que representan los contenidos con los que se trabaja en cada OA. El comportamiento de estas clases es por ejemplo: Máquina de Turing, Conjunto y Grafo, o sea clases implementadas y que gestionan una o varias instancias de estas clases.

- **Vista:** Presenta el modelo en un formato adecuado, como en una página web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario. En el caso de la presente investigación la interfaz está compuesta por dos regiones; una a la izquierda que contiene un menú desde el cual se puede acceder a los diferentes marcos de trabajo diseñados para cada uno de los OA que conforman la segunda región y que se encuentran a la derecha. Esta interfaz es cargada en un navegador web cuando se ejecuta el applet.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista. En el caso de la presente investigación las clases controladoras contienen las principales funcionalidades, algoritmos y operaciones implementadas para los diferentes contenidos que se manejan en los OA propuestos.

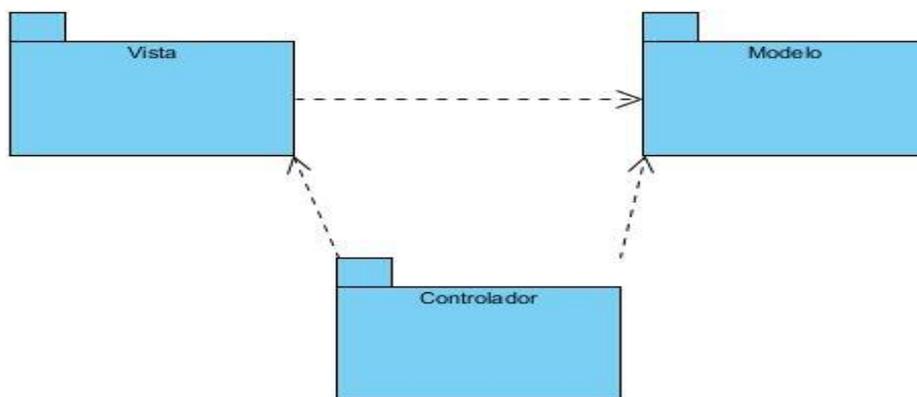


Figura 19. Diagrama de paquetes

3.1.5 Modelo de Despliegue

El modelo de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos (Universidad Salesiana de Bolivia).

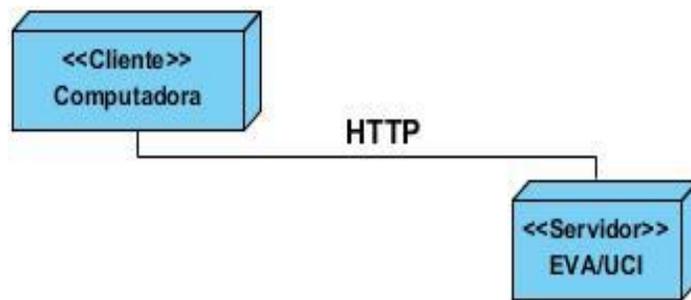


Figura 20. Modelo de despliegue

3.2 Validación de los OA

Estrategia de pruebas desde la perspectiva del OA como software

Para realizar las pruebas de los OA se tomaron en cuenta dos niveles de ellas, las pruebas unitarias con sus métodos de caja blanca y las pruebas de sistema. La puesta en práctica de las pruebas de caja blanca requiere del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones del diseño o el código. Se basa en la comprobación de los caminos lógicos del software dado un código específico. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Entre las pruebas de sistema aplicables a los OA desarrollados se incluyen las pruebas funcionales que permiten asegurar que los OA realizan correctamente todas las funciones detalladas en las Historias de Usuarios. Se incluyen además las pruebas de comunicación que en el caso de la presente investigación se centran fundamentalmente en probar las interfaces hombre/máquina. Por otro lado las pruebas de facilidad de uso permiten comprobar la adaptabilidad de los OA a las necesidades de los usuarios para asegurar que se acomoda a su modo habitual de trabajo. Finalmente dentro de las pruebas de sistemas están las pruebas de operación para comprobar la correcta implementación de los procedimientos de operación, incluyendo la planificación y control de trabajos.

Diseño de casos de pruebas

El diseño consiste en la confección de los distintos casos de prueba según la técnica identificada previamente. Cada caso de prueba va acompañado del resultado que ha de producir el software al ejecutarlo para detectar un posible fallo en el mismo. Definen un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada método de prueba proporciona distintos

criterios para generar estos casos o datos de prueba (Peñalver Romero, 2008).

La **prueba de la caja blanca** es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecuten al menos una vez todos los caminos independientes de cada módulo
- Se utilicen las decisiones en su parte verdadera y en su parte falsa
- Se ejecuten todos los bucles en sus límites
- Se utilicen todas las estructuras de datos internas

Dentro de las técnicas que se utilizan en las pruebas de caja blanca se encuentra el método del camino básico que permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez. La complejidad ciclomática forma parte del método del camino básico, para aplicar esta técnica se deben de resolver varias fórmulas, de las cuales se deben obtener igual resultado para asegurar que el código analizado no presenta ningún tipo de error. En la figura 21 se muestra el método diferencia () del OA-TC al cual le fue aplicado la prueba de caja blanca, con 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.

```
public Conjunto diferencia(Conjunto AR, Conjunto BR) {  
    Conjunto R = null;  
    ListaSE<String> con = new ListaSE<String>();  
  
    for (int i = 0; i < AR.getElementos().Longitud(); i++) {  
        if (esta_elemento(BR, AR.getElementos().Obtener(i)) == -1) {  
            con.Adicionar(AR.getElementos().Obtener(i));  
        }  
    }  
    R = new Conjunto(con, "R");  
    return R;  
}
```

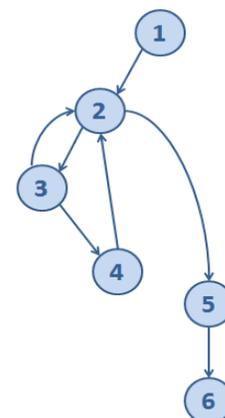


Figura 21. Método diferencia () del OA-TC y grafo de ejecución

Fórmulas para calcular complejidad ciclomática:

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

Siendo "A" la cantidad total de aristas y "N" la cantidad total de nodos.

Se puede usar también:

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

Siendo "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

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

Siendo "R" la cantidad total de regiones, para cada fórmula "V (G)" representa el valor del cálculo.

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 3, lo que significa que existen tres posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo.

- Camino #1: 1-2-5-6
- Camino #2: 1-2-3-2-5-6
- Camino #3: 1-2-3-4-2-5-6

En la figura 22 se muestra la aplicación de la prueba de caja blanca al método `si_es_final ()` del OA-MT. El análisis de la funcionalidad arrojó como resultado el grafo de ejecución al cual se le aplicó la técnica de la complejidad ciclomática.

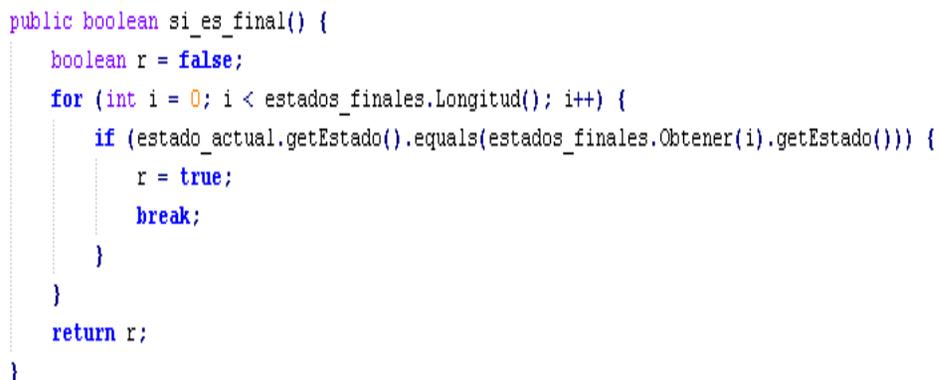


Figura 22. Método `si_es_final ()` del OA-MT y grafo de ejecución

Fórmulas para calcular complejidad ciclomática:

$$V(G) = (A - N) + 2 \quad V(G) = (6-5) + 2 \quad V(G) = 3$$

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

Se puede usar también:

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

Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

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

Siendo “R” la cantidad total de regiones, para cada formula “V (G)” representa el valor del cálculo.

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 3, lo que significa que existen tres posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo.

- Camino #1: 1-2-5
- Camino #2: 1-2-3-4-5
- Camino #3: 1-2-3-2-5

En la figura 23 se muestra la aplicación de la prueba de caja blanca al método grafo_simple () del OA-TG. El análisis de la funcionalidad arrojó como resultado el grafo de ejecución al cual se le aplicó la técnica de la complejidad ciclomática.

```
public boolean grafo_simple() {  
    boolean res = true;  
    for (int i = 0; i < matriz_ady.length; i++) {  
        for (int j = 0; j < matriz_ady.length; j++) {  
            if (matriz_ady[i][j] > 0 && i == j) {  
                res = false;  
                break;  
            }  
        }  
    }  
    return res;  
}
```

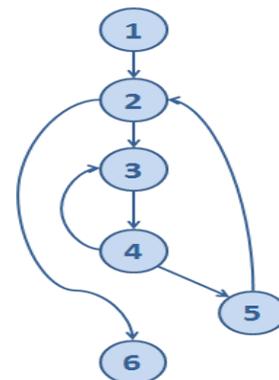


Figura 23 Método grafo_simple () del OA-TG y grafo de ejecución

Fórmulas para calcular complejidad ciclomática:

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

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

Se puede usar también:

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

Siendo "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

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

Siendo "R" la cantidad total de regiones, para cada fórmula "V (G)" representa el valor del cálculo.

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 3, lo que significa que existen tres posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo.

- Camino #1: 1-2-6
- Camino #2: 1-2-3-4-5-2-6
- Camino #3: 1-2-3-4-3-4-5-2-6

A partir del valor de la complejidad ciclomática se obtiene el número de caminos independientes, que brindan un valor límite para el número de pruebas que se deben diseñar.

Validación de las funcionalidades

Para la validación de las funcionalidades se aplicaron pruebas de sistema. En las iteraciones 2, 4 y 6 se probaron las historias de usuario que tienen una prioridad Alta para los OA TC, MT y TG respectivamente. Para los tres OA se cuentan con un total de 14 HU que presentan esa prioridad, para el OA-TC son las referentes a las funcionalidades: Realizar operaciones entre conjuntos y Evaluar contenido de TC, para OA-MT son: Experimentar con ejemplos de MT, Experimentar con ejercicios propuestos de MT y Comprobar solución de MT y para el OA-TG son: Mostrar matriz de adyacencia, Representar gráficamente grafos y Evaluar contenido TG. En el presente capítulo solamente se hace referencia a algunas de las HU referentes a las funcionalidades "Evaluar contenido de TC" del OA-TC, "Experimentar con ejercicios propuestos" del OA-MT y "Evaluar contenido de TG" del OA-TG, el resto de las pruebas de sistema se pueden encontrar en el anexo 6.

Caso de prueba. HU Evaluar intersección.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Intersección. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de evaluación “Intersección”.**Escenario:** “Realizar evaluación Intersección”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Intersección entre conjuntos (el usuario selecciona para cada inciso los conjuntos para realizar la operación y llena los campos con las posibles respuestas en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje indicando el error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje indicando el error especificando el o los incisos que contiene el error.		

Tabla 9. Caso de prueba HU “Realizar evaluación Unión”

Caso de prueba. HU Resolver ejercicio de MT.

Descripción de la funcionalidad: En esta historia el usuario resuelve ejercicios propuestos de MT. Si el usuario selecciona un ejercicio tiene la posibilidad de entrar los datos correspondientes para la solución del mismo. Si los datos entrados son correctos son guardados como parte de la solución. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido. La entrada de datos está limitada a conveniencia del usuario. Luego de finalizar la entrada de datos se comprueba si la solución del ejercicio es correcta.

Condiciones de ejecución:**Flujo central:** Opción del menú de experimentación “Ejercicios propuestos”.**Escenario:** “Resolver ejercicios propuestos de MT”.

Clases Válidas	Resultado esperado	Resultado	Observaciones
Resolver ejercicios propuestos	Luego de entrar los datos		

de MT (el usuario selecciona el ejercicio y entra los datos requeridos: funciones de transición, estado inicial y estados finales y comprueba la solución).	de la solución, si el proceso ha sido correcto, se muestra un mensaje al usuario indicando que los datos fueron guardados como parte de la solución.		
Clase inválida	Resultado esperado	Resultado	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error indicando el error.		

Tabla 10. Caso de prueba HU “Resolver ejercicio de MT”

Caso de prueba. HU Evaluar camino simple.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en el contenido de TG de camino simple. Si los datos entrados por el usuario tienen el formato correcto cuando se termine la evaluación estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Camino simple”.

Escenario: “Realizar evaluación de camino simple”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación sobre la habilidad para identificar caminos simples en grafos (el usuario entra los datos correspondientes para los encisos que requieran entrada de datos y selecciona la respuesta en los encisos que requieran selección).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje indicando el error.		

Respuestas con formatos incorrectos.	Se debe mostrar un mensaje indicando el error especificando el o los incisos que contienen error.		
Inciso de selección simple sin seleccionar.	Se debe mostrar un mensaje indicando el error.		

Tabla 11. Caso de prueba HU “Realizar evaluación de camino simple”

Evaluación de la calidad como producto pedagógico

Para evaluar la calidad de los OA se utilizó la guía de (Toll Palma, 2011), aunque se modificó el apartado referente al aspecto tecnológico, donde solo se evalúan 10 aspectos. Los aspectos eliminados son los referentes a los metadatos. A partir de la modificación de la sección mencionada anteriormente se modifica la forma de asignar la categoría de la evaluación, la forma de calcular los nuevos valores se precisa en la propia metodología seguida, la tabla siguiente muestra los valores relacionados con la escala.

Rangos de escala	Nivel de Calidad alcanzado por el OA
$E \geq 33$	Muy Adecuado
$30 \leq E < 33$	Adecuado
$21 \leq E < 30$	Poco adecuado
$E < 21$	No adecuado

Tabla 12. Rangos evaluativos para validar la calidad

La evaluación de los OA fue realizada por seis profesores del colectivo de Matemática Discreta, se seleccionaron por llevar más de tres años impartiendo la asignatura y dominar las tecnologías. A los evaluadores se les entregó un fichero MS Excel con los criterios y se les explicó que significaba cada uno de ellos y la forma de evaluarlos. Luego de consolidar la información del **OA Teoría de Conjuntos** se obtuvo la siguiente tabla:

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
Formativo	17.14	16.71	17.14	17.57	17.14	17.14
Diseño	8.29	8.57	8.00	8.00	8.29	8.29
Técnico	7.71	8.00	8.00	7.14	7.43	7.71
Evaluación	33.14	33.29	33.14	32.71	32.86	33.14

Tabla 13. Calificación de evaluadores para el OA-TC

A partir de estos valores cuatro de los seis profesores consideran que el OA tiene categoría de **muy adecuado**, lo que es un resultado que valida la importancia y el valor del OA. Solo en el aspecto de los metadatos y de la facilidad de indexado en un repositorio las evaluaciones fueron bajas. En el anexo 7 pueden verse las evaluaciones de los expertos.

En el caso del OA-TG los resultados fueron mejores pues los seis catalogan el OA de **muy adecuado**. Esto se debe fundamentalmente a que en el OA se incluye más información que la que se trabaja en la asignatura, ello se debe a que el grafo se trabaja en otras asignaturas de la carrera y en la MD hay conceptos que no son trabajados.

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
Formativo	17.57	17.14	17.57	17.57	17.57	17.14
Diseño	8.57	8.57	8.00	8.00	8.57	8.29
Técnico	8.00	8.29	8.00	7.71	8.29	8.00
Evaluación	34.14	34.00	33.57	33.29	34.43	33.43

Tabla 14. Calificación de evaluadores para el OA-TG

En el caso del OA-MT cuatro de los seis profesores consideran que el OA tiene categoría de **muy adecuado**. Esto se debe a que los profesores ya cuentan con un recurso que permite experimentar y ver de manera gráfica a las MT.

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
Formativo	18.00	16.29	17.57	17.57	17.14	17.14
Diseño	8.57	8.57	8.29	8.29	8.57	8.00
Técnico	8.00	8.00	8.00	7.43	8.00	7.71
Evaluación	34.57	32.86	33.86	33.29	33.71	32.86

Tabla 15. Calificación de evaluadores para el OA-MT

Como software los OA mantienen una funcionalidad muy aceptable cumpliendo con los requerimientos establecidos por el usuario así como con los requisitos no funcionales. Los mismos se implementaron utilizando una metodología ágil factible para su desarrollo. Se logró solucionar todas las No Conformidades encontradas en cada una de las tres iteraciones de prueba realizadas. Desde el punto de vista pedagógico se pudo constatar la aceptación que tuvieron los OA dentro de un grupo de expertos en materia docente, validando así su utilización para facilitar el Proceso de Enseñanza Aprendizaje de la Matemática Discreta.

Los resultados de la evaluación de los OA tanto desde el punto de vista de producto de software y como producto pedagógico permiten constatar la calidad de los mismos y del proceso que se siguió en su construcción.

Conclusiones parciales

En este capítulo se expusieron los resultados de la implementación de los OA-IE propuestos, a partir de los artefactos elaborados, para ello se llevaron a cabo estrategias de pruebas que permitieran obtener un producto de calidad. Una vez realizadas un grupo de entrevistas a profesores de MD respecto a los requisitos que debían cumplir los OA-IE desarrollados para comprobar su calidad, se llega a las siguientes conclusiones:

1. Se detectaron 17 funcionalidades, que se derivaron en 31 HU y se clasificaron de acuerdo a su prioridad, donde 14 de ellos obtuvieron la prioridad alta, 15 media y 2 de baja. A partir de ahí se definió el plan de iteraciones, estimando el tiempo de desarrollo del mismo en 18 semanas.
2. Se describió y validó cada HU con los profesores de MD que colaboran con la investigación antes de pasar a la fase de desarrollo.
3. Se obtuvieron OA-IE de fácil uso y comprensión para todos los usuarios con acceso a los mismos, el acceso a los OA-IE se puede realizar en todo momento y desde cualquier lugar en la universidad, garantizando su accesibilidad e interoperabilidad.
4. La realización de varias iteraciones de pruebas unitarias de caja blanca y pruebas de sistema permitió la detección de (32) No Conformidades (NC) que fueron introducidas durante la etapa de codificación y corregidas en iteraciones posteriores. Las NC más relevantes fueron referentes a: faltas de ortografía, no correspondencia de lo descrito en las historias de usuario con el funcionamiento de los OA, errores de validación en la entrada de datos, errores en la navegación por la información y en la aplicación de los métodos de evaluación.
5. La aplicación de la guía de evaluación de la calidad de Toll (Toll Palma, 2011) permitió evaluar la calidad de los OA-IE implementados desde el punto de vista del pedagógico a partir del criterio de diferentes profesores, estos se seleccionaron a partir de su experiencia en la docencia de la asignatura. El proceso arrojó como resultado que cada uno de los OA desarrollados son muy adecuados según el valor obtenido en la escala propuesta en la guía utilizada.

Conclusiones

El desarrollo de la presente investigación hizo posible la realización de un conjunto de Objetos de Aprendizaje interactivos y experimentales. Para ello se cumplió con el objetivo general a través de las tareas de investigación propuestas por lo que se pudo arribar a las siguientes conclusiones:

1. Se analizó la importancia de la MD en las Ciencias de la Computación como base de gran parte de los procesos digitales ya que ofrece fundamentos matemáticos para diferentes aspectos de la informática. Se estudiaron además las posibilidades que brinda el EVA como parte del entorno tecnológico en la UCI y se analizaron los elementos de la MD disponibles en este espacio, donde se detectaron problemas en cuanto al aprovechamiento de TIC en el PEA de la MD.
2. Luego de un análisis de diferentes autores que han realizado estudios sobre los OA se definió como OA a toda pieza digital de material educativo, cuyo principal potencial es la reutilización dentro de distintos contextos aplicables a la educación. Para su desarrollo se seleccionó como IDE el Netbeans que en conjunto con el lenguaje Java permiten la construcción de applets. Como metodología de desarrollo de software se seleccionó XP e ISD-MeLO como metodología enfocada en el diseño y desarrollo de contenido educativo. Para la evaluación de la calidad de los OA se eligió la guía de evaluación de Toll (Toll Palma, 2011) para evaluar el OA desde el punto de vista pedagógico la cual se complementó con pruebas de software.
3. A partir de una encuesta realizada a algunos de los profesores de MD se evidenció el poco uso de OA en el PEA de la MD debido a la poca existencia de estos recursos en la UCI. El 89 % de los profesores de MD plantea nunca haber usado OA, de ellos el 63 % expresa que no existen los adecuados mientras que al 28% les resulta difícil adaptarlos a las clases. A pesar de ello los profesores plantean que sería factible la utilización de OA con un alto grado de interactividad que permitan experimentación.
4. Se realizó una encuesta a los profesores de MD para seleccionar los contenidos para el desarrollo de OA. El procesamiento de la encuesta permitió seccionar los temas: Máquinas de Turing, Teoría de Conjuntos y Teoría de Grafos. Algunos de los parámetros evaluados fueron: grado de complejidad, posibilidades de experimentar y resultado en las evaluaciones. De manera general se potenciaron las posibilidades de representación gráfica de estos contenidos.

5. El diseño de los OA partió de definir una Unidad de Información, donde se brinda la posibilidad de la 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.
6. Las funcionalidades de los OA fueron identificadas con la ayuda de un grupo de profesores de MD vinculados a la investigación. Estas funcionalidades fueron traducidos en historias de usuario que posteriormente fueron implementadas logrando como resultado un conjunto de OA de fácil uso y con un alto nivel de interacción desde el punto de vista experimental y evaluativo.
7. Se diseñaron pruebas a realizar sobre los escenarios seleccionados para comprobar y validar las funcionalidades de cada OA como software, en muchos casos permitieron corregir las no conformidades encontradas. Se validó además el objeto de OA desde el punto de vista pedagógico a través de un panel de experto que evaluaron los OA mediante la guía de evaluación seleccionada. Se obtuvo una evaluación final de muy adecuada para cada uno de los OA desarrollados, siendo la base para asegurar que los OA están listos para ser puestos a disposición de la comunidad universitaria.

Recomendaciones

1. Ampliar el objeto de aprendizaje vinculado al contenido de Máquinas de Turing buscando la forma de graficar la misma, se propone animar la gráfica de la misma durante el proceso de reconocimiento de una cadena, esto permitiría que el estudiante se lleve una mejor idea de su funcionamiento.
2. En el caso del objeto de aprendizaje grafo adicionar los contenidos vinculados a la asignatura de programación, pudiendo permitir ejecutar paso a paso los algoritmos para calcular el camino mínimo entre dos vértices o la construcción del árbol de expansión mínima.
3. Generalizar el prototipo de applet mediante la implementación de otros objetos de aprendizaje que tengan como características ser interactivos y experimentales.
4. Exportar el resultado de la evaluación para que el profesor pueda saber los resultados de las evaluaciones realizadas por los estudiantes, o para que el propio estudiante valore su desarrollo.

Bibliografía

- López Guzmán, Clara. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning.* España : s.n., 2005.
- Murillo Alfaro, Félix. 1999.** *Herramientas Case.* 1999.
- Beck, Robert. 2008.** *What are learning objects?* Wisconsin-Milwaukee : s.n., 2008.
- Björne, R y Stanley. 1999.** *A combinatory miscellany.* Cambridge : s.n., 1999.
- Canavelli, otros y J, C. 2007.** *Discrete Mathematics And Digital Technology.* ICME, the International Congress on Mathematical Education. [En línea] 2007. [Citado el: 8 de 12 de 2011.] <http://icme11.org/node/650>.
- Carrazana Cordero, Delia Rosa . 2009.** *Multimedia de Matemática Discreta.* 2009.
- Chan, María Elena. 2011.** *OBJETOS DE APRENDIZAJE: una herramienta para la innovación educativa.* 2011.
- de la Torre, Anibal. 2006.** *Introducción a la plataforma Moodle.* 2006.
- . **2006.** *Plataforma Moodle: Gestión y administración de un curso.* 2006.
- . **2006.** *Plataforma Moodle: Primeros pasos con Moodle.* 2006.
- Dpto CCIA. 2005.** *Lenguaje Java y Entorno de Desarrollo.* 2005.
- s/f.** *El lenguaje C# y la plataforma .NET.* s/f.
- Fuentes, Mompié y Cisnero, Iyatne y Chacón, Claritza. 2009.** *“Módulo de objetos interactivos de aprendizaje para apoyar el proceso de enseñanza-aprendizaje en la disciplina de Matemática Aplicada.* La Habana : s.n., 2009.
- García Alba, María Jesús. 2010.** *Análisis del desarrollo de extensiones paraMoodle: Desarrollo de un módulo para la gestión delaboratorios docentes.* 2010.
- García de Jalón, Javier, y otros. 2000.** *Aprende Java como si estuviera en primero.* San Sebastián : s.n., 2000.
- García Hernández, Alién , Chirino Esquijarosa, Dariel y Cid Escalona, Lilian. 2011.** *La guía de ejercicios. Instrumento para el estudio de la Matematica Discreta en el modelo de formacion centrado en el aprendizaje.* 2011.
- García Hernández, Alién y Chirino Esquijarosa, Dariel. 2011.** *La guía de ejercicios, herramienta en el modelo de Formación centrado en el aprendizaje.* 2011.
- García Hernández, Alién, Verdecia, Edistio Yoel y Chirino Esquijarosa, Dariel. 2011.** *Objetos de Aprendizaje para la enseñanza de la Matematica Discreta en el modelo de formacion centrado en el aprendizaje.* 2011.
- Grupo Satelite s.a de c.v. 2011.** [En línea] 2011. [Citado el: 8 de 12 de 2011.] http://www.gruposatelite.net/index.php?option=com_content&view=article&id=66&Itemid=69.
- H Canós, José, Letelier, Patricio y Penadés, María del Carmen. s/f.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n., s/f.

2011. IEEE. *Learning Object Metadata Working Group*. [En línea] 2011. [Citado el: 11 de 12 de 2011.] <http://ltsc.ieee.org/wg12/index.html>.
- López García, Pablo y Sein-Echaluce Lacleta, María Luisa. s/f.** *MOODLE: Difusión y funcionalidades*. s/f.
- López, Clara. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. 2005.
- Luján Mora, Sergio. s/f.** *C++ Paso a Paso*. s/f.
- Marion, Bill y Baldwind, Doug. 2007.** *On the Implementation of the Discrete Mathematic course*. 2007.
- Martínez, Alejandro y Martínez, Raúl. s/f.** *Guía a Rational Unified Process*. s.l. : Albacete, s/f.
- Mason y otros. 2003.** *Learning in the Connected Economy*. Londres : Open University, 2003.
- Merrill, M David. 1999.** *INSTRUCTIONAL TRANSACTION THEORY (ITT):INSTRUCTIONAL DESIGN BASED ON KNOWLEDGE OBJECTS THE INSTRUCTIONAL SIMULATOR OR IDVISUALIZER*. 1999.
- Mestre Gómez, Ulises, Fonseca Pérez, Juan José y Valdés Tamayo, Pedro Roberto. 2007.** *Entornos virtuales de enseñanza aprendizaje*. La Habana : Editorial Universitaria, 2007.
- Microsfot. 2010.** Visual Studio. *Visual Studio 2010 Ultimate*. [En línea] 2010. [Citado el: 12 de 12 de 2011.] <http://www.microsoft.com/spain/visualstudio/products/2010-editions/ultimate/features>.
- Morales Peña, Hugo Humberto. 2011.** *Propuestas del contenido para un primer curso de Matemáticas computacionales en Ingeniería de sistemas*. 2011.
- Morales, Erna. 2005.** Propuesta de Evaluación de Objetos de Aprendizaje. [En línea] 2005. <http://www.uoc.edu/symposia/spdece05/pdf/ID06.pdf>.
- Moya Martínez, Antonia María. 2009.** *Las nuevas tecnologías en la educación*. 2009.
- Oracle Corporation.** NetBeans. [En línea] [Citado el: 8 de 12 de 2011.] www.netbeans.org.
- Ossadán Núñez, Yanko. 2006.** *Propuesta para el diseño de Objetos de Aprendizaje*. Chile : s.n., 2006.
- Peñalver Romero, Gladys Marsi. 2008.** *Metodología ágil para proyectos de software libre*. La Habana: UCI : s.n., 2008.
- Pernalet Chirinos, Doris y Delgado Herrera, Mayela. 2009.** *Desarrollo de un Objeto de Aprendizaje con un Enfoque de Calidad sobre árboles binarios de búsqueda*. 2009.
- PHP. 2011.** PHP. [En línea] 2011. [Citado el: 11 de 12 de 2011.] <http://php.net/manual/es/intro-whatis.php>.
- . 2011. PHP. [En línea] 2011. [Citado el: 12 de 12 de 2011.] <http://www.php.net/manual/es/intro-whatcando.php>.
- Piattini, M. 1996.** *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Madrid : s.n., 1996.
- Pressman, R. 2002.** *Software engineering. A Practitioner's Approach*. Madrid : McGraw-Hill, 2002.

- Prieto, Alberto, Lloris, Antonio y Torres, Juan Carlos. 2004.** *Introducción a la Informática*. Madrid : MAc Graw Hill, 2004.
- Rosado, L y Herreros, J R. 2002.** *Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física*. Madrid : s.n., 2002.
- Rosario, Jimmy. 2005.** *La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual*. 2005.
- Rosen, y otros. 2004.** *Discrete Mathematics and its applications*. 2004.
- Ruiz, Roberto Eduardo. 2006.** *Modelo de Integración de Competencias en Objetos". Tecnologías y Educación a Distancia*. 2006.
- Soto García, Hamlet y Reventos Liranza, Alexander. 2010.** *Módulo de Actividad "Evaluación" para la plataforma MOODLE*. 2010.
- Toll Palma, Yuniet del Carmen. 2011.** *Guía de evaluación de la calidad de los Objetos de Aprendizaje producidos en la Universidad de las Ciencias Informáticas*. 2011.
- UCI. 2011.** *Informe semestral de Matemática Discreta 1. Curso 2010-2011*. Habana : s.n., 2011.
- Universidad Carlos III de Madrid. s/f.** *Ingeniería de Software III*. Madrid : s.n., s/f.
- Universidad Salesiana de Bolivia.** [En línea] [Citado el: 6 de 2 de 2012.] <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc..>
- Velázquez, César. 2005.** *La Importancia de la Definición de la Calidad del Contenido de un Objeto de Aprendizaje*. 2005.
- Vidal Castro, Christian. 2008.** *Calidad en Objetos de Aprendizaje*. [En línea] 2008. <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Christian%20Vidal%20-%20Calidad%20en%20OA%20-%20Pres.pdf>.
- Wiley, David A. 2000.** *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. 2000.
- Xiuguo y Wu. 2009.** IEEE-Digital Library. [En línea] 2009. [Citado el: 8 de 12 de 2011.] [http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4959102&abstractAccess=no&userType=inst.978-1-4244-3581-4 ..](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4959102&abstractAccess=no&userType=inst.978-1-4244-3581-4..)

Anexos

Anexo 1: Encuesta para la selección de los contenidos para desarrollar OA

ENCUESTA DIAGNÓSTICO DE LA SITUACIÓN ACTUAL DEL USO DE OBJETOS DE APRENDIZAJE EN LAS ASIGNATURAS MATEMÁTICA DISCRETA 1 Y 2

Usted ha sido seleccionado para colaborar con la investigación *“Construcción de objetos de aprendizaje para la matemática discreta”*. En tal sentido se elabora esta encuesta cuyo objetivo es: *Diagnosticar la situación actual de la utilización de los Objetos de aprendizaje en las asignaturas Matemática Discreta 1 y 2.*

Por ello le pedimos que lea con detenimiento la información que se le solicita y responda marcando con una x o con una oración breve, se agradece la con sinceridad en su respuestas a las preguntas.

Muchas Gracias por su ayuda....

Facultad: _____ Años de experiencia: _____

Pregunta # 1. Utiliza usted objetos de aprendizaje en sus clases: SI () NO ()

De seleccionar NO diga alguna de las causas:

Pregunta # 2. Llene la siguiente tabla:

Para las columnas que expresen grado escriba (muy bajo, bajo, medio, alto o muy alto), para la de los resultados escriba solo (alto, medio o bajo)

Tema	Grado de complejidad	Posibilidad de representar gráficamente	Posibilidad de experimentar	Grado de dificultad en la asimilación	Grado de dificultad en la impartición	Resultados en evaluaciones
Teoría de conjuntos						
Relaciones binarias						
Lógica						
Teoría de números						
Máquinas de Turing						
Teoría combinatoria						
Teoría de Grafos						
Relaciones de recurrencia						

Anexo 2: Interfaces de los ejercicios evaluativos del OA-TC.

The screenshot shows the 'Teoría de Conjuntos' applet interface. On the left is a navigation tree with 'Evaluación' expanded to show 'Producto cartesiano' and 'Conjunto Potencia'. The main area is divided into two sections:

- Producto cartesiano:** Shows two sets, $S = \{f, d, k, w, 4, 8, z, 3, 6, 5\}$ and $D = \{1, 3, 5, 1, d, 4, a, x, 2\}$. Below them are dropdown menus for 'S' and 'D' followed by an 'X' operator and an equals sign leading to an empty text box for the result. An example shows $(a,b), (a,c)$. A 'Generar Conjuntos' button is present.
- Conjunto Potencia:** Shows a set $Q = \{e, o, l\}$. Below it is a dropdown menu for 'Q' followed by an equals sign and an empty text box for the result. An example shows $\emptyset, \{a, y, 7, 4\}, \{1, 4, 8, o, p\}, \{1, 7\}, \{8\}$.

At the bottom right, a timer shows '3 : 14 : 10' and 'Copyright © 2012 UCI.' is at the bottom center.

Figura 24. Interfaz de evaluación Producto Cartesiano y Conjunto Potencia

The screenshot shows the 'Teoría de Conjuntos' applet interface for the 'Intersección' exercise. The navigation tree on the left has 'Evaluación' expanded to 'Intersección'. The main area contains:

- Intersección:** Labeled 'Ej 2:', it lists six problems (a-f) using dropdown menus for sets and the intersection symbol \cap .
 - a) $D \cap F = \{ \}$
 - b) $T \cap M = \{ \}$
 - c) $E \cap Q = \{ \}$
 - d) $C \cap I \cap H = \{ \}$
 - e) $L \cap [B \cap U] = \{ \}$
 - f) $[N \cap P] \cap [V \cap R] = \{ \}$
 An example shows $1, 2, q, 4, 5, a, 3, e, a$. A 'Generar Conjuntos' button is present.
- Lista de Conjuntos:** A scrollable list of sets:
 - $D = \{0\}$
 - $F = \{8, m, 2\}$
 - $T = \{5, q, e\}$
 - $M = \{0, t, f, 1\}$
 - $Q = \{2, x, 0, 4, z, i, y\}$
 - $E = \{7, c, i, u, 5, p, 2\}$
 - $S = \{p, 2, r, t, 3, 6, 4, o\}$
 - $K = \{f, m, 4, 6, 2, 8, 7, y\}$
 - $A = \{5, 7, k, m, 0, 2, i, o, h\}$
 - $G = \{j, 6, 3, 4, m, o, h, b, u\}$

At the bottom right, a timer shows '3 : 10 : 29' and 'Copyright © 2012 UCI.' is at the bottom center.

Figura 25 Evaluación de Intersección

Teoría de Conjuntos

- Conjuntos
 - Teoría
 - Experimentación
 - Evaluación
 - Intersección
 - Unión
 - Diferencia
 - Diferencia Simétrica**
 - Combinados
 - Otros
 - Reporte de notas
 - Sobre el Applet

Diferencia Simétrica

Ej 4:

a) Δ = { }

b) Δ = { }

c) Δ = { }

d) Δ Δ = { }

e) Δ [Δ] = { }

f) [Δ] Δ [Δ] = { }

Ejemplo: 1,2,q,4,5,d,3,e,a

Lista de Conjuntos

F = \emptyset
 D = {8, z}
 A = {g, 4, b}
 L = {2, 0, 5, y}
 O = {n, e, 2, 0}
 U = {8, c, 1, 3}
 R = {d, 0, z, 1, 2}
 B = {6, 1, m, f, x}
 M = {4, 5, 3, j, y, d, 1}
 C = {v, i, 3, 6, 5, 2, 7, 0}

Copyright © 2012 UCI. 3 : 11 : 29

Figura 26. Interfaz de evaluación de Diferencias simétrica

Teoría de Conjuntos

- Conjuntos
 - Teoría
 - Experimentación
 - Evaluación
 - Intersección
 - Unión
 - Diferencia**
 - Diferencia Simétrica
 - Combinados
 - Otros
 - Reporte de notas
 - Sobre el Applet

Diferencia

Ej 2:

a) \ = { }

b) \ = { }

c) \ = { }

d) \ \ = { }

e) \ [\] = { }

f) [\] \ [\] = { }

Ejemplo: 1,2,q,4,5,d,3,e,a

Lista de Conjuntos

K = {c}
 R = {u, y}
 U = {5, 8}
 P = {2, o, 8, 6}
 L = {e, x, p, i, 1, 0, h}
 G = {a, d, u, f, 5, m, w, s, j}
 E = {5, 0, k, 2, 4, 8, j, p, 6, '}
 Q = {1, s, 7, q, e, 8, b, 2, 4}
 A = {3, h, b, q, 6, k, 4, g, t}
 N = {c, u, 1, 8, 6, 2, t, k, x}

Copyright © 2012 UCI. 3 : 11 : 10

Figura 27. Interfaz de evaluación de Diferencia

Teoría de Conjuntos

- Conjuntos
- Teoría
- Experimentación
- Evaluación
 - Intersección
 - Unión
 - Diferencia
 - Diferencia Simétrica
 - Combinados**
 - Otros
 - Reporte de notas
 - Sobre el Applet

Ejercicios Combinados

Ej 5:

a) $[E \cup Q] \setminus [S \cap J] = \{ \text{ } \}$

b) $[B \Delta [S \cup K]] \setminus K = \{ \text{ } \}$

c) $[I \cap Q \cap Q] \cup [B \setminus B] = \{ \text{ } \}$

d) $[P \Delta S] \setminus [V \Delta K] = \{ \text{ } \}$

e) $[H \Delta N] \cap [J \Delta [D \cup R]] = \{ \text{ } \}$

f) $[E \cup A \cup O] \cap [N \cup L \cup C] = \{ \text{ } \}$

Lista de Conjuntos

C = ∅
F = {5}
R = {2}
P = {2, 4, i}
E = {7, 8, u}

Ejemplo: 1,2,q,4,5,d,3,e,a

Copyright © 2012 UCI. 3 : 12 : 29

Figura 28. Interfaz de evaluación de Ejercicios Combinados

Teoría de Conjuntos

- Conjuntos
- Teoría
- Experimentación
- Evaluación
 - Intersección
 - Unión**
 - Diferencia
 - Diferencia Simétrica
 - Combinados
 - Otros
 - Reporte de notas
 - Sobre el Applet

Unión

Ej 1: **Lista de Conjuntos**

a) $B \cup P = \{ \text{ } \}$

b) $O \cup K = \{ \text{ } \}$

c) $T \cup S = \{ \text{ } \}$

d) $F \cup U \cup A = \{ \text{ } \}$

e) $M \cup [N \cup R] = \{ \text{ } \}$

f) $[L \cup V] \cup [C \cup E] = \{ \text{ } \}$

Ejemplo: 1,2,q,4,5,d,3,e,a

Copyright © 2012 UCI. 3 : 10 : 53

Figura 29. Interfaz de evaluación de Unión

Anexo 3: Interfaces de los ejercicios evaluativos del OA-TG.

Teoría de Grafos

- Teoría de Grafos
 - Teoría
 - Experimentación
 - Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz**
 - Reporte de notas
 - Sobre el Applet

Seleccione el grafo que corresponda con cada Matriz:

$Q_b = \{V(Q_b) = \{y, d, w, x, a, t, g, v, j\}, E(Q_b) = \{(v, d), (w, j), (y, d), (y, w), (y, x), (y, a), (y, t), (y, g), (y, v), (y, j), (d, w), (d, x), (d, a), (d, t), (d, g), (d, v), (d, j), (w, x), (w, a), (w, t), (w, g), (w, v), (w, j), (x, a), (x, t), (x, g), (x, v), (x, j), (a, t), (a, g), (a, v), (a, j), (t, g), (t, v), (t, j), (g, v), (g, j), (v, j)\}$
 $P_a = \{V(P_a) = \{o, x, l, t, v, w\}, E(P_a) = \{(o, t), (x, x), (l, w), (o, l), (o, x), (o, v), (o, w), (x, l), (x, t), (x, v), (x, w), (l, t), (l, v), (l, w), (t, v), (t, w), (v, w)\}$
 $I_e = \{V(I_e) = \{j, x, f, r, i, e, u, m\}, E(I_e) = \{(x, e), (i, f), (m, f), (j, x), (f, r), (i, e), (u, m)\}$

a)	y	d	w	x	a	t	g	v	j	b)	k	n	d	g	p	c	m	o	z	c)	o	x	l	t	v	w
y	0	1	1	1	1	1	1	1	1	k	0	0	1	0	1	0	1	1	0	o	0	1	1	1	1	1
d	1	0	1	1	1	1	1	1	1	n	0	0	1	0	1	0	1	1	1	x	1	2	1	1	1	1
w	1	1	0	1	1	1	1	1	1	d	1	1	0	1	0	0	0	1	1	l	1	1	0	1	1	1
x	1	1	1	2	1	1	1	1	1	g	0	0	1	0	0	0	0	0	0	t	1	1	1	0	1	1
a	1	1	1	1	0	1	1	1	1	p	1	1	0	0	2	1	1	1	1	v	1	1	1	1	0	1
t	1	1	1	1	1	0	1	1	1	c	0	0	0	0	1	0	0	0	1	w	1	1	1	1	1	0
g	1	1	1	1	1	1	0	1	1	m	1	1	0	0	1	0	0	1	1							
v	1	1	1	1	1	1	1	0	1	o	1	1	1	0	1	0	1	0	0							
j	1	1	1	1	1	1	1	1	0	z	0	1	1	0	1	1	1	0	0							

a) b) c)

Copyright © 2012 UCI. 3 : 17 : 23

Figura 30. Interfaz de evaluación de Matriz de adyacencia

Teoría de Grafos

- Teoría de Grafos
 - Teoría
 - Experimentación
 - Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo**
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Q_b = \{V(Q_b) = \{y, d, w, x, a, t, g, v, j\}, E(Q_b) = \{(v, d), (w, j), (y, d), (y, w), (y, x), (y, a), (y, t), (y, g), (y, v), (y, j), (d, w), (d, x), (d, a), (d, t), (d, g), (d, v), (d, j), (w, x), (w, a), (w, t), (w, g), (w, v), (w, j), (x, a), (x, t), (x, g), (x, v), (x, j), (a, t), (a, g), (a, v), (a, j), (t, g), (t, v), (t, j), (g, v), (g, j), (v, j)\}$
 $P_a = \{V(P_a) = \{o, x, l, t, v, w\}, E(P_a) = \{(o, t), (x, x), (l, w), (o, l), (o, x), (o, v), (o, w), (x, l), (x, t), (x, v), (x, w), (l, t), (l, v), (l, w), (t, v), (t, w), (v, w)\}$
 $I_e = \{V(I_e) = \{j, x, f, r, i, e, u, m\}, E(I_e) = \{(x, e), (i, f), (m, f), (j, x), (f, r), (i, e), (u, m)\}$
 $Z_c = \{V(Z_c) = \{w, p, q, t, y\}, E(Z_c) = \{(w, t), (w, y), (q, t), (t, y), (p, q), (p, t), (p, y)\}$
 $Y = \{V(Y) = \{k, n, d, g, p, c, m, o, z\}, E(Y) = \{(p, m), (k, m), (l, m), (n, d), (n, g), (n, o), (n, z), (d, g), (d, o), (d, z), (g, o), (g, z), (o, z)\}$
 $R_g = \{V(R_g) = \{f, v, c, s, l, i, d, z\}, E(R_g) = \{(i, s), (l, l), (d, s), (f, v), (f, c), (f, s), (v, c), (v, s), (c, s), (s, l), (s, i), (s, d), (l, d), (l, z), (i, z), (d, z)\}$

1- Seleccione en la siguiente lista los grafos que sean Completos:

a) <input type="checkbox"/> Qb	f) <input type="checkbox"/> Bg
b) <input type="checkbox"/> Pa	g) <input type="checkbox"/> Ka
c) <input type="checkbox"/> Ie	h) <input type="checkbox"/> Ta
d) <input type="checkbox"/> Zc	i) <input type="checkbox"/> Pb
e) <input type="checkbox"/> Y	j) <input type="checkbox"/> N

Copyright © 2012 UCI. 3 : 16 : 19

Figura 31. Interfaz de evaluación Grafo Completo

Teoría de Grafos

- Teoría de Grafos
 - Teoría
 - Experimentación
 - Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico**
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Seleccione el grafo que corresponda con cada Gráfico:

$Qb = \{ V (Qb) = \{ y, d, w, x, a, t, g, v, j \}, E (Qb) = \{ (v, d), (w, j),$
 $Pa = \{ V (Pa) = \{ o, x, l, t, v, w \}, E (Pa) = \{ (o, t), (x, x), (l, w),$
 $le = \{ V (le) = \{ j, x, f, r, i, e, u, m \}, E (le) = \{ (x, e), (i, f), (m, f),$

a)

b)

c)

Seleccione el grafo que corresponda con la imagen:

a)

b)

c)

Figura 32. Interfaz de evaluación de Representación gráfica

Teoría de Grafos

- Teoría de Grafos
 - Teoría
 - Experimentación
 - Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices**
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Og = \{ V (Og) = \{ w, q, l, h, g, t, x, z, j, v \}, E (Og) = \{ (l, x), (w, l),$
 $Lg = \{ V (Lg) = \{ h, z, l, s, t, k, r, u \}, E (Lg) = \{ (l, k), (r, z), (k, u)$
 $Nc = \{ V (Nc) = \{ f, e, n, w, t \}, E (Nc) = \{ (w, f), (f, f), (t, t), (t, w)$
 $Ve = \{ V (Ve) = \{ n, b, y, a, l, g, q, t \}, E (Ve) = \{ (a, b), (q, l), (l, t)$
 $Bb = \{ V (Bb) = \{ f, j, c, y, i, l, s, o, g \}, E (Bb) = \{ (g, g), (i, c), (j,$
 $Cb = \{ V (Cb) = \{ u, n, a, x, o, b, t, d, k \}, E (Cb) = \{ (a, k), (x, t),$

a) ¿ Cuál es el grado del vértice en el grafo ?

b) ¿ Cuál es el grado del vértice en el grafo ?

c) Seleccione un vértice de grado en el grafo

d) Seleccione un vértice de grado en el grafo

e) El vértice es de grado en el grafo Si No

Copyright © 2012 UCI. 3 : 15 : 29

Figura 33. Interfaz de evaluación de Vértices colgantes

Teoría de Grafos

- Teoría de Grafos
- Teoría
- Experimentación
- Evaluación
 - Grafo Simple**
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Og = \{V(Og) = \{w, q, l, h, g, t, x, z, j, v\}, E(Og) = \{(l, x), (w, l)\}$
 $Lg = \{V(Lg) = \{h, z, l, s, t, k, r, u\}, E(Lg) = \{(l, k), (r, z), (k, u)\}$
 $Nc = \{V(Nc) = \{f, e, n, w, t\}, E(Nc) = \{(w, f), (f, f), (t, t), (t, w)\}$
 $Ve = \{V(Ve) = \{n, b, y, a, l, g, q, t\}, E(Ve) = \{(a, b), (q, l), (l, t)\}$
 $Bb = \{V(Bb) = \{f, j, c, y, i, l, s, o, g\}, E(Bb) = \{(g, g), (i, c), (j, c)\}$
 $Ch = \{V(Ch) = \{u, n, a, v, o, h, t, d, k\}, E(Ch) = \{(a, k), (x, t)\}$

1- Seleccione en la siguiente lista los grafos que sean Simples:

a) <input type="checkbox"/> Og	f) <input type="checkbox"/> Cb
b) <input type="checkbox"/> Lg	g) <input type="checkbox"/> E
c) <input type="checkbox"/> Nc	h) <input type="checkbox"/> Le
d) <input type="checkbox"/> Ve	i) <input type="checkbox"/> Qa
e) <input type="checkbox"/> Bb	j) <input type="checkbox"/> Zb

Copyright © 2012 UCI. 3 : 14 : 57

Figura 34. Interfaz de evaluación de Grafos simples

Teoría de Grafos

- Teoría de Grafos
- Teoría
- Experimentación
- Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple**
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Qb = \{V(Qb) = \{y, d, w, x, a, t, g, v, j\}, E(Qb) = \{(v, d), (w, j)\}$
 $Pa = \{V(Pa) = \{o, x, l, t, v, w\}, E(Pa) = \{(o, t), (x, x), (l, w), (l, v)\}$
 $le = \{V(le) = \{j, x, f, r, i, e, u, m\}, E(le) = \{(x, e), (i, f), (m, f)\}$
 $Zc = \{V(Zc) = \{w, p, q, t, y\}, E(Zc) = \{(w, t), (w, y), (q, t), (t, y)\}$
 $Y = \{V(Y) = \{k, n, d, g, p, c, m, o, z\}, E(Y) = \{(p, m), (k, m), (l, o)\}$

a) Escriba un camino del vértice al vértice en el grafo :

b) ¿ Es la siguiente sucesión de vértices un camino de a en el grafo ? Si No

c) ¿Cuál es la longitud del camino en el grafo ?

Copyright © 2012 UCI. 3 : 16 : 34

Figura 35. Interfaz de evaluación de Camino simple

Teoría de Grafos

- Teoría de Grafos
- Teoría
- Experimentación
- Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes**
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Og = \{V(Og) = \{w, q, l, h, g, t, x, z, j, v\}, E(Og) = \{(l, x), (w, l), (w, q), (w, t), (w, z), (w, j), (w, v), (q, l), (q, t), (q, z), (q, j), (q, v), (l, h), (l, g), (l, t), (l, x), (l, z), (l, j), (l, v), (h, g), (h, t), (h, x), (h, z), (h, j), (h, v), (g, t), (g, x), (g, z), (g, j), (g, v), (t, x), (t, z), (t, j), (t, v), (x, z), (x, j), (x, v), (z, j), (z, v), (j, v)\}$

$Lg = \{V(Lg) = \{h, z, l, s, t, k, r, u\}, E(Lg) = \{(l, k), (r, z), (k, u), (h, z), (h, l), (h, s), (h, t), (h, k), (h, r), (h, u), (z, l), (z, s), (z, t), (z, k), (z, r), (z, u), (l, s), (l, t), (l, k), (l, r), (l, u), (s, t), (s, k), (s, r), (s, u), (t, k), (t, r), (t, u), (k, r), (k, u), (r, u)\}$

$Nc = \{V(Nc) = \{f, e, n, w, t\}, E(Nc) = \{(w, f), (f, f), (t, t), (t, w), (f, e), (f, n), (f, t), (e, n), (e, w), (e, t), (n, w), (n, t), (w, t)\}$

$Ve = \{V(Ve) = \{n, b, y, a, l, g, q, t\}, E(Ve) = \{(a, b), (q, l), (l, t), (n, b), (n, y), (n, a), (n, l), (n, g), (n, q), (n, t), (b, y), (b, a), (b, l), (y, a), (y, l), (y, g), (y, q), (y, t), (a, l), (a, g), (a, q), (a, t), (l, g), (l, q), (l, t), (g, q), (g, t), (q, t)\}$

$Bb = \{V(Bb) = \{f, j, c, y, i, l, s, o, g\}, E(Bb) = \{(g, g), (i, c), (j, c), (f, j), (f, c), (j, o), (j, s), (j, g), (c, y), (c, i), (c, l), (c, s), (c, o), (c, g), (y, i), (y, l), (y, s), (y, o), (y, g), (i, l), (i, s), (i, o), (i, g), (l, s), (l, o), (l, g), (s, o), (s, g), (o, g)\}$

$Cb = \{V(Cb) = \{u, n, a, x, o, b, t, d, k\}, E(Cb) = \{(a, k), (x, t), (u, n), (u, a), (u, x), (u, o), (u, b), (u, t), (u, d), (u, k), (n, a), (n, x), (n, o), (n, b), (n, t), (n, d), (n, k), (a, x), (a, o), (a, b), (a, t), (a, d), (a, k), (x, o), (x, b), (x, t), (x, d), (x, k), (o, b), (o, t), (o, d), (o, k), (b, t), (b, d), (b, k), (t, d), (t, k), (d, k)\}$

a) Tiene el grafo vértices colgantes ? Si No

b) Seleccione un vértice colgante en el grafo

c) Seleccione un vértice colgante en el grafo

d) En el grafo el vértice es colgante ? Si No

e) En el grafo el vértice es colgante ? Si No

Copyright © 2012 UCI. 3 : 15 : 58

Figura 36. Interfaz de evaluación de Vértices colgantes

Teoría de Grafos

- Teoría de Grafos
- Teoría
- Experimentación
- Evaluación
 - Grafo Simple
 - Vértices adyacentes
 - Grado de vértices
 - Vértices aislados**
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Na = \{V(Na) = \{s, y, g, k, z, f, r, t, m, w\}, E(Na) = \{(g, w), (y, k), (s, y), (s, g), (s, k), (s, z), (s, f), (s, r), (s, t), (s, m), (s, w), (y, g), (y, k), (y, z), (y, f), (y, r), (y, t), (y, m), (y, w), (g, k), (g, z), (g, f), (g, r), (g, t), (g, m), (g, w), (k, z), (k, f), (k, r), (k, t), (k, m), (k, w), (z, f), (z, r), (z, t), (z, m), (z, w), (f, r), (f, t), (f, m), (f, w), (r, t), (r, m), (r, w), (t, m), (t, w), (m, w)\}$

$Cg = \{V(Cg) = \{w, x, u, g, b, v, p, f, z, k\}, E(Cg) = \{(w, w), (k, w), (w, x), (w, u), (w, g), (w, b), (w, v), (w, p), (w, f), (w, z), (w, k), (x, u), (x, g), (x, b), (x, v), (x, p), (x, f), (x, z), (x, k), (u, g), (u, b), (u, v), (u, p), (u, f), (u, z), (u, k), (g, b), (g, v), (g, p), (g, f), (g, z), (g, k), (b, v), (b, p), (b, f), (b, z), (b, k), (v, p), (v, f), (v, z), (v, k), (p, f), (p, z), (p, k), (f, z), (f, k), (z, k)\}$

$Da = \{V(Da) = \{q, y, f, g, w, b, j, v\}, E(Da) = \{(q, w), (y, v), (g, w), (q, y), (q, f), (q, g), (q, b), (q, j), (q, v), (y, f), (y, g), (y, b), (y, j), (y, v), (f, g), (f, b), (f, j), (f, v), (g, b), (g, j), (g, v), (b, j), (b, v), (j, v)\}$

$X = \{V(X) = \{n, c, u, m, f, h, b, v, q\}, E(X) = \{(f, c), (q, h), (f, l), (n, c), (n, u), (n, m), (n, f), (n, h), (n, b), (n, v), (n, q), (c, u), (c, m), (c, f), (c, h), (c, b), (c, v), (c, q), (u, m), (u, f), (u, h), (u, b), (u, v), (u, q), (m, f), (m, h), (m, b), (m, v), (m, q), (f, h), (f, b), (f, v), (h, b), (h, v), (h, q), (b, v), (b, q), (v, q)\}$

$Ec = \{V(Ec) = \{h, e, v, a, t, i, q, s, d\}, E(Ec) = \{(e, i), (s, i), (q, h), (h, e), (h, v), (h, a), (h, t), (h, i), (h, q), (h, s), (h, d), (e, v), (e, a), (e, t), (e, i), (e, q), (e, s), (e, d), (v, a), (v, t), (v, i), (v, q), (v, s), (v, d), (a, t), (a, i), (a, q), (a, s), (a, d), (t, i), (t, q), (t, s), (t, d), (i, q), (i, s), (i, d), (q, s), (q, d), (s, d)\}$

$Ke = \{V(Ke) = \{n, l, a, m, r\}, E(Ke) = \{(r, l), (n, l)\}$

a) Tiene el grafo vértices aislados ? Si No

b) Seleccione un vértice aislado en el grafo

c) Seleccione un vértice aislado en el grafo

d) En el grafo el vértice es aislado ? Si No

e) En el grafo el vértice es aislado ? Si No

Copyright © 2012 UCI. 3 : 15 : 44

Figura 37. Interfaz de evaluación de Vértices aislados

Teoría de Grafos

- Teoría de Grafos
 - Teoría
 - Experimentación
 - Evaluación
 - Grafo Simple
 - Vértices adyacentes**
 - Grado de vértices
 - Vértices aislados
 - Vértices colgantes
 - Grafo completo
 - Camino simple
 - Gráfico
 - Matriz
 - Reporte de notas
 - Sobre el Applet

Grafos

$Og = \{ V (Og) = \{ w, q, l, h, g, t, x, z, j, v \}, E (Og) = \{ (l, x), (w, l) \}$
 $Lg = \{ V (Lg) = \{ h, z, l, s, t, k, r, u \}, E (Lg) = \{ (l, k), (r, z), (k, u) \}$
 $Nc = \{ V (Nc) = \{ f, e, n, w, t \}, E (Nc) = \{ (w, f), (f, f), (t, t), (t, w) \}$
 $Ve = \{ V (Ve) = \{ n, b, y, a, l, g, q, t \}, E (Ve) = \{ (a, b), (q, l), (l, t) \}$

a) En el grafo los vértices y son adyacentes.
 Si No

b) En el grafo los vértices y son adyacentes.
 Si No

c) Seleccione dos vértices adyacentes del grafo

d) Seleccione dos vértices adyacentes del grafo

e) Seleccione dos vértices adyacentes del grafo

Copyright © 2012 UCI.

3 : 15 : 14

Figura 38. Interfaz de evaluación de Vértices adyacentes

Anexo 4: Historias de Usuario de los OA.

Historias de Usuario del OA-TC

Historia de Usuario	
Número: HUTC1	Usuario: Estudiante, Profesor
Nombre Historia: Calcular intersección	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite calcular la intersección entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Intersección” en el menú del área de experimentación y se seleccionan los conjuntos para realizar la operación. Para proceder a mostrar el resultado de la operación existen dos variantes. Una primera variante es luego de haber seleccionado los conjuntos, proceder a mostrar directamente el resultado que ofrece el OA. En la segunda variante es el usuario quien introduce la respuesta al ejercicio. En ambas variantes el resultado es mostrado mediante diagramas de Venn.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Para la segunda variante en caso de que la respuesta introducida por el usuario no tenga el formato correcto o deje el campo en blanco se muestra un mensaje “Formato de la respuesta incorrecto”. 	

Tabla 16. HU Calcular intersección

Historia de Usuario	
Número: HUTC2	Usuario: Estudiante, Profesor
Nombre Historia: Calcular unión	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite calcular la unión entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Unión” en el menú del área de experimentación y se seleccionan los conjuntos para realizar la operación. Para proceder a mostrar el resultado de la operación existen dos variantes. Una primera variante es luego de haber seleccionado los conjuntos, proceder a mostrar directamente el resultado que ofrece el OA. En la segunda variante es el usuario quien introduce la respuesta al ejercicio. En ambas variantes el resultado es mostrado mediante diagramas de Venn.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 2. Para la segunda variante en caso de que la respuesta introducida por el usuario no tenga el formato correcto o deje el campo en blanco se muestra un mensaje “Formato de la respuesta incorrecto”. 	

Tabla 17. HU Calcular Unión

Historia de Usuario	
Número: HUTC3	Usuario: Estudiante, Profesor
Nombre Historia: Calcular Diferencia	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite calcular la diferencia entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Diferencia” en el menú del área de experimentación, se seleccionan los conjuntos para realizar la operación. Para proceder a mostrar el resultado de la operación existen dos variantes. Una primera variante es luego de haber seleccionado los conjuntos, proceder a mostrar directamente el resultado que ofrece el OA. En la segunda variante es el usuario quien introduce la respuesta al ejercicio. En ambas variantes el resultado es mostrado mediante diagramas de Venn.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Para la segunda variante en caso de que la respuesta introducida por el usuario no tenga el formato correcto o deje el campo en blanco se muestra un mensaje “Formato de la respuesta incorrecto”. 	

Tabla 18. HU Calcular diferencia

Historia de Usuario	
Número: HUTC4	Usuario: Estudiante, Profesor
Nombre Historia: Calcular Diferencia Simétrica	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite calcular la diferencia simétrica entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Diferencia Simétrica” en el menú del área de experimentación, se seleccionan los conjuntos para realizar la operación. Para proceder a mostrar el resultado de la operación existen dos variantes. Una primera variante es luego de haber seleccionado los conjuntos, proceder a mostrar directamente el resultado que ofrece el OA. En la segunda variante es el usuario quien introduce la respuesta al ejercicio. En ambas variantes el resultado es mostrado mediante diagramas de Venn.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Para la segunda variante en caso de que la respuesta introducida por el usuario no tenga el formato correcto o deje el campo en blanco se muestra un mensaje “Formato de la respuesta incorrecto”. 	

Tabla 19. HU Calcular diferencia simétrica

Historia de Usuario	
Número: HUTC5	Usuario: Estudiante, Profesor
Nombre Historia: Hallar Conjunto Potencia	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite hallar el conjunto potencia de un conjunto. Para realizar la operación el usuario selecciona la opción “Conjunto Potencia” en el menú del área de experimentación, se selecciona el conjunto para realizar la operación y se procede a mostrar el resultado mediante su forma extensional.	
Observaciones:	

Tabla 20. HU Calcular conjunto potencia

Historia de Usuario	
Número: HUTC6	Usuario: Estudiante, Profesor
Nombre Historia: Calcular Producto Cartesiano	
Referencia: FOAC1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite calcular el producto cartesiano entre dos conjuntos. Para realizar la operación el usuario selecciona la opción “Producto Cartesiano” en el menú del área de experimentación, se seleccionan los conjuntos para realizar la operación y se procede a mostrar el resultado mediante su forma extensional.	
Observaciones:	

Tabla 21. HU Calcular producto cartesiano

Historia de Usuario	
Número: HUTC7	Usuario: Estudiante, Profesor
Nombre Historia: Evaluar contenido TC	
Referencia: FOAC2	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite evaluar al usuario sobre cada una de las operaciones entre conjuntos disponibles en el panel de experimentación. Para realizar la evaluación se selecciona el contenido a evaluar en el menú del panel de evaluación, seguidamente se muestra una interfaz donde aparecen diferentes ejercicios evaluativos referentes al contenido seleccionado, se cuenta además con un listado de conjuntos que pueden ser seleccionados para realizar	

operaciones. Se brinda la posibilidad de generar nuevos conjuntos de manera automática. Una vez entrados los datos requeridos en las evaluaciones se procede a confirmar la misma mediante un botón identificado como “OK”.
Observaciones: <ol style="list-style-type: none"> 1. En caso de que existan errores en la entrada de datos se muestra un mensaje informativo “Formato de la respuesta incorrecto”, indicando el o los encisos que presentan error. 2. En caso de que el usuario no especifique algunos de los campos se le muestra un mensaje informativo “Debe llenar todos los campos”.

Tabla 22. HU Evaluar contenido TC

Historia de Usuario	
Número: HUTC8	Usuario: Estudiante, Profesor
Nombre Historia: Graficar conjunto	
Referencia: FOAC3	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Baja
Descripción: La historia de usuario permite mostrar gráficamente los conjuntos. Para realizar esta operación el usuario selecciona la opción “Gestionar Conjunto” en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de conjuntos creados el conjunto que se desea mostrar. Una vez indicado el conjunto se selecciona la opción “Mostrar Conjunto” y el conjunto es graficado mediante diagramas de Venn.	
Observaciones:	

Tabla 23. HU Graficar conjunto

Historia de Usuario	
Número: HUTC9	Usuario: Estudiante, Profesor
Nombre Historia: Crear conjunto	
Referencia: FOAC4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Baja
Descripción: La historia de usuario permite crear nuevos conjuntos. Para realizar esta operación el usuario selecciona la opción “Gestionar Conjunto” en el menú del área de experimentación, se muestra una interfaz donde se indica el nombre del conjunto a crear. Una vez creado el conjunto se actualiza el listado donde se muestran los conjuntos creados.	
Observaciones:	
1. En caso de que el usuario no especifique el nombre del conjunto se le muestra un mensaje informativo “Debe especificar el nombre del conjunto”.	

Tabla 24. HU Crear conjunto

Historia de Usuario	
Número: HUTC10	Usuario: Estudiante, Profesor
Nombre Historia: Eliminar conjunto	
Referencia: FOAC4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Baja
Descripción: La historia de usuario permite eliminar conjuntos. Para realizar esta operación el usuario selecciona la opción "Gestionar Conjunto" en el menú del área de experimentación, se muestra una interfaz donde se indica el nombre del conjunto a eliminar. Una vez eliminado el conjunto se actualiza el listado donde se muestran conjuntos creados.	
Observaciones: 1. En caso de que el usuario no especifique el nombre del conjunto se le muestra un mensaje informativo "Debe especificar el nombre del conjunto".	

Tabla 25. HU Eliminar conjunto

Historia de Usuario	
Número: HUTC11	Usuario: Estudiante, Profesor
Nombre Historia: Adicionar elemento a un conjunto	
Referencia: FOAC4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Baja
Descripción: La historia de usuario permite adicionar elementos a un conjunto. Para realizar esta operación el usuario selecciona la opción "Gestionar Conjunto" en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de conjuntos creados el conjunto que se desea modificar. Una vez indicado el conjunto se especifica el símbolo del elemento a adicionar. Finalmente luego de adicionar el elemento se actualiza el listado donde se muestran los conjuntos creados.	
Observaciones: 1. En caso de que el usuario no especifique el símbolo del elemento se le muestra un mensaje informativo "Debe especificar el elemento".	

Tabla 26. HU Adicionar elemento a un conjunto

Historia de Usuario	
Número: HUTC12	Usuario: Estudiante, Profesor
Nombre Historia: Eliminar elemento de un conjunto	
Referencia: FOAC4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Baja
Descripción: La historia de usuario permite eliminar elementos de un conjunto. Para realizar esta operación el usuario selecciona la opción “Gestionar Conjunto” en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de conjuntos creados el conjunto que se desea modificar. Una vez indicado el conjunto se especifica el símbolo del elemento a eliminar. Finalmente luego de eliminar el elemento se actualiza el listado donde se muestran los conjuntos creados.	
Observaciones: 1. En caso de que el usuario no especifique el símbolo del elemento se le muestra un mensaje informativo “Debe especificar el elemento”.	

Tabla 27. HU Eliminar elemento de un conjunto

Historias de Usuario del OA-MT

Historia de Usuario	
Número: HUMT1	Usuario: Estudiante, Profesor
Nombre Historia: Experimentar con cadenas	
Referencia: FOAMT1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite a los usuarios experimentar con diferentes cintas a partir de ejemplos de Máquinas propuestas. Para realizar la operación el usuario selecciona la opción “Ejemplos propuesto” en el menú de experimentación luego se selecciona el ejemplo en una lista desplegable y seguidamente aparecerá una descripción asociada al ejemplo seleccionado, y será posible experimentar con diferentes cadenas entradas por el usuario.	
Observaciones: 1. En caso de que no se seleccione un ejemplo se mostrará un mensaje “Debe seleccionar un ejemplo.” 2. En caso de dejar el campo en blanco se muestra un mensaje “Entre la cadena a evaluar.”	

Tabla 28. HU Experimentar con cadenas

Historia de Usuario	
Número: HUMT2	Usuario: Estudiante, Profesor
Nombre Historia: Tracear cadenas.	
Referencia: FOAMT1, FOAMT2	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite a los usuarios tracear cadenas tanto de los ejemplos como de los ejercicios resueltos. Para tracear la cinta luego de analizar la cadena se selecciona la opción "Tracear cadena" en el marco de trabajo "Ejemplos propuestos" o "Ejercicios propuestos", una vez seleccionada se podrá contar con controles que permites ir moviendo el cabezal por cada símbolo de la cinta de manera progresiva o regresiva.	
Observaciones:	

Tabla 29. HU Tracear cadenas

Historia de Usuario	
Número: HUMT3	Usuario: Estudiante, Profesor
Nombre Historia: Resolver ejercicios propuestos.	
Referencia: FOAMT2	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite resolver ejercicios de Máquinas de Turing propuestas. Para realizar la operación el usuario selecciona la opción "Ejercicios propuestos" en el menú del área de experimentación, se selecciona el ejercicio a resolver en una lista de ejercicios y se muestra la orden del mismo. Una vez mostrado el ejercicio el usuario entra los datos de las funciones, del estado inicial y los estados finales. Las funciones y estados finales se adicionarán a una lista de funciones y de estados respectivamente, donde podrán ser visualizadas por el usuario. Luego de finalizar el ejercicio se comprueba la respuesta donde la aplicación informa al usuario si la solución esta correcto o no.	
Observaciones:	
<ol style="list-style-type: none"> 1. En caso de dejar algunos de los datos de las funciones en blanco se muestra un mensaje "Debe llenar todos los campos". 2. En caso de dejar el campo de estados finales en blanco se muestra un mensaje "Debe llenar el campo". 	

Tabla 30. HU Resolver ejercicios propuestos

Historia de Usuario	
Número: HUMT4	Usuario: Estudiante, Profesor
Nombre Historia: Evaluar contenido de MT.	
Referencia: FOAMT3	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite a los usuarios evaluarse sobre el contenido de MT. Para realizar la evaluación se selecciona la opción “Resolver ejercicio” en el menú de evaluación donde se podrá seleccionar el ejercicio a resolver. Una vez resuelto el ejercicio se procede a guardar la evaluación mediante un botón identificado como “Terminar”.	
Observaciones:	

Tabla 31. HU Evaluar contenido de MT

Historia de Usuario	
Número: HUMT5	Usuario: Estudiante, Profesor
Nombre Historia: Crear nuevo ejercicio	
Referencia: FOAMT4	
Programador responsable: Isael Herrera Chica	Prioridad en negocio: Media
Descripción: La historia de usuario permite a los usuarios la creación de nuevos ejercicios. Para crear nuevos ejercicios se selecciona la opción “Nuevo ejercicio” en el menú referente al panel de evaluación. Seguidamente se muestra una interfaz donde se brinda al usuario la posibilidad de definir el alfabeto y las condiciones del nuevo ejercicio. Una vez creado el ejercicio se adicionará automáticamente a la lista de ejercicios disponibles para la evaluación.	
Observaciones:	
<ol style="list-style-type: none"> 1. En el caso de que el usuario no defina el alfabeto o defina un alfabeto de un solo símbolo se muestra un mensaje “El alfabeto debe ser de al menos dos símbolos”. 2. En el caso de que el usuario seleccione una condición y deje el campo seleccionado en blanco se muestra un mensaje “Debe llenar los campos habilitados”. 3. En caso de que existan errores en la entrada de datos de las condiciones se mostrará un mensaje notificando al usuario el error. 	

Tabla 32. HU Crear nuevo ejercicio

Historias de Usuario del OA-TG

Historia de Usuario	
Número: HUTG1	Usuario: Estudiante, Profesor
Nombre Historia: Generar matriz de adyacencia	
Referencia: FOATG1	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite a los usuarios generar la matriz de adyacencia de un determinado grafo. Para realizar la operación el usuario selecciona la opción "Matriz de adyacencia" en el menú referente al panel de experimentación. Seguidamente aparecerá una interfaz donde se podrá seleccionar el grafo a visualizar. Una vez seleccionado el grafo se muestra la matriz de adyacencia referente al mismo.	
Observaciones:	

Tabla 33. HU Generar matriz de adyacencia

Historia de Usuario	
Número: HUTG2	Usuario: Estudiante, Profesor
Nombre Historia: Generar representación gráfica del grafo.	
Referencia: FOATG2	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
Descripción: La historia de usuario permite a los usuarios generar el gráfico de un determinado grafo. Para realizar la operación el usuario selecciona la opción "Representación gráfica" en el menú referente al panel de experimentación. Seguidamente aparecerá una interfaz donde se podrá seleccionar el grafo a visualizar. Una vez seleccionado el grafo, el mismo se muestra de forma gráfica. Luego de dibujado el grafo se brinda la posibilidad de modificar la imagen del grafico mediante la redistribución de los vértices del grafo en el área de dibujo.	
Observaciones:	

Tabla 34. HU Generar representación gráfica del grafo

Historia de Usuario	
Número: HUTG3	Usuario: Estudiante, Profesor
Nombre Historia: Evaluar contenido de la TG.	
Referencia: FOATG3	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Alta
<p>Descripción: La historia de usuario permite evaluar al usuario sobre algunas de las operaciones sobre grafos disponibles en el panel de experimentación. Para realizar la evaluación se selecciona el contenido a evaluar en el menú del panel de evaluación, seguidamente se muestra una interfaz donde aparecen diferentes ejercicios evaluativos referentes al contenido seleccionado, se cuenta además con un listado de grafos generados aleatoriamente sobre los cuales se realizan las operaciones. Una vez entrados los datos requeridos en las evaluaciones se procede a confirmar la misma mediante un botón identificado como “Terminar”.</p>	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. En caso de que existan errores en la entrada de datos se muestra un mensaje informativo “Formato de los datos incorrecto”, indicando el o los encisos que presentan error. 2. En caso de que el usuario no especifique algunos de los campos se le muestra un mensaje informativo “Debe llenar todos los campos”. 3. En caso de que el usuario no marque una opción requerida en algún inciso se le muestra un mensaje informativo “Debe seleccionar una opción”, indicando el o los encisos requeridos. 	

Tabla 35. HU Evaluar contenido de la TG

Historia de Usuario	
Número: HUTG4	Usuario: Estudiante, Profesor
Nombre Historia: Crear grafo	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
<p>Descripción: La historia de usuario permite crear nuevos grafos. Para realizar esta operación el usuario selecciona la opción “Gestionar grafo” en el menú del área de experimentación, se muestra una interfaz donde se indica el nombre del grafo a crear. Una vez creado el grafo se actualiza el listado donde se muestran los grafos creados.</p>	
Observaciones:	

Tabla 36. HU Crear grafo

Historia de Usuario	
Número: HUTG5	Usuario: Estudiante, Profesor
Nombre Historia: Eliminar grafo	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
Descripción: La historia de usuario permite eliminar grafos. Para realizar esta operación el usuario selecciona la opción "Gestionar grafo" en el menú del área de experimentación, aparece una interfaz donde se muestra un listado con los grafos creados donde se selecciona el grafo a eliminar. Una vez eliminado el grafo se actualiza el listado donde se muestran los grafos creados.	
Observaciones:	

Tabla 37. HU Eliminar grafo

Historia de Usuario	
Número: HUTG6	Usuario: Estudiante, Profesor
Nombre Historia: Adicionar vértice a un grafo	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
Descripción: La historia de usuario permite adicionar vértices a un grafo. Para realizar esta operación el usuario selecciona la opción "Gestionar grafo" en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de grafos creados el grafo que se desea modificar. Una vez indicado el grafo se especifica el identificador del vértice a adicionar. Finalmente luego de adicionar el vértice se actualiza el listado donde se muestran los vértices del grafo seleccionado.	
Observaciones:	

Tabla 38. HU Adicionar vértice a un grafo

Historia de Usuario	
Número: HUTG7	Usuario: Estudiante, Profesor
Nombre Historia: Eliminar vértices del grafo.	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
Descripción: La historia de usuario permite eliminar vértices de un grafo. Para realizar esta operación el usuario selecciona la opción "Gestionar grafo" en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de grafos creados el grafo que se desea modificar. Una vez indicado el grafo se especifica el vértice a eliminar seleccionándolo en la lista de vértices del grafo seleccionado. Finalmente luego de eliminar el vértice se actualiza el listado donde se muestran los vértices del grafo seleccionado.	
Observaciones:	

Tabla 39. HU Eliminar vértices del grafo.

Historia de Usuario	
Usuario: Estudiante, Profesor	
Nombre Historia: Adicionar aristas al grafo.	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
Descripción: La historia de usuario permite adicionar aristas a un grafo. Para realizar esta operación el usuario selecciona la opción "Gestionar grafo" en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de grafos creados el grafo que se desea modificar. Una vez indicado el grafo se especifican los vértices entre los cuales se desea crear la arista mediante dos listados desplegables donde se muestran los vértices del grafo seleccionado. Finalmente luego de adicionar la arista se actualiza el listado donde se muestran las aristas del grafo seleccionado.	
Observaciones:	

Tabla 40. HU Adicionar aristas al grafo.

Historia de Usuario	
Usuario: Estudiante, Profesor	
Nombre Historia: Eliminar aristas del grafo.	
Referencia: FOATG4	
Programador responsable: Isael Herrera Chica	Prioridad en Negocio: Media
Descripción: La historia de usuario permite eliminar aristas de un grafo. Para realizar esta operación el usuario selecciona la opción "Gestionar grafo" en el menú del área de experimentación, se muestra una interfaz donde se selecciona en el listado de grafos creados el grafo que se desea modificar. Una vez indicado el grafo se especifica la arista a eliminar seleccionándolo en la lista de aristas del grafo seleccionado. Finalmente luego de eliminar la arista se actualiza el listado donde se muestran las aristas del grafo seleccionado.	
Observaciones:	

Tabla 41. HU Eliminar aristas del grafo.

Anexo 5: Cronograma de Iteraciones.

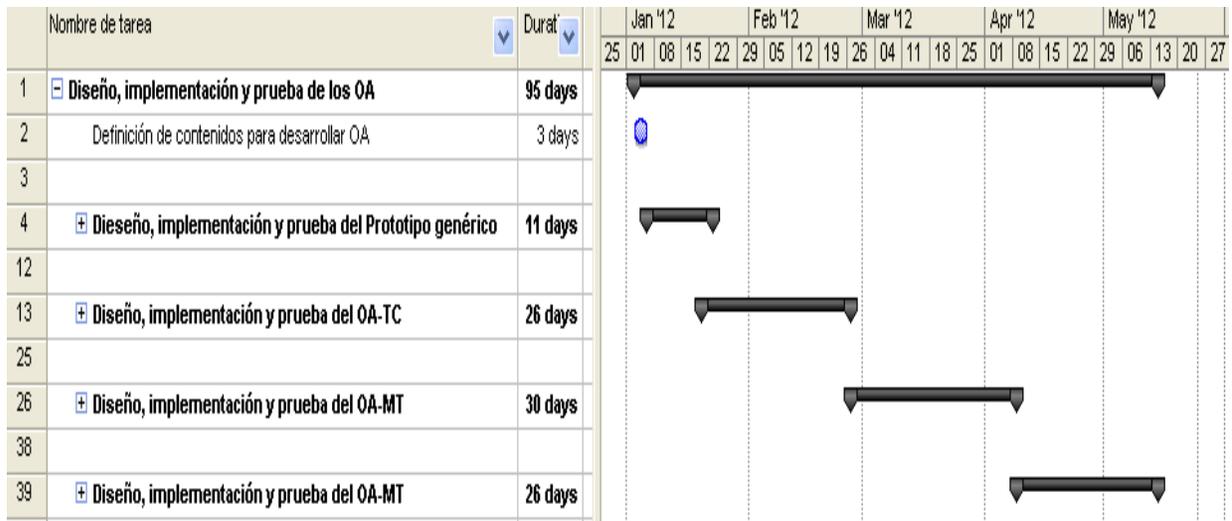


Figura 39. Diagrama de Gantt de las Iteraciones.

Anexo 6: Diseño de casos de prueba.

A continuación se muestran las descripciones de los casos de prueba diseñados para validar la calidad de los OA desarrollados desde el punto de vista de software. Al finalizar se muestra una tabla donde se muestra un resumen de la cantidad de no conformidades para cada una de las iteraciones de prueba realizadas.

Diseño de casos de prueba del OA-TC

Caso de prueba. HU Realizar operación de intersección

Descripción de la funcionalidad: En esta historia el usuario se realiza la operación de Intersección entre conjuntos. Existen dos variantes para realizar la operación. La primera consiste en mostrar directamente la respuesta propuesta por la aplicación. La segunda alternativa es que el usuario introduzca los datos de la solución. En esta última variante si los datos entrados por el usuario tienen el formato correcto estos son guardados y muestra el resultado mediante Diagramas de Venn. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Intersección”.

Escenario: “Realizar operación Intersección”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar operación de Intersección entre conjuntos (el usuario selecciona los conjuntos para realizar la operación y procede a mostrar el resultado o entra la posible respuesta y seguidamente comprueba su respuesta).	Luego de realizar la operación, si el proceso ha sido correcto, el conjunto resultante se muestra mediante diagramas de Venn.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error.		

Tabla 42. Caso de prueba HU Realizar operación de intersección

Caso de prueba. HU Realizar operación de Unión

Descripción de la funcionalidad: En esta historia el usuario se realiza la operación de Unión entre conjuntos. Existen dos variantes para realizar la operación. La primera consiste en mostrar directamente la respuesta propuesta por la aplicación. La segunda alternativa es que el usuario introduzca los datos de la solución. En esta última variante si los datos entrados por el usuario tienen el formato correcto estos son guardados y muestra el resultado mediante Diagramas de Venn. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Unión”.

Escenario: “Realizar operación Unión”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar operación de Unión entre conjuntos (el usuario selecciona los conjuntos para realizar la operación y procede a mostrar el resultado o entra la posible respuesta y seguidamente comprueba su respuesta).	Luego de realizar la operación, si el proceso ha sido correcto, el conjunto resultante se muestra mediante diagramas de Venn.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error.		

Tabla 43. Caso de prueba HU Realizar operación de Unión

Caso de prueba. HU Realizar operación de Diferencia

Descripción de la funcionalidad: En esta historia el usuario se realiza la operación de Diferencia entre conjuntos. Existen dos variantes para realizar la operación. La primera consiste en mostrar directamente la respuesta propuesta por la aplicación. La segunda alternativa es que el usuario introduzca los datos de la solución. En esta última variante si los datos entrados por el usuario tienen el formato correcto estos son guardados y muestra el resultado mediante Diagramas de Venn. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de experimentación “Diferencia”.**Escenario:** “Realizar operación Diferencia”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar operación de diferencia entre conjuntos (el usuario selecciona los conjuntos para realizar la operación y procede a mostrar el resultado o entra la posible respuesta y seguidamente comprueba su respuesta).	Luego de realizar la operación, si el proceso ha sido correcto, el conjunto resultante se muestra mediante diagramas de Venn.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error.		

Tabla 44. Caso de prueba

Caso de prueba. HU Realizar operación de Diferencia Simétrica

Descripción de la funcionalidad: En esta historia el usuario se realiza la operación de Diferencia Simétrica entre conjuntos. Existen dos variantes para realizar la operación. La primera consiste en mostrar directamente la respuesta propuesta por la aplicación. La segunda alternativa es que el usuario introduzca los datos de la solución. En esta última variante si los datos entrados por el usuario tienen el formato correcto estos son guardados y muestra el resultado mediante Diagramas de Venn. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de experimentación “Diferencia Simétrica”.**Escenario:** “Realizar operación Diferencia Simétrica”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar operación de diferencia simétrica entre conjuntos (el usuario selecciona los conjuntos para realizar la operación y procede a mostrar el resultado o entra la posible respuesta y seguidamente comprueba su respuesta).	Luego de realizar la operación, si el proceso ha sido correcto, el conjunto resultante se muestra mediante diagramas de Venn.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error.		

Tabla 45. Caso de prueba HU Realizar operación de Diferencia Simétrica

Caso de prueba. HU Evaluar contenido de TC.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Intersección. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Intersección”.

Escenario: “Realizar evaluación Intersección”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Intersección entre conjuntos (el usuario selecciona para cada inciso los conjuntos para realizar la operación y llena los campos con las posibles respuestas en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		

Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error especificando el o los encisos que contiene el error.		
--------------------------------------	---	--	--

Tabla 46. Caso de prueba HU Evaluar contenido de TC.

Caso de prueba. HU Evaluar contenido de TC.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Unión. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Unión”.

Escenario: “Realizar evaluación Unión”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Unión entre conjuntos (el usuario selecciona para cada inciso los conjuntos para realizar la operación y llena los campos con las posibles respuestas en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error especificando el o los encisos que contiene el error.		

Tabla 47. Caso de prueba HU Evaluar contenido de TC.

Caso de prueba. HU Evaluar contenido de TC.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Diferencia. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje

que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Diferencia”.

Escenario: “Realizar evaluación Diferencia”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Diferencia entre conjuntos (el usuario selecciona para cada inciso los conjuntos para realizar la operación y llena los campos con las posibles respuestas en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error especificando el o los incisos que contiene el error.		

Tabla 48. Caso de prueba HU Evaluar contenido de TC.

Caso de prueba. HU Evaluar contenido de TC.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Diferencia Simétrica. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Diferencia Simétrica”.

Escenario: “Realizar evaluación Diferencia Simétrica”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Diferencia Simétrica entre conjuntos (el usuario selecciona para cada inciso los conjuntos para realizar la operación y llena los campos con las posibles respuestas en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
Respuestas con formatos incorrectos.	Se debe mostrar un mensaje de error especificando el o los incisos que contiene el error.		

Tabla 49. Caso de prueba HU Evaluar contenido de TC.

Caso de prueba. HU Evaluar contenido de TC

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa en la operación de Producto cartesiano. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación "Otros".

Escenario: "Realizar evaluación Producto cartesiano".

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación de Producto cartesiano entre conjuntos (el usuario selecciona los conjuntos para realizar la operación y llena el campo con la posible respuesta en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		

Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		
Respuesta con formato incorrecto.	Se debe mostrar un mensaje de error especificando el error		

Tabla 50. Caso de prueba HU Evaluar contenido de TC

Caso de prueba. HU Evaluar contenido de TC

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre la habilidad de determinar el Conjunto potencia de un conjunto. Si los datos entrados por el usuario tienen el formato correcto estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Otros”.

Escenario: “Realizar evaluación Conjunto potencia”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de la operación Conjunto potencia sobre un conjuntos (el usuario selecciona el conjunto para realizar la operación y llena el campo con la posible respuesta en el formato especificado en el ejemplo).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		
Respuesta con formato incorrecto.	Se debe mostrar un mensaje de error especificando el error.		

Tabla 51. Caso de prueba HU Evaluar contenido de TC

Caso de prueba. HU Crear conjunto

Descripción de la funcionalidad: En esta historia el usuario crea un nuevo conjunto. Si se entran los datos correctamente se crea el nuevo conjunto y se actualiza la lista de conjuntos. En caso contrario se muestra un

mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar Conjunto”.

Escenario: “Crear conjunto”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Crear un nuevo conjunto (el usuario introduce el nombre del nuevo conjunto a crear (un carácter correspondiente a una letra mayúscula del alfabeto Español exceptuando la letra “ñ”), el cual se añade a la lista de conjuntos).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de conjuntos donde se muestra el nuevo conjunto vacío.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		

Tabla 52. Caso de prueba HU Crear conjunto

Caso de prueba. HU Eliminar conjunto

Descripción de la funcionalidad: En esta historia el usuario elimina un conjunto. Si se entra el nombre del conjunto correctamente se crea el nuevo conjunto y se actualiza la lista de conjuntos. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar Conjunto”.

Escenario: “Eliminar conjunto”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Eliminar un conjunto (el usuario introduce el nombre del conjunto a eliminar de los conjuntos disponibles en la lista de conjuntos).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de conjuntos de la cual se borra el conjunto eliminado.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones

Campo vacío.	Se debe mostrar un mensaje de error.		
--------------	--------------------------------------	--	--

Tabla 53. Caso de prueba HU Eliminar conjunto

Caso de prueba. HU Adicionar elementos a un conjunto

Descripción de la funcionalidad: En esta historia el usuario adiciona un elemento a un conjunto. Si se entra el nombre del elemento a adicionar correctamente se añade el nuevo elemento al conjunto especificado y se actualiza la lista de conjuntos. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar Conjunto”.

Escenario: “Adicionar elementos a un conjunto”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Adicionar elementos a un conjunto (el usuario selecciona el conjunto al cual se le desea adicionar el elementos y especifica el nombre del elemento (una letra minúscula correspondiente al alfabeto Español exceptuando la letra “ñ”). El elemento se adiciona a la lista de elementos del conjunto especificado).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de conjuntos donde se muestra el conjunto con el nuevo elemento.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		

Tabla 54. Caso de prueba HU Adicionar elementos a un conjunto

Caso de prueba. HU Eliminar elementos a un conjunto

Descripción de la funcionalidad: En esta historia el usuario elimina un elemento a un conjunto. Si se entra el nombre del elemento del elemento a eliminar correctamente se añade el nuevo elemento al conjunto especificado y se actualiza la lista de conjuntos. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar Conjunto”.

Escenario: “Eliminar elementos a un conjunto”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Eliminar elementos a un conjunto (el usuario selecciona el conjunto al cual se le desea eliminar elementos y especifica el nombre del elemento, el cual se elimina de la lista de elementos del conjunto especificado).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de conjuntos.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		

Tabla 55. Caso de prueba HU Eliminar elementos a un conjunto

Diseño de casos de prueba del OA-MT

Caso de prueba. HU Experimentar con cadenas

Descripción de la funcionalidad: En esta historia se brinda la posibilidad de experimentar con cadenas sobre un ejemplo de Máquina de Turing previamente seleccionado. Si el proceso se realiza correctamente se muestra en un listado los distintos estados por los que pasa la cinta.

Condiciones de ejecución: Seleccionar un ejemplo de Máquina de Turing.

Flujo central: Opción del menú de experimentación “Ejemplos prácticos”.

Escenario: “Experimentar con cadenas”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Experimentar con cadenas (el usuario selecciona un ejemplo de Máquina de Turing e introduce la cadena con la que desea experimentar sobre el ejemplo seleccionado).	Luego de realizar la operación, si el proceso ha sido correcto, se muestra en un listado el conjunto de estados por los que pasa la cinta al ser analizada por la Máquina de Turing seleccionada.		
Clase inválida	Resultado esperado	Resultado de la	Observaciones

		prueba	
Campo vacío.	Se debe mostrar un mensaje de error.		
No se selecciona un ejemplo.	Se debe mostrar un mensaje informando que se seleccione un ejemplo.		

Tabla 56. Caso de prueba HU Experimentar con cadenas

Caso de prueba. HU Experimentar con cadenas

Descripción de la funcionalidad: En esta historia se brinda la posibilidad de experimentar con cadenas sobre Máquina de Turing previamente resueltas. Si el proceso se realiza correctamente se muestra en un listado los distintos estados por los que pasa la cinta.

Condiciones de ejecución: Resolver la Máquina de Turing seleccionada.

Flujo central: Opción del menú de experimentación “Ejercicios propuestos”/ Opción del menú de experimentación “Resolver ejercicios”.

Escenario: “Experimentar con cadenas”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Experimentar con cadenas (el usuario selecciona el ejercicio de Máquina de Turing a resolver, seguidamente resuelve el ejercicio e introduce la cadena con la que desea experimentar sobre el ejercicio resuelto).	Luego de realizar la operación, si el proceso ha sido correcto, se muestra en un listado el conjunto de estados por los que pasa la cinta al ser analizada por la Máquina de Turing seleccionada.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
Campo vacío.	Se debe mostrar un mensaje de error.		
No se selecciona un ejercicio.	Se debe mostrar un mensaje informando que se seleccione un ejemplo.		
No se entran las funciones de transición	Se debe mostrar un mensaje informando que se deben entrar las funciones de transición.		

No se entran el o los estados finales	Se debe mostrar un mensaje informando que se deben entrar los estados finales.		
---------------------------------------	--	--	--

Tabla 57. Caso de prueba HU Experimentar con cadenas

Caso de prueba. HU Resolver ejercicio propuestos.

Descripción de la funcionalidad: En esta historia el usuario resuelve ejercicios propuestos de MT. Si el usuario selecciona un ejercicio tiene la posibilidad de entrar los datos correspondientes para la solución del mismo. Si los datos entrados son correctos son guardados como parte de la solución. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido. La entrada de datos está limitada a conveniencia del usuario. Luego de finalizar la entrada de datos se comprueba si la solución del ejercicio es correcta.

Condiciones de ejecución: Seleccionar un ejercicio propuesto.

Flujo central: Opción del menú de experimentación “Ejercicios propuestos”.

Escenario: “Resolver ejercicios propuestos”.

Clases Válidas	Resultado esperado	Resultado	Observaciones
Resolver ejercicios propuestos de MT (el usuario selecciona el ejercicio y entra los datos requeridos: funciones de transición, estado inicial y estados finales y comprueba la solución).	Luego de entrar los datos de la solución, si el proceso ha sido correcto, se muestra un mensaje al usuario indicando que los datos fueron guardados como parte de la solución.		
Clase inválida	Resultado esperado	Resultado	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
No se selecciona un ejercicio.	Se debe mostrar un mensaje informando que se seleccione un ejemplo.		
No se entran las funciones de transición	Se debe mostrar un mensaje informando que se deben entrar las funciones de transición.		
No se entran el o los estados finales	Se debe mostrar un mensaje informando que		

	se deben entrar los estados finales.		
--	--------------------------------------	--	--

Tabla 58. Caso de prueba HU Resolver ejercicio propuestos.

Caso de prueba. HU Evaluar contenido de MT.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa con ejercicios evaluativos propuestos de MT. Si el usuario selecciona un ejercicio tiene la posibilidad de entrar los datos correspondientes para la solución del mismo. Si los datos entrados son correctos se guardan como parte de la solución y se guarda la evaluación. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido. La entrada de datos está limitada a conveniencia del usuario.

Condiciones de ejecución: Seleccionar un ejercicio evaluativo.

Flujo central: Opción del menú de evaluación “Resolver ejercicios”.

Escenario: “Resolver ejercicios”.

Clases Válidas	Resultado esperado	Resultado	Observaciones
Resolver ejercicios de MT (el usuario selecciona el ejercicio y entra los datos requeridos: funciones de transición, estado inicial y estados finales y comprueba la solución).	Luego de entrar los datos de la solución, si el proceso ha sido correcto, se muestra un mensaje al usuario indicando que los datos fueron guardados como parte de la solución y se guarda la evaluación.		
Clase inválida	Resultado esperado	Resultado	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
No se selecciona un ejercicio.	Se debe mostrar un mensaje informando que se seleccione un ejemplo.		
No se entran las funciones de transición	Se debe mostrar un mensaje informando que se deben entrar las funciones de transición.		
No se entran el o los estados finales	Se debe mostrar un mensaje informando que se deben entrar los estados finales.		

Tabla 59. Caso de prueba HU Evaluar contenido de MT.

Caso de prueba. HU Crear nuevo ejercicio de MT.

Descripción de la funcionalidad: En esta historia el usuario crea nuevos ejercicios de Máquinas de Turing, la cual se añade a la lista de ejercicios evaluativos. Se brinda la posibilidad de definir el alfabeto y una serie de condiciones para la creación de la nueva Máquina. Si los datos entrados son correctos el ejercicio se crea correctamente. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Nuevo ejercicio”.

Escenario: “Nuevo ejercicio”.

Clases Válidas	Resultado esperado	Resultado	Observaciones
Crear nuevo ejercicio de MT (el usuario define el alfabeto y las condiciones que se desea que cumpla la nueva Máquina de Turing).	Luego de entrar los datos si el proceso ha sido correcto, se muestra un mensaje al usuario indicando que los datos fueron guardados como parte del nuevo ejercicio.		
Clase inválida	Resultado esperado	Resultado	Observaciones
Campos vacíos.	Se debe mostrar un mensaje de error.		
El alfabeto no posee un mínimo de 2 caracteres.	Se debe mostrar un mensaje informando el error.		
No se selecciona al menos una condición.	Se debe mostrar un mensaje informando el error.		
Existen campos en contradicción.	Se debe mostrar un mensaje informando el error.		

Tabla 60. Caso de prueba HU Crear nuevo ejercicio de MT.

Diseño de casos de prueba del OA-TG

Caso de prueba. HU Evaluar contenido de TG.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre sus .habilidades para identificar vértices adyacentes en un grafo. Si los datos de la evaluaciones son entrados correctamente estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló con éxito. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de evaluación “Vértices Adyacentes”.**Escenario:** “Realizar evaluación Vértices Adyacentes”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de vértices adyacentes (el usuario selecciona para cada inciso las respuestas, en el caso de los incisos de selección simple, debe seleccionar una opción).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
No se selecciona una opción en los incisos de selección simple	Se debe mostrar un mensaje de error.		

Tabla 61. Caso de prueba HU Evaluar contenido de TG.

Caso de prueba. HU Evaluar contenido de TG.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre sus .habilidades para determinar el grado de los vértices de en un grafo. Si los datos de la evaluaciones son entrados correctamente estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló con éxito. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de evaluación “Grado de un vértice”.**Escenario:** “Realizar evaluación Grado de vértices”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación el grado de los vértices de en un grafo (el usuario selecciona para cada inciso las respuestas, en el caso de los incisos de selección simple, debe seleccionar una opción y llenar los campos de texto en los incisos que lo requieran).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		

Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
No se seleccione una opción en los encisos de selección simple	Se debe mostrar un mensaje de error.		
Campos vacíos	Se debe mostrar un mensaje de error.		

Tabla 62. Caso de prueba HU Evaluar contenido de TG.

Caso de prueba. HU Evaluar contenido de TG.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre sus .habilidades para determinar si un grafo posee vértices aislados. Si los datos de la evaluaciones son entrados correctamente estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló con éxito. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:

Flujo central: Opción del menú de evaluación “Vértice aislado”.

Escenario: “Realizar evaluación Vértice aislado”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de Vértice aislado (el usuario selecciona para cada inciso las respuestas, en el caso de los encisos de selección simple, debe seleccionar una opción).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
No se seleccione una opción en los encisos de selección simple	Se debe mostrar un mensaje de error.		

Tabla 63. Caso de prueba HU Evaluar contenido de TG.

Caso de prueba. HU Evaluar contenido de TG.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre sus .habilidades para identificar los vértices colgantes en un grafo. Si los datos de la evaluaciones son entrados correctamente estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló con éxito. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de evaluación “Vértice colgante”.**Escenario:** “Realizar evaluación Vértice colgante”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación de Vértice colgante (el usuario selecciona para cada enciso las respuestas, en el caso de los encisos de selección simple, debe seleccionar una opción).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones
No se selecciona una opción en los encisos de selección simple	Se debe mostrar un mensaje de error.		

Tabla 64. Caso de prueba HU Evaluar contenido de TG.

Caso de prueba. HU Evaluar contenido de TG.

Descripción de la funcionalidad: En esta historia el usuario se auto evalúa sobre sus .habilidades para determinar caminos simples en un grafo. Si los datos de la evaluaciones son entrados correctamente estos son guardados y se muestra un mensaje que haga saber al usuario que la acción se desarrolló con éxito. En caso contrario se muestra un mensaje al usuario advirtiéndole del error cometido.

Condiciones de ejecución:**Flujo central:** Opción del menú de evaluación “Camino simple”.**Escenario:** “Realizar evaluación Camino simple”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Realizar evaluación sobre camino simple en un grafo (el usuario selecciona para cada enciso las respuestas, en el caso de los encisos de selección simple, debe seleccionar una opción y llenar los campos de texto en los encisos que lo requieran).	Luego de realizar la evaluación, si el proceso ha sido correcto, la evaluación es guardada y se le muestra un mensaje al usuario indicando que la evaluación fue realizada con éxito.		
Clase inválida	Resultado esperado	Resultado de la	Observaciones

		prueba	
No se seleccione una opción en los encisos de selección simple	Se debe mostrar un mensaje de error.		
Campos vacíos	Se debe mostrar un mensaje de error.		

Tabla 65. Caso de prueba HU Evaluar contenido de TG.

Caso de prueba. HU Crear grafo

Descripción de la funcionalidad: En esta historia el usuario crea nuevos grafos. Si se entran los datos correctamente se crea el nuevo grafo y se actualiza la lista de grafos. En caso contrario la aplicación no acepta los datos y no realiza ninguna acción.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Crear grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Crear un nuevo grafo (el usuario introduce el nombre del nuevo grafo a crear (una cadena de cuatro caracteres con letra inicial mayúscula correspondientes al alfabeto Español exceptuando la letra “ñ”), el cual se añade a la lista de grafos).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de grafos donde se muestra el nuevo grafo creado.		
Clase inválida	Resultado esperado	Resultado de la prueba	Observaciones

Tabla 66. Caso de prueba HU Crear grafo

Caso de prueba. HU Eliminar grafo

Descripción de la funcionalidad: En esta historia el usuario elimina un nuevo grafo. Se selecciona el grafo a eliminar en la lista de grafos y seguidamente se elimina.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Eliminar grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Eliminar un grafo (el usuario selecciona un grafo de los disponibles en la lista de grafos y se procede a eliminarlo).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de grafos donde se borra el grafo eliminado.		

Tabla 67. Caso de prueba HU Eliminar grafo

Caso de prueba. HU Adicionar vértice a un grafo

Descripción de la funcionalidad: En esta historia el usuario adiciona un vértice a un grafo. Si se entra el nombre del vértice a adicionar correctamente se añade el nuevo vértice al grafo especificado y se actualiza la lista de vértices del propio grafo. En caso contrario la aplicación no acepta los datos y no realiza ninguna acción.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Adicionar vértice a un grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Adicionar vértice a un grafo (el usuario selecciona el grafo al cual se le desea adicionar el vértice y especifica el nombre del vértice a adicionar (una letra minúscula correspondiente al alfabeto Español), el cual se adiciona a la lista de vértices del grafo especificado).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de vértices donde se muestra el nuevo vértice.		

Tabla 68. Caso de prueba HU Adicionar vértice a un grafo

Caso de prueba. HU Eliminar vértice de un grafo

Descripción de la funcionalidad: En esta historia el usuario eliminar un vértice de un grafo. Se selecciona el vértice a eliminar en la lista de vértices y seguidamente se elimina.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Eliminar vértice de un grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Eliminar vértice de un grafo (el usuario selecciona el grafo al cual se le desea eliminar el vértice y selecciona el vértice a eliminar correspondiente al grafo seleccionado).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de vértices donde se borra el vértice eliminado.		

Tabla 69. Caso de prueba HU Eliminar vértice de un grafo

Caso de prueba. HU Adicionar arista a un grafo

Descripción de la funcionalidad: En esta historia el usuario adicionar una arista a un grafo. Se selecciona el nombre de los vértices que conformaran la arista a adicionar y se añade la nueva arista al grafo especificado y se actualiza la lista de aristas del propio grafo

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Adicionar arista a un grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Adicionar arista a un grafo (el usuario selecciona el grafo al cual se le desea adicionar la arista y especifica el nombre de los vértices que formarán la nueva arista).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de aristas donde se muestra la nueva arista		

Tabla 70. Caso de prueba HU Adicionar arista a un grafo

Caso de prueba. HU Eliminar arista de un grafo

Descripción de la funcionalidad: En esta historia el usuario eliminar una arista de un grafo. Se selecciona la arista a eliminar en la lista de aristas y seguidamente se elimina.

Condiciones de ejecución:

Flujo central: Opción del menú de experimentación “Gestionar grafo”.

Escenario: “Eliminar arista de un grafo”.

Clases Válidas	Resultado esperado	Resultado de la prueba	Observaciones
Eliminar arista de un grafo (el usuario selecciona el grafo al cual se le desea eliminar la arista y selecciona la arista que desea eliminar correspondiente al grafo seleccionado).	Luego de realizar la operación, si el proceso ha sido correcto, se actualiza la lista de aristas donde se borra la arista eliminada.		

Tabla 71. Caso de prueba HU Eliminar arista de un grafo

Resumen de No Conformidades por Iteraciones de pruebas.

Nombre de Historia de usuario	Iteración 1	Iteración 2	Iteración 3
Navegar por la información	1	0	0
Realizar operación de intersección	3	1	0
Realizar Operación de Unión	2	0	0
Realizar Operación de Diferencia	2	1	0
Realizar Operación de Diferencia Simétrica	3	2	0
Realizar Operación de Producto cartesiano	2	0	0
Realizar Operación de Conjunto potencia	3	2	0
Evaluar contenido de TC	5	2	0
Experimentar con cadenas	1	0	0
Resolver ejercicios propuestos	1	0	0
Evaluar contenido de MT	1	0	0
Crear nuevo ejercicio de MT	2	1	0
Generar representación gráfica del grafos	1	0	0
Adicionar aristas al grafo	1	0	0
Evaluar contenido de TG	4	2	0

Tabla 72. Resumen de No conformidades por Iteraciones

Anexo 7: Evaluaciones de los expertos.

Evaluación de los expertos para el OA-TC

No	Aspecto formativo	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Presentación y explicación del tema a trata	3	3	3	3	3	3
2	Estructuración lógica de los contenidos	3	3	3	3	3	3
3	Exhortación del desarrollo de habilidades y competencias al estudiante	2	2	2	3	2	3
4	Reflexión sobre lo aprendido	3	3	3	3	3	3
5	Autoevaluación sobre el contenido mostrado en el OA	3	3	3	3	3	3
6	Calidad de los contenidos	3	3	3	3	3	3
7	Adecuación de los objetivos de aprendizaje	3	3	3	3	3	3
8	Retroalimentación que proporciona el contenido mostrado:	3	3	3	3	3	3
9	Motivación	3	3	3	3	3	3
10	Organización del trabajo individual y/o colaborativo de los estudiantes	2	2	2	2	2	2
11	Nivel de presentación de recursos audiovisuales	3	2	3	3	3	3
12	Participación activa durante el aprendizaje mediante actividades interactivas	3	3	3	3	3	3
13	Indicación de meta-metadatos	0	0	0	0	0	0
14	Verificación de las fuentes de información utilizadas	3	3	3	3	3	3
15	Aporte según el contenido de los recursos audiovisuales a los estudiantes	3	3	3	3	3	2
Total		40	39	40	41	40	40

Tabla 73. Evaluación del aspecto formativo del OA-TC

No	Aspecto de Diseño y Presentación	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Correspondencia entre los recursos audiovisuales y el contenido mostrado	3	3	3	3	3	3
2	Visibilidad del texto	3	3	3	3	3	3
3	Rapidez para cargar recursos audiovisuales	2	3	2	3	2	3
4	Proporción del texto respecto a la distribución de los contenidos dentro del OA	3	3	3	3	3	2
5	El uso de colores para los contenidos	3	3	3	3	3	3
6	Manejo de formatos uniformes dentro de los OA	3	3	3	2	3	3
7	Diversidad en la representación del contenido mostrado	3	3	3	3	3	3
8	El diseño de la información audiovisual	3	3	3	3	3	3
9	Visibilidad de las imágenes	3	3	3	2	3	3
10	Evaluación del nivel de organización de las imágenes y texto	3	3	3	3	3	3
Total		29	30	29	28	29	29

Tabla 74. Evaluación del aspecto de diseño y presentación del OA-TC

No	Aspecto tecnológico	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Usabilidad	3	3	3	2	2	3
2	Accesibilidad	2	2	2	2	2	2
3	Reusabilidad	3	3	3	3	3	3
4	Facilidad de indexado del OA dentro de un repositorio	2	2	2	2	2	2
5	Compatibilidad con distintos navegadores	3	3	3	3	3	3
6	Nivel de organización de la estructura de archivos	3	3	3	3	3	3
7	Calidad de las imágenes	2	3	3	2	2	2
8	Integridad de los enlaces de navegación por la estructura didáctica	3	3	3	2	3	3
9	Correspondencia con la estructura didáctica	3	3	3	3	3	3
10	Calidad de la redacción y ortografía en la exposición del contenido	3	3	3	3	3	3
Total		27	28	28	25	26	27

Tabla 75. Evaluación del aspecto tecnológico del OA-TC

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
AF	17,14	16,71	17,14	17,57	17,14	17,14
ADP	8,29	8,57	8,00	8,00	8,29	8,29
DT	7,71	8,00	8,00	7,14	7,43	7,71
ET	33,14	33,29	33,14	32,71	32,86	33,14
Evaluación	Muy adecuado	Muy adecuado	Muy adecuado	Adecuado	Muy adecuado	Muy adecuado

Tabla 76. Resumen de evaluaciones de expertos del OA-TC

kf	0,4286	35	fmax	45
kd	0,2857	105	dmax	30
kt	0,2857		tmax	30
			emax	36,43

Valores para la escala	60%	21,86
	80%	29,14
	90%	32,79

Tabla 77. Valores para la escala

Evaluación de los expertos para el OA-MT

No	Aspecto formativo	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Presentación y explicación del tema a tratar	3	3	3	3	3	3
2	Estructuración lógica de los contenidos	3	3	3	3	3	3
3	Exhortación del desarrollo de habilidades y competencias al estudiante	3	2	3	3	2	3
4	Reflexión sobre lo aprendido	3	3	3	3	3	3
5	Autoevaluación sobre el contenido mostrado en el OA	3	3	3	3	3	3
6	Calidad de los contenidos	3	3	3	3	3	3
7	Adecuación de los objetivos de aprendizaje	3	2	2	3	2	3
8	Retroalimentación que proporciona el contenido mostrado:	3	3	3	3	3	3
9	Motivación	3	3	3	3	3	3
10	Organización del trabajo individual y/o colaborativo de los estudiantes	3	2	3	2	3	2
11	Nivel de presentación de recursos audiovisuales	3	2	3	3	3	3
12	Participación activa durante el aprendizaje mediante actividades interactivas	3	3	3	3	3	3
13	Indicación de meta-metadatos	0	0	0	0	0	0
14	Verificación de las fuentes de información utilizadas	3	3	3	3	3	3
15	Aporte según el contenido de los recursos audiovisuales a los estudiantes	3	3	3	3	3	2
Total		42	38	41	41	40	40

Tabla 78. Evaluación del aspecto formativo del OA-MT

No	Aspecto de Diseño y Presentación	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Correspondencia entre los recursos audiovisuales y el contenido mostrado	3	3	3	3	3	3
2	Visibilidad del texto	3	3	3	3	3	3
3	Rapidez para cargar recursos audiovisuales	3	3	2	3	3	2
4	Proporción del texto respecto a la distribución de los contenidos dentro del OA	3	3	3	3	3	2
5	El uso de colores para los contenidos	3	3	3	3	3	3
6	Manejo de formatos uniformes dentro de los OA	3	3	3	2	3	3
7	Diversidad en la representación del contenido mostrado	3	3	3	3	3	3
8	El diseño de la información audiovisual	3	3	3	3	3	3
9	Visibilidad de las imágenes	3	3	3	3	3	3
10	Evaluación del nivel de organización de las imágenes y texto	3	3	3	3	3	3
Total		30	30	29	29	30	28

Tabla 79. Evaluación del aspecto de diseño y presentación del OA-MT

No	Aspecto tecnológico	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Usabilidad	3	3	3	3	3	3
2	Accesibilidad	2	2	2	2	2	2
3	Reusabilidad	3	3	3	3	3	3
4	Facilidad de indexado del OA dentro de un repositorio	2	2	2	2	2	2
5	Compatibilidad con distintos navegadores	3	3	3	3	3	3
6	Nivel de organización de la estructura de archivos	3	3	3	3	3	3
7	Calidad de las imágenes	3	3	3	2	3	2
8	Integridad de los enlaces de navegación por la estructura didáctica	3	3	3	2	3	3
9	Correspondencia con la estructura didáctica	3	3	3	3	3	3
10	Calidad de la redacción y ortografía en la exposición del contenido	3	3	3	3	3	3
Total		28	28	28	26	28	27

Tabla 80. Evaluación del aspecto tecnológico del OA-MT

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
AF	18,00	16,29	17,57	17,57	17,14	17,14
ADP	8,57	8,57	8,29	8,29	8,57	8,00
DT	8,00	8,00	8,00	7,43	8,00	7,71
ET	34,57	32,86	33,86	33,29	33,71	32,86
Evaluación	Muy adecuado					

Tabla 81. Resumen de evaluaciones de expertos del OA-MT

kf	0,4286	35	fmax	45
kd	0,2857	105	dmax	30
kt	0,2857		tmax	30
			emax	36,43

Valores para la escala	60%	21,86
	80%	29,14
	90%	32,79

Tabla 82. Valores para la escala

Evaluación de los expertos para el OA-MT

No	Aspecto formativo	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Presentación y explicación del tema a tratar	3	3	3	3	3	3
2	Estructuración lógica de los contenidos	3	3	3	3	3	3
3	Exhortación del desarrollo de habilidades y competencias al estudiante	3	3	3	3	3	3
4	Reflexión sobre lo aprendido	3	3	3	3	3	3
5	Autoevaluación sobre el contenido mostrado en el OA	3	3	3	3	3	3
6	Calidad de los contenidos	3	3	3	3	3	3
7	Adecuación de los objetivos de aprendizaje	3	3	3	3	3	3
8	Retroalimentación que proporciona el contenido mostrado:	3	3	3	3	3	3
9	Motivación	3	3	3	3	3	3
10	Organización del trabajo individual y/o colaborativo de los estudiantes	2	2	2	2	2	2
11	Nivel de presentación de recursos audiovisuales	3	2	3	3	3	3
12	Participación activa durante el aprendizaje mediante actividades interactivas	3	3	3	3	3	3
13	Indicación de meta-metadatos	0	0	0	0	0	0
14	Verificación de las fuentes de información utilizadas	3	3	3	3	3	3
15	Aporte según el contenido de los recursos audiovisuales a los estudiantes	3	3	3	3	3	2
Total		41	40	41	41	41	40

Tabla 83. Evaluación del aspecto formativo del OA-TG

No	Aspecto de Diseño y Presentación	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Correspondencia entre los recursos audiovisuales y el contenido mostrado	3	3	3	3	3	3
2	Visibilidad del texto	3	3	3	3	3	3
3	Rapidez para cargar recursos audiovisuales	3	3	3	3	3	3
4	Proporción del texto respecto a la distribución de los contenidos dentro del OA	3	3	3	3	3	2
5	El uso de colores para los contenidos	3	3	3	3	3	3
6	Manejo de formatos uniformes dentro de los OA	3	3	3	2	3	3
7	Diversidad en la representación del contenido mostrado	3	3	3	3	3	3
8	El diseño de la información audiovisual	3	3	3	3	3	3
9	Visibilidad de las imágenes	3	3	3	2	3	3
10	Evaluación del nivel de organización de las imágenes y texto	3	3	3	3	3	3
Total		30	30	30	28	30	29

Tabla 84. Evaluación del aspecto de diseño y presentación del OA-TG

No	Aspecto tecnológico	Evaluador					
		E1	E2	E3	E4	E5	E6
1	Usabilidad	3	3	3	3	3	3
2	Accesibilidad	3	3	2	2	3	3
3	Reusabilidad	3	3	3	3	3	3
4	Facilidad de indexado del OA dentro de un repositorio	2	2	2	2	2	2
5	Compatibilidad con distintos navegadores	3	3	3	3	3	3
6	Nivel de organización de la estructura de archivos	3	3	3	3	3	3
7	Calidad de las imágenes	2	3	3	3	3	2
8	Integridad de los enlaces de navegación por la estructura didáctica	3	3	3	2	3	3
9	Correspondencia con la estructura didáctica	3	3	3	3	3	3
10	Calidad de la redacción y ortografía en la exposición del contenido	3	3	3	3	3	3
Total		28	29	28	27	29	28

Tabla 85. Evaluación del aspecto de tecnológico del OA-TG

ASPECTO	Evaluador					
	E1	E2	E3	E4	E5	E6
AF	17,57	17,14	17,57	17,57	17,57	17,14
ADP	8,57	8,57	8,00	8,00	8,57	8,29
DT	8,00	8,29	8,00	7,71	8,29	8,00
ET	34,14	34,00	33,57	33,29	34,43	33,43
Evaluación	Muy adecuado					

Tabla 86. Resumen de evaluaciones de los expertos del OA-TG

kf	0,4286	35	fmax	45
kd	0,2857	105	dmax	30
kt	0,2857		tmax	30
			emax	36,43

Valores para la escala	60%	21,86
	80%	29,14
	90%	32,79

Tabla 87. Valores para la escala