



Universidad de las Ciencias Informáticas
Facultad 1

**Sistema interactivo - experimental para el proceso
de enseñanza – aprendizaje de la Matemática
Discreta en la carrera Ingeniería en Ciencias
Informáticas**

Trabajo de Diploma presentado en opción al título de **Ingeniero en
Ciencias Informáticas**

Autor: José Angel Alvarez Abraira.

Tutor: MsC. Alién García Hernández.

La Habana, 2016

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 _____.

José Angel Alvarez Abraira
Firma del autor

MsC. Alién García Hernández
Firma del tutor

DATOS DE CONTACTO

DATOS DEL AUTOR:

Nombre y apellidos: José Angel Alvarez Abaira.

Correo electrónico: jaalvarez@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.

DATOS DEL TUTOR:

Nombre y apellidos: Alién García Hernández.

Correo electrónico: agarciah@uci.cu

Situación laboral: Profesor asistente.

Años de graduado: 6 años.

Especialidad de graduación: Ingeniero en Ciencias Informáticas. Máster en Ciencias Matemática (mención: enseñanza de la Matemática) por la Universidad de la Habana.

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 idea detrás de los computadores digitales puede explicarse diciendo que estas máquinas están destinadas a llevar a cabo cualquier operación que pueda ser realizada por un equipo humano.”

Alan Turing

Mi felicidad no estaría completa el día de hoy si me llegó a olvidar de esas personas que estuvieron presentes para ayudarme, por eso sería imperdonable dejar de agradecer a todos ellos. Agradezco:

A la Revolución por darme la oportunidad de cumplir mis sueños de estudiar informática y a todos los profesores que lo hicieron posible.

A mi mamá, padrasto, hermano y abuelo por su apoyo incondicional, el cariño y la confianza que depositaron en mi, y por estar siempre pendientes y preocupados acerca de mis estudios y mi vida personal. Por no dejar de pensar ni un momento en que hoy estarían sentados aquí disfrutando de este momento.

A mi tutor, quien fue mi profesor de Matemática Discreta, el profesor más querido por el grupo en mi primer año; por asignarme la responsabilidad de realizar esta tesis y depositar toda su confianza en tan importante tarea para él y por estar siempre dispuesto a ayudarme en cualquier momento.

A Eleany y Chabelly, dos personitas que sin darnos cuenta y saber por qué, nos fuimos convirtiendo en seres humanos inseparables y me brindaron su apoyo, ánimo y confianza tanto en la tesis como en mi vida personal y en estos dos años de convivencia los problemas, contratiempos, distancia, estudios, etc. no pudieron evitar que construyéramos una gran familia juntos.

A todos mis amigos y familiares que de alguna forma contribuyeron a materializar este sueño, a mis compañeros de aula quienes siempre depositaron su confianza y cariño en mi.

A Ortelio, Pita, Yoel, Rachel y Lexys que aportaron ideas y contribuyeron conmigo en la realización de esta investigación además de ser amigos en todo momento.

En general a todas las personas que hicieron posible que hoy me realice como Ingeniero en Ciencias Informáticas; a todos MIL GRACIAS.

Este triunfo de estar me forjando hoy como Ingeniero en Ciencias Informáticas no puedo dejar de dedicarlo a las personas que han sido muy importantes para mí. Es por ello que no puedo dejar pasar por alto ofrendar esta nueva victoria en mi vida. . .

A mi madre, autora y guía de mis días, quien con todo el cariño siempre confió en mí y nunca se opuso a que me encaminara en búsqueda de cumplir todas mis metas.

A mi padrastr, quien más que eso ha sido un padre y siempre me ha brindado su apoyo y preocupación incondicional y siempre ha hecho hasta lo imposible por mejorar mi estancia aquí.

A mi hermano, graduado de ingeniería informática también, que con cariño y apoyo incondicional ha estado pendiente de que me supere y aprenda algo nuevo, dispuesto a ayudarme en cualquier momento.

A mi abuelo, que no se encuentra aquí el día de hoy pero está impaciente esperando por la noticia, por cumplir todos mis antojos, confiar en mí y depositar todo su cariño.

A mi tutor que no tenía diferencias entre mañanas, tardes y noches, siempre estuvo incondicional para mí y hoy está festejando también este triunfo.

A Chabelly, a quien aún no he encontrado un término para denominarla ya que amiga se queda corto, por ser protagonista de mis felicidades, paño de mis tristezas y víctima de mis locuras en la UCI. Agradezco que finalmente, hoy pueda estar presente aquí.

A Eleany, quien más que amiga fue una hermana, las palabras para ella se quedarían cortas y en la UCI, es la otra protagonista de mis felicidades, paño de mis tristezas y víctima de mis locuras.

A Daniel, mi hermano desde segundo año, el cual por motivos personales hoy se encuentra de licencia y hoy no está graduándose con nosotros.

A todos mis familiares, amigos, compañeros de aula y profesores que siempre creyeron en que este sueño se haría realidad.

A todas las personas aquí presentes hoy, por acercarse a ser protagonistas de este sueño hecho realidad.

Y por último y muy importante a una persona que tanto deseó ser partícipe de este momento, tanto luchó porque hoy me forje como Ingeniero: "mi abuela", quien por cuestiones de la vida hoy no se encuentra presente luego de abandonar este mundo inesperadamente hace un poco más de 2 años y 4 meses, pero sé que desde donde quiera que se encuentre en este momento nos está mirando y festejando este triunfo.

La Universidad de las Ciencias Informáticas (UCI) es un centro de altos estudios para formar profesionales de la informática que no está ajeno a las nuevas tendencias del uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el Proceso de Enseñanza – Aprendizaje (PEA). En la presente investigación se realizó un estudio acerca del PEA de la Matemática Discreta (MD), el cual se encuentra muy limitado por varios factores que traen consigo un desaprovechamiento de las potencialidades de las TIC para el apoyo a la educación. Para ello fueron utilizados métodos a nivel teóricos y empíricos como la modelación, entrevistas, encuestas, etc., que de manera general permitieron conocer el estado actual del PEA de la MD en la UCI para luego encontrar una propuesta de solución acorde a las necesidades existentes. Luego de ser implementada la propuesta de solución se obtuvo un sistema web interactivo - experimental que representa un gran aporte para el PEA de la MD, al brindar un sistema donde los estudiantes y profesores sean los gestores de su propio conocimiento a través de la experimentación, evaluaciones, consulta de contenido y competencias en línea acerca de seis (6) de los temas que se imparten en la asignatura. El sistema puede ser aplicable a cualquier contexto educativo ya que además de poder ser utilizado en los cursos presenciales o semi-presenciales, también puede ser utilizado en la educación a distancia que es una de las nuevas tendencias alcanzadas por la educación.

Palabras claves: contenido, competencia, evaluación, experimental, interactivo, proceso de enseñanza – aprendizaje

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
DESARROLLO.....	8
CAPÍTULO 1: LOS OBJETOS DE APRENDIZAJES INTERACTIVOS - EXPERIMENTALES Y SU UTILIZACIÓN EN EL PROCESO DE ENSEÑANZA – APRENDIZAJE DE LA MATEMÁTICA DISCRETA.	8
1.1 Los sistemas web en el proceso de enseñanza – aprendizaje de la Matemática Discreta.	8
1.2 Caracterización del uso de los Objetos de Aprendizaje en el Proceso de Enseñanza – Aprendizaje de la Matemática Discreta en la UCI.	15
1.3 Metodologías y herramientas para el desarrollo del Sistema Web Interactivo - Experimental como apoyo a la docencia.	18
1.3.5 Tecnologías, herramientas y lenguajes a utilizar para el desarrollo del Objeto de Aprendizaje Interactivo - experimental.	26
Conclusiones parciales del capítulo.	31
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA INTERACTIVO - EXPERIMENTAL.....	32
2.1 Análisis.	32
2.1.1 Análisis de los contenidos de la MD que contendrá el OA-IE a desarrollar.....	32
2.1.2 Modelo de dominio.....	33
2.2 Diseño.	34
2.2.1 Unidad de información.....	34
2.2.2 Unidad de experimentación.....	35
2.2.3 Unidad de evaluación y retroalimentación.....	35
2.2.4 Funcionalidades del sistema.....	36
2.2.5 Requisitos no funcionales.....	38
2.2.6 Historias de usuario.....	39
2.2.7 Plan de entregas.....	40
2.2.8 Plan de iteraciones.....	41
2.2.9 Tarjetas CRC.....	42
2.2.10 Arquitectura del software.....	42
2.2.11 Modelo de datos.....	47
2.2.12 Diagrama de despliegue.....	48
Conclusiones parciales del capítulo.	49
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA PROPUESTA DE SOLUCIÓN.	50
3.1 Estándar de codificación.	50
3.2 Pruebas funcionales al sistema.	52
3.2.1 Pruebas de aceptación.....	52
3.2.2 Evaluación de la calidad desde el punto de vista pedagógico.	56
Conclusiones parciales del capítulo.	57
CONCLUSIONES.....	58

RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS	60
GLOSARIO DE TÉRMINOS	I
ACRÓNIMOS	II

Tabla 1: Relación de temas de la MD que se imparten en la ICI. (UCI, Plan de Estudio "D" del Ingeniero en Ciencias Informáticas, 2013)	9
Tabla 2: Comparación entre las metodologías tradicionales y ágiles. (Figuroa, Solís, & Cabrera, 2007)	20
Tabla 3: Relación de funcionalidades definidas para el sistema (Elaboración propia)	36
Tabla 4: HU_1 Experimentar con la Teoría de conjuntos (Elaboración propia)	39
Tabla 5: HU_2 Exponer teoría de la Teoría de conjuntos (Elaboración propia)	40
Tabla 6: Plan de entregas (Elaboración propia)	40
Tabla 7: Plan de iteraciones (Elaboración propia)	41
Tabla 8: Tarjeta CRC "Contenido" (Elaboración propia)	42
Tabla 9: Tarjeta CRC "Experimentacion.Conjuntos" (Elaboración propia)	42
Tabla 10: Estándares de codificación a utilizar en la implementación del sistema (Guía de estilo para el código Python - PEP 8 en Español, 2013)	50
Tabla 11: Caso de prueba de aceptación "Exponer contenido de la Teoría de conjuntos"	52
Tabla 12: Caso de prueba de aceptación "Experimentar con las Máquinas de Turing"	53
Tabla 13: Escala para medir la calidad del OA-IE (García A. , 2014)	56

Ilustración 1: Objeto de Aprendizaje Interactivo - Experimental. (García A. , 2014).....	12
Ilustración 2: Modelo de dominio (Elaboración propia).	34
Ilustración 3: Funcionamiento del MTV de Django (Infante, 2012).	43
Ilustración 4: Ejemplo de utilización del patrón MVT en el sistema (Elaboración propia).....	44
Ilustración 5: Código para obtener los cinco (5) primeros estudiantes del ranking de las competencias en línea (Elaboración propia).	45
Ilustración 6: Código para mostrar el listado de grupos docentes eliminados (Elaboración propia).	45
Ilustración 7: Código para validar la cardinalidad de un conjunto (Elaboración propia).	46
Ilustración 8: Código para incluir a un estudiante una vez que haya sido eliminado (Elaboración propia).	46
Ilustración 9: Código para deshabilitar a un administrador del sistema (Elaboración propia).	47
Ilustración 10: Modelo de datos (Elaboración propia).	48
Ilustración 11: Diagrama de despliegue (Elaboración propia).....	49

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) son el conjunto de herramientas aplicadas principalmente a la gestión de la información, que abarcan un abanico muy amplio de soluciones. De acuerdo a (Corrales, 2009):

Las TIC 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.

Cuando se habla de las TIC, se hace referencia a toda aquella tecnología de algún modo, está coordinada por un ordenador. Nos estamos refiriendo a los programas informáticos, al multimedia, al video digital, CD-ROM¹, DVD², a los CD musicales, a los videojuegos... y, sobre todo, a los grandes avances de la red de redes.”

De manera general el autor de esta investigación asume a las TIC como un conjunto de medios ya sea a través de radio, televisión, informática, telefonía, etc., que permite la captura, producción, almacenamiento, tratamiento y presentación de información mediante distintos medios (voz, imágenes, señales acústicas, videos, etc.), específicamente se hace énfasis en la utilización de los programas informáticos.

En la actualidad las TIC se han desarrollado aceleradamente, lo que ha traído consigo un elevado auge en el desarrollo de la sociedad, por lo que constituye así un factor de incalculable relevancia en la gran mayoría de los procesos para brindar todas las facilidades que sus ventajas presentan (Huidobro, 2007):

- Brindar grandes beneficios y adelantos a la salud y la educación.
- Potenciar a las personas y actores sociales de redes de apoyo e intercambio y listas de discusión.
- Apoyar a personas y empresas a presentar y vender sus productos a través de la Internet.
- Impartir nuevos conocimientos para la empleabilidad que requieren muchas competencias (integración, trabajo en equipo, motivación, disciplina, etc.).
- Ofrecer nuevas formas de trabajo, como teletrabajo.

¹ Del inglés “*Compact Disc-Read Only Memory*”.

² Del inglés “*Digital Versatile Disc*”.

- Brindar facilidades, exactitud, menores riesgos y costos.
- Dar acceso al flujo de conocimientos e información para empoderar y mejorar las vidas de las personas.
- Permitir el aprendizaje interactivo y la educación a distancia.

Precisamente estas dos últimas ventajas sirven como base a lo que hoy se le denomina “Tecnología educativa” (TE) que consiste en:

un espacio de conocimiento pedagógico sobre los medios, la cultura y la educación en el que se cruzan las aportaciones de distintas disciplinas de las ciencias sociales; es una disciplina que estudia los procesos de enseñanza y de transmisión de la cultura mediados tecnológicamente en distintos contextos educativos. (Area, 2009)

En décadas anteriores la misma ha oscilado entre dos visiones o perspectivas: Una es la representada por la equiparación de TE a los medios y recursos instructivos (fundamentalmente audiovisuales). La otra perspectiva, ha sido entender la TE como un campo de estudio caracterizado por diseñar y controlar científicamente los procesos de enseñanza. (Area, 2009)

De manera general el autor de esta investigación asume a las TE como las TIC aplicadas a la educación, en este caso específicamente la informática.

La tecnología educativa ha evolucionado bastante en las últimas décadas, como podemos observar por su presencia en diferentes currículums de formación de profesores y profesionales de la enseñanza, el aumento de sus investigaciones, la proliferación de sus publicaciones, y el crecimiento de sus Jornadas y Eventos. Por otra parte, ha sido una de las disciplinas que más ha evolucionado como consecuencia de los cambios internos que han sufrido las ciencias por las transformaciones que en líneas generales la tecnología está recibiendo desde el movimiento de “Ciencia, Tecnología y Sociedad”. (Cabero, 2006)

Como muestra de los elementos que componen a la TE es de vital importancia señalar el papel que han jugado las multimedias, Entornos Virtuales de Enseñanza – Aprendizaje (EVEA) y Objetos de Aprendizaje (OA); todos ellos están diseñados para cumplir distintos objetivos pero un elemento muy importante que contienen en común es que constituyen modelos semi-presenciales o educación a distancia donde se definen las siguientes ventajas según (Moya, 2009):

- El aula se concibe como un espacio virtual, donde se relacionan profesor y alumno, así como los propios alumnos entre sí, lo que permite eliminar las barreras geográficas existentes.
- Los discentes³ pueden adaptar los calendarios y horarios, fijando el ritmo de trabajo que ellos consideren más adecuado en función de sus necesidades. Esto permite el desarrollo de una relación asíncrona, donde no es necesario que profesor y alumno coincidan en el mismo momento de tiempo.

Como se puede observar este nuevo modelo de educación ofrece una mayor flexibilidad a los alumnos especialmente en la fijación de sus propios horarios.

Esto no implica que las TIC eduquen ni reemplacen al profesor.

Son un recurso poderoso cuya incorporación debe considerar nuevos roles al profesor, nuevos contenidos, programas y formas de evaluar. Su incorporación exige revisar sistémica, corporativa y transversalmente los actuales contextos educativos, desde la estimación de la hora y cargas académicas, hasta las mallas y contenidos. (Benvenuto, 2003)

Como se expresó anteriormente los OA constituyen un elemento importante dentro de las TE, los cuales según la (Serrano, 2010) se definen como cualquier entidad, digital o no digital, que puede ser utilizada, reutilizada o referenciada durante el aprendizaje apoyado en la tecnología. Como ejemplos de aprendizajes apoyados por la tecnología se incluyen: los sistemas de entrenamiento basados en computadoras, los ambientes de aprendizaje interactivos, los sistemas inteligentes de instrucción apoyada por computadoras, sistemas de aprendizaje a distancia y los ambientes de aprendizaje colaborativo.

Además (Wiley, 2000) definió a los OA como cualquier recurso digital que puede ser reusado como soporte para el aprendizaje.

Los acervos de objetos de aprendizaje pueden ser utilizados por docentes que localizan, en un sistema compartido en red, recursos que puede utilizar dentro de sus cursos. Estos recursos aluden a una temática, tratándolo de manera unitaria: conteniendo la información requerida para lograr un objetivo de aprendizaje (Chan, 2001); por lo que también puede tratarse a los sistemas de aprendizaje como OA ya que

³ Sinónimo de alumno.

precisamente la presente investigación consiste en la elaboración de un sistema web interactivo - experimental.

La UCI es un centro cubano de altos estudios con la misión de:

Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio – trabajo como modelo de formación. Servir de soporte a la industria cubana de la informática. (UCI, Portal UCI, 2010)

En función de cumplir con la misión encomendada la UCI cuenta con el Plan de Estudio de Ingeniería en Ciencias Informáticas (ICI), dentro de la cual se incluye la Matemática Discreta (MD) como una de las asignaturas esenciales en el primer año de la carrera. Los temas que se imparten en MD se encuentran divididos en dos asignaturas Matemática Discreta 1 (MD1) y Matemática Discreta 2 (MD2) durante el primer y segundo semestre respectivamente, impartándose los siguientes temas: Teoría de conjuntos, Relaciones binarias, Lógica, Técnicas de demostraciones, Teoría de la computabilidad, Relaciones de recurrencia, Teoría combinatoria y Teoría de grafos. A lo largo de la investigación se tratará a la MD como una sola asignatura independientemente de que se imparta en ambos semestres.

Tras un estudio realizado a las opciones con las que contaban los estudiantes y profesores para realizar su auto - preparación con el apoyo de las TIC se pudo evidenciar la presencia de un EVEA donde se pueden encontrar contenidos, orientaciones y recursos de la asignatura MD; pero al realizar un análisis a los recursos de la asignatura MD del EVEA se percibe que en dichos espacios solo se encuentran documentos, la mayoría 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.

Por otra parte, en el Repositorio de Objetos de Aprendizaje (ROA) de la UCI se encuentra una pobre variedad de OA relacionados con la MD. Por lo general los OA publicados poseen un bajo nivel de interactividad y no permiten la experimentación, lo que trae consigo la poca comprensión de los contenidos más complejos y el bajo efecto de realismo del contenido presentado.

En conjunto con lo anteriormente expuesto se detectan deficiencias en el Proceso de Enseñanza – Aprendizaje (PEA) de la MD en la UCI, por lo que de manera general se plantea la siguiente **situación problemática**:

- La asignatura posee un alto nivel de dificultad para los estudiantes y la mayoría del claustro de profesores no posee experiencia en su impartición; lo que dificulta el PEA.
- Solo se encuentran disponibles para el estudio individual de los estudiantes, documentos en formato “.doc”, “.pdf” y “.ppt”, de esta manera se desaprovechan las ventajas que brindan las TIC en cuanto a la utilización de OA para ayudar en la asimilación de los contenidos con un nivel mayor de dificultad.
- Existe dentro del programa de la MD una pobre variedad de ejercicios y tareas que limitan la asimilación adecuada de los contenidos.
- Insuficiente existencia de recursos que apoyen el PEA de la MD para los contenidos de más difícil y lenta asimilación, que pueden ser representados gráficamente.
- Por lo general, los OA de MD que existen en la universidad, son limitados y poseen un bajo nivel de interactividad. Ninguno de ellos permite la experimentación.

Toda esta situación trae consigo una contradicción, pues a pesar de las altas exigencias de la formación de los estudiantes en la asignatura MD en la UCI, no se disponen de recursos, que basados en el uso de las TIC, amplíen las capacidades de asimilación de los contenidos.

En correspondencia con lo antes expuesto se plantea como **problema de investigación**: ¿Cómo contribuir, con el uso de las TIC, al proceso de enseñanza – aprendizaje de la Matemática Discreta en la carrera Ingeniería en Ciencias Informáticas?, por lo que se define como **objeto de estudio** el proceso de enseñanza – aprendizaje de la Matemática Discreta en la Universidad de las Ciencias Informáticas.

Para dar solución al problema existente la presente investigación se traza como **objetivo general**: Desarrollar un Sistema Web interactivo - experimental para apoyar el proceso de enseñanza – aprendizaje de la Matemática Discreta en la carrera Ingeniería en Ciencias Informáticas; por lo que se plantea como **campo de acción** los objetos de aprendizaje para la enseñanza de la Matemática Discreta con énfasis en los sistemas web.

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

1. ¿Cuáles son los presupuestos teóricos que sustentan el desarrollo y la utilización de un Sistema Web interactivo - experimental en el proceso de enseñanza – aprendizaje de la Matemática Discreta en la Ingeniería en Ciencias Informáticas?

2. ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y diseño del Sistema Web interactivo - experimental para el proceso de enseñanza – aprendizaje de la Matemática Discreta en la Ingeniería en Ciencias Informáticas?
3. ¿Cómo implementar, a partir del análisis y diseño realizado, el Sistema Web interactivo - experimental para el proceso de enseñanza – aprendizaje de la Matemática Discreta en la Ingeniería en Ciencias Informáticas?
4. ¿Qué resultados se obtendrán al validar el Sistema Web interactivo - experimental para el proceso de enseñanza – aprendizaje de la Matemática Discreta en la Ingeniería en Ciencias Informáticas?

Para ello se plantean las siguientes **tareas**:

1. Sistematización de los principales referentes teóricos que sustentan el desarrollo y la utilización de una Sistema Web interactivo - experimental en el proceso de enseñanza- aprendizaje de la Matemática Discreta en la Ingeniería en Ciencias Informáticas.
2. Análisis y diseño del Sistema Web interactivo - experimental.
3. Implementación del Sistema Web interactivo - experimental.
4. Valoración de la calidad como software y como recurso pedagógico del Sistema Web interactivo - experimental elaborado.

Para el desarrollo de esta investigación se utilizan los siguientes métodos a **nivel teórico**:

- **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.
- **Inductivo – deductivo:** Permite arribar a conclusiones lógicas a partir de conocimientos antes adquiridos, así como el planteamiento de los objetivos y algunos requisitos principales para la realización de los Objetos de Aprendizajes Interactivos - experimentales (OA-IE).
- **Analítico – sintético:** Permite seleccionar y resumir los elementos más relevantes relacionados con los OA-IE, además detalla la información para el modelado del negocio y el desarrollo del OA sobre tecnologías web.

Además de los siguientes métodos a **nivel empírico**:

- **Entrevista:** Permite, a través de una serie de preguntas con el cliente obtener la información necesaria que permita determinar las características, cualidades y requisitos que debe tener el objeto de aprendizaje interactivo - experimental a implementar.

- **Encuestas:** Se realizan a profesores y a estudiantes, para recoger datos y criterios que permitan mantener actualizado el diagnóstico de la situación problemática.
- **Revisión documental:** Permite analizar los documentos publicados en el espacio creado por el EVEA de la UCI para la MD y determinar las características que deben tener los OA.
- **Estadística:** Permite el análisis de las encuestas y la revisión documental de las notas de los estudiantes y para valorar la factibilidad de aplicación de los OA-IE.

El presente documento de trabajo de diploma está compuesto por introducción, tres (3) capítulos, conclusiones generales, recomendaciones y referencias bibliográficas que fueron utilizadas en el desarrollo de la investigación, y por último para complementar la investigación se presentan una serie de anexos.

El capítulo 1 titulado “Los Objetos de Aprendizajes interactivos - experimentales y su utilización en el proceso de enseñanza – aprendizaje de la Matemática Discreta” aborda los aspectos teóricos que apoyan la investigación, se definen conceptos fundamentales, la importancia de la MD para la informática y del empleo de los OA principalmente basados en tecnologías web, además de la caracterización del estado de la enseñanza de la MD en la UCI y un estudio de las metodologías, herramientas y tecnologías a utilizar en el proceso de desarrollo.

El capítulo 2 se titula “Análisis y diseño del Sistema interactivo – experimental” y en su contenido se realiza un estudio de la encuesta realizada a profesores y estudiantes que ya cursaron la asignatura MD, la cual permite conocer el estado del PEA de la MD en la UCI y la elección de los contenidos de la MD que serán incluidos en el sistema. Además, se define la propuesta de solución y se generan una serie de artefactos que permiten una mejor comprensión de lo que se desea implementar en el OA-IE.

En el capítulo 3 titulado “Implementación y pruebas de la propuesta de solución” se definen una serie de estándares de codificación que debe seguir la implementación del sistema y una serie de pruebas al sistema, para ello se realizan pruebas de aceptación en conjunto con el cliente, pruebas de cargas y estrés y la validación como recurso pedagógico a través de la guía propuesta por Toll Palma en el año 2011.

DESARROLLO

CAPÍTULO 1: Los Objetos de Aprendizajes Interactivos - Experimentales y su utilización en el proceso de enseñanza – aprendizaje de la Matemática Discreta.

En el presente capítulo se exponen los principales elementos teóricos que respaldan a esta investigación. Para ello se comienza realizando un análisis acerca de la MD en la ICI, se define así la importancia que tiene para la carrera y otros aspectos que evidencian la necesidad de la utilización de los OA-IE en el PEA de la MD en la UCI, así como los fundamentos teóricos que sustentan a los OA-IE.

1.1 Los sistemas web en el proceso de enseñanza – aprendizaje de la Matemática Discreta.

La MD es la parte de las matemáticas que estudia objetos discretos.

Definir el concepto “discreto” sin entrar en demasiadas formalidades no es sencillo, pero se puede apelar a ciertos ejemplos matemáticos conocidos y contraponerlo al concepto de continuo que es la idea central del curso de Bases de Matemáticas. Lo discreto es lo finito o lo que, si no es finito, presenta el aspecto de los números naturales, objetos bien separados entre sí; lo continuo es lo no finito, lo infinitesimalmente próximo, como los números reales, y de ahí el concepto de límite y las ideas que de dicho concepto se derivan. Esta rama de las matemáticas surge como una disciplina que unifica diversas áreas tradicionales de las Matemáticas (combinatoria, probabilidad, geometría de polígonos, aritmética, grafos,...), como consecuencia de, entre otras cosas, su interés en la informática y las telecomunicaciones: la información se manipula y almacena en los ordenadores en forma discreta (palabras formadas por ceros y unos), se necesita 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). (Álvarez, 2009)

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. Dicho esto, surgen así cursos y libros de textos que incluían temas muy clásicos y modernos de la Matemática. Entre los primeros se pueden mencionar la Teoría Elemental de Números, la Combinatoria de la enumeración, la Teoría de Grafos, Árboles, Lógica proposicional, Lógica de predicados, Lógica difusa, Álgebra de Boole, Lenguajes y Máquinas de Estados Finitos, Teoría de Algoritmos, Complejidad Computacional, etc. (Herrera, 2012).

Como bien puede evidenciarse los temas que aborda la MD están vinculados estrechamente al área de la informática, de ahí la importancia que esta asignatura tiene para la ICI, pues todas las estructuras y procesos que son objeto de la MD se pueden asociar con algún concepto de la informática, por ejemplo:

“Formalizar” con lenguajes de programación, “procesos finitos” con algoritmos, “números enteros” con criptografía, “contar” con estudiar la capacidad de una computadora o la complejidad de un algoritmo, “enumerar” con Bases de Datos (BD), “representar las relaciones” con planos de redes... Además, la MD provee a los estudiantes de madurez en el razonamiento lógico y de las habilidades matemáticas necesarias para resolver problemas; y de los fundamentos matemáticos para muchas asignaturas de la especialidad, como estructuras de datos, algoritmos y lenguajes formales, BD, compilación, seguridad informática, sistemas operativos e inteligencia artificial. (UCI, Entorno Virtual de Enseñanza Aprendizaje, 2014)

Debido a la importancia de la MD para la ICI y de acuerdo a lo expuesto en el Plan de Estudio de la institución (UCI, Plan de Estudio "D" del Ingeniero en Ciencias Informáticas, 2013) la MD se incluye como asignatura para perseguir los siguientes objetivos:

- Adquirir una concepción y métodos científicos para la solución de problemas a través del conocimiento de los desarrollos fundamentales de la MD para la modelación y la simulación de estructuras y procesos que intervienen en la solución computacional de problemas.

Para dar cumplimiento a los objetivos planteados a lo largo del primer año se imparten los siguientes temas:

Tabla 1: Relación de temas de la MD que se imparten en la ICI. (UCI, Plan de Estudio "D" del Ingeniero en Ciencias Informáticas, 2013)

Matemática Discreta 1	Matemática Discreta 2
Tema 1: Teoría de conjuntos y relaciones binarias.	Tema 1: Teoría combinatoria
Tema 2: Lógica matemática.	Tema 2: Relaciones de recurrencia.
Tema 3: Teoría de la computabilidad.	Tema 3: Teoría de grafos.

De acuerdo con (García A. , 2014) la mayoría de estos temas se tornan complicados para los estudiantes, y precisamente es esta la principal razón por la que sería de mayor utilidad apoyarse en las TIC para mejorar el PEA.

Las TIC pueden cambiar el trabajo de los alumnos y profesores en la enseñanza y el aprendizaje.

Cuando los ordenadores estén plenamente al alcance de los alumnos y los profesores estén bien preparados para usarlos, los alumnos podrán realizar la mayor parte de las tareas de clase utilizando recursos de la red, preparando trabajos en el ordenador y consultando bases de datos especiales y software educativo que los ayuden a entender mejor las matemáticas. (Carnoy, 2004)

Existen diferentes medios tecnológicos que pueden ser utilizados en el PEA, a continuación, se mencionan algunos de ellos, los cuales permiten desarrollar habilidades de forma práctica en los discentes:

Multimedias: Desde la aparición de la digitalización, se ha utilizado la multimedia para representar una combinación de diferentes tipos de medios, tales como imágenes, archivos de audio y videos. Este término se ha convertido en el medio central para comunicar y cooperar con los demás en la vida cotidiana. Por lo tanto, ya no solo se limita a la documentación y conservación, entretenimiento o colecciones multimedia personales; más bien, se ha convertido en una parte integral de las herramientas y sistemas que están proporcionando soluciones a los desafíos, incluyendo la sociedad de hoy en día ante los retos relacionados con la salud, el envejecimiento, la educación, la participación social, la sostenibilidad y el transporte inteligente. Los medios de comunicación y la multimedia han evolucionado y permitido habilitar aplicaciones interactivas y de cooperación en el seno de la sociedad. (ASPP, 2015)

Entornos Virtuales de Enseñanza – Aprendizaje (EVEA): Permite acomodarse a la disponibilidad horaria de los alumnos, con la concepción integradora-educacional de uso de recursos virtuales basándose en el paradigma pedagógico donde los alumnos tienen un rol activo en su proceso de aprendizaje basado en la exploración y crítica del conocimiento, con los nuevos enfoques visuo-espaciales que permite el espacio virtual con la posibilidad del intercambio de información bidireccional alumnos – alumnos y alumnos – tutores (Algieri, Mazzoglio, & Castro, 2012).

Objetos de aprendizaje (OA): Como se expresó en la introducción, este término posee una gran variedad de definiciones por parte de distintos autores. Para (López C. , 2010) un OA es una unidad del aprendizaje independiente y autónomo que está predispuesto a su reutilización en diversos contextos instruccionales. Pero sin embargo (Chan, 2001) lo define 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.

Además, un OA está definido también por (Chiappe, 2009) 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. Por otra parte se puede referir al término OA como un recurso digital modular identificado de manera única y etiquetado con metadatos⁴, que puede ser utilizado para el apoyo de la enseñanza (MÓDULO IV: TIC'S Y ENTORNOS VIRTUALES, 2008).

Pero debido a la variedad de definiciones que existen, el autor de esta investigación asume la de OA-IE definida por (García A. , 2014) que lo presenta como:

Un software de carácter educativo que permite la manipulación de sus parámetros a partir de la interacción de quien lo utiliza y produciendo una retroalimentación. Posee un contenido claramente identificable. Su principal potencial es la reutilización dentro de distintos contextos educativos, además de posibilitar la evaluación automatizada del aprendizaje del contenido”.

Entiéndase por software interactivo a todos los programas de computador que gestionan contenidos audiovisuales. El concepto de interactividad debiera verse más como un grado que como una conquista. La interactividad entendida como relación entre el hombre y la máquina de forma que cada uno de ellos responde a los estímulos del otro. Entre el software educativo podemos encontrar diseños con un elevado grado de interactividad que sofistican la relación con el usuario (De la Rosa, 2006).

Por otro lado, el autor de esta investigación considera que un software interactivo no solo debe definirse por la relación usuario – máquina, sino también por la interactividad usuario – usuario, la cual es muy importante para un sistema educativo debido a que permite realizar foros, encuestas, respuestas a inquietudes, entre otros. En este tema es importante destacar la importancia que posee la interacción entre

⁴ Datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos. Es "Información sobre información" o "datos sobre los datos".

estudiantes, así como entre estudiante – profesor, este aspecto es uno de los fundamentales que posibilita el desarrollo de la educación a distancia.

Pero la interactividad no es solamente el principal aspecto, pues uno de los mayores problemas en la mayoría de los sistemas existentes es que solo se dedican a la presentación de información, sin permitir una retroalimentación, donde el estudiante pueda comprobar resultados obtenidos en problemas resueltos durante su estudio, donde pueda aplicar los conocimientos adquiridos; es ahí donde se introduce el término de software experimental conceptualizado por (Método experimental, 2012) como un programa basado en la observación y manipulación sistemática de la realidad para evolucionar el conocimiento que se tiene sobre un fenómeno concreto; implica la observación, manipulación, registro de variables (dependiente, independiente, intervinientes, etc.) que afectan un objeto de estudio.

A raíz de lo expresado anteriormente el autor arriba a la conclusión de que un software interactivo – experimental consiste en un programa informático que conlleva a una estrecha relación entre los estudiantes y profesores, y brinda una herramienta donde el usuario pueda observar el comportamiento de determinadas variables en cualquier instante de tiempo. Este tipo de medio de enseñanza se corresponde con las tendencias actuales de la educación a distancia y para observar algunas de las ventajas que lo sitúan por encima de los OA tradicionales se puede interpretar la siguiente figura:



Ilustración 1: Objeto de Aprendizaje Interactivo - Experimental. (García A. , 2014)

Una buena práctica para desarrollar un OA-IE es hacerlo a través de sistemas web, debido a que estos posibilitan la interconexión de usuarios a través de la Internet.

Los sistemas web, conocidos también como aplicaciones web, son aquellos que se alojan en un servidor de Internet o sobre una Intranet (red local).

Su aspecto es similar al de las páginas web, pero en realidad los sistemas web poseen funcionalidades muy potentes que brindan respuestas a casos particulares. Estos sistemas pueden ser utilizados en cualquier navegador web (Chrome, Firefox, Internet Explorer, etc.) independientemente del sistema operativo que se utilice. (Báez, 2012)

Las aplicaciones web trabajan con BD que permiten procesar y mostrar información de forma dinámica para el usuario. Los sistemas desarrollados en plataformas web tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan como para los usuarios que operan en el sistema. Este tipo de diferencias se ven reflejadas en los costos, en la rapidez de obtención de la información, en la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable.

Pero esta investigación está basada en la utilización de los sistemas web en el PEA.

La integración entre lo social y lo tecnológico, donde las nuevas herramientas y aplicaciones proporcionan servicios a los usuarios, y esos usuarios generan contenidos, información y comunicación conlleva a un amplio abanico de posibilidades a nivel educativo, puesto a que permite la participación social de un grupo de personas para elaborar una serie de contenidos, saltando la barrera de la individualidad en la formación a través de las nuevas tecnologías, y acercándose más a la filosofía del profesor como mediador, y al alumno como verdadero valedor de sus conocimientos, convirtiéndolo en una parte muy activa de su formación, e incluso la formación a cualquier hora y en cualquier lugar, siempre que podamos acceder a la información a través de un dispositivo móvil. (Moreno, 2012)

(Moreno, 2012) en su trabajo, también expone las características que se deben tener presente a la hora de aplicar un sistema web con carácter educativo:

- Interactividad: es fundamental que la web permita un contacto interactivo entre dos o más sujetos.
- Aplicaciones dinámicas y de estándares abiertos, donde el contenido sea modificable continuamente por la participación de distintos usuarios que accedan a una determinada web.
- Colaborativas y participativas: se debe fomentar la colaboración y participación entre los usuarios a la hora de elaborar contenidos o de aclarar informaciones ofrecidas o solicitadas por éstos.

- Aplicaciones simples e intuitivas: debe estar adaptada a todo tipo de usuario, desde los más experimentados en temas informáticos hasta aquellos que presentan un nivel básico.
- Carácter Beta: puesto que todos los contenidos y aplicaciones relacionados con la Web 2.0 se van mejorando continuamente.
- Gratuidad de las aplicaciones: permite la mayor participación entre todos los usuarios de la red.

Se define según (O'Reilly, 2003) a la Web 2.0 como un cambio de paradigma sobre la concepción de Internet y sus funcionalidades, ahora la unidireccionalidad se abandona y se orientan más a facilitar la máxima interacción entre los usuarios y el desarrollo de redes sociales donde puedan expresarse y opinar, buscar y recibir información de interés, colaborar y crear conocimiento, compartir contenidos. Como ejemplos de Web 2.0 se pueden distinguir:

- Aplicaciones para expresarse/crear y publicar/difundir: blog, wiki...
- Aplicaciones para publicar/difundir y buscar información: podcast, YouTube...
- Aplicaciones para buscar/acceder a información de la que nos interesa estar siempre bien actualizados: RSS, Bloglines, GoogleReader...
- Redes sociales: Ning, Twitter, Second Life...
- Otras aplicaciones: calendarios, libros virtuales compartidos, plataformas de tele formación, pizarras digitales...

Como puede observarse los recursos educativos basados en tecnologías web forman parte también de la web 2.0 y es importante destacar que independientemente de la distancia o de si la enseñanza es presencial, es el estudiante quien toma las decisiones sobre el aprendizaje. Estas decisiones según (de Kereki, 2003) afectan a todos los aspectos del aprendizaje y si éste se realizará o no, como son:

- ¿qué aprendizaje?: selección del contenido o destreza.
- ¿cómo aprender?: métodos, media, itinerario.
- ¿dónde aprender?: lugar del aprendizaje.
- ¿cuándo aprender?: comienzo y fin, ritmo.
- ¿a quién recurrir?: tutor, amigos, colegas, profesores...
- ¿cómo será la valoración del aprendizaje?
- Aprendizajes posteriores.

1.2 Caracterización del uso de los Objetos de Aprendizaje en el Proceso de Enseñanza – Aprendizaje de la Matemática Discreta en la UCI.

De manera general se puede decir que los Repositorios de Objetos de Aprendizaje (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*⁵, preparadas tecnológicamente para inter-operar 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 intercomunicarse 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. (Herrera, 2012)

En el ROA de la UCI se encuentran OA relacionados con la MD como: Ejercicios resueltos sobre Teoría Combinatoria, Entrevista a Alan Turing, Introducción a las deducciones lógicas, Mapas de Karnaugh, Sistema Numérico Binario, entre otros. Estos OA no abarcan todos los contenidos de la asignatura y además poseen dos grandes dificultades (García A. , 2014):

- Bajo nivel de interactividad: impide dar el control de navegación a los usuarios para que exploren a su propia voluntad el contenido, no emplean mecanismos para la evaluación, retroalimentación y la colaboración.
- No permiten la experimentación: trae consigo que los estudiantes no puedan experimentar con los temas de la asignatura.

⁵ Educación y capacitación a través de Internet.

Esto trae consigo el bajo efecto de realismo en el contenido presentado y una elevada dificultad en la comprensión de la información abstracta o completa, aun cuando existen contenidos como las Máquinas de Turing y la Teoría de grafos que permiten y es de gran importancia su representación gráfica.

Además de la poca existencia de OA en el ROA de la UCI se puede evidenciar que la gran mayoría de los estudiantes desconocen la existencia del ROA, y los que la conocen no utilizan los OA que el mismo proporciona, limitándose así de un aprendizaje interactivo a través de mapas conceptuales, simulaciones, gráficos, animaciones y otros elementos multimedia.

Una correcta implementación, divulgación y utilización de un OA-IE elevaría la calidad del aprendizaje de la MD en la UCI. Es por eso que se realiza un estudio de sistemas ya existentes relacionados con este tema, y en el ámbito nacional se encontraron los siguientes:

- **Conjunto de Objetos de Aprendizaje para la Matemática Discreta:** Una tesis desarrollada en el 2012 en la UCI. Consiste en brindar un conjunto de OA-IE para apoyar el PEA. Se trata de una aplicación de escritorio desarrollada sobre el lenguaje de programación Java donde se detecta como primera deficiencia que se abordan solamente los temas Teoría de conjuntos, Teoría de grafos, Máquinas de Turing, Teoría combinatoria y Relaciones de Recurrencia. Además, es un sistema muy estático donde se brinda el contenido de cada uno de esos temas de una forma poco amigable para el usuario. En lo que corresponde a la experimentación solo se permiten realizar operaciones sencillas, no cumple con todos los elementos y operaciones que se pueden realizar en cada uno de los temas.

Para la experimentación y evaluación ya existen definidos un conjunto de operaciones, por lo que no permite al usuario ser el creador de sus ejercicios ni existe una relación entre la teoría de cada tema y la práctica. Positivamente se refleja la representación gráfica de los contenidos que lo necesitan.

Es importante no dejar a un lado que al ser una aplicación de escritorio no existe una interacción entre usuarios que la consumen, por lo tanto, no permite la educación a distancia.

A modo de resumen no se evidencia correctamente la definición de lo que es un OA-IE en realidad.
- **Caribbean Mind Forge (CMF):** No constituye un OA para la MD, pero es un sistema desarrollado en la UCI para proporcionar un espacio donde las personas del mundo puedan intercambiar experiencias y conocimientos; probar, mejorar y compartir habilidades en la resolución de

problemas matemáticos y de agilidad mental; así como convertirse en una opción de entretenimiento inteligente (UCI, Caribbean Mind Forge, 2012). Se analiza este sistema con el objetivo de tomar experiencias positivas acerca de cómo realizar las competencias en línea.

Además, se detectan ocho (8) OA presentes en el ROA de la UCI que se analizan a continuación:

- Ejercicios resueltos sobre Teoría combinatoria, Entrevista a Alan Turing, Introducción a las deducciones lógicas, Sistema numérico binario, Surgimiento de la Teoría de grafos, son cinco (5) de los OA que se encuentran en el ROA; los mismos no permiten experimentación ni interactividad, más bien están dirigidos a la presentación de información a través de documentos, presentaciones electrónicas, videos e imágenes.
- Mapas de Karnaugh, Operaciones con conjuntos y Ordenamiento topológico son los restantes OA que se encuentran en el ROA; ellos a un pequeño nivel permiten la interactividad, pero están dedicados solo a la evaluación de determinados ejercicios en cuanto a preguntas de verdadero y falso y preguntas de marcas la(s) respuesta(s) correcta(s).

En el ámbito internacional también se encuentran diferentes *applets*⁶ relacionados con temas de la MD, según (Herrera, 2012) algunos ejemplos de ellos son:

- ***Turing Machine Applet de Whartman Associates:*** 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.
- ***Turing Machine Applet de la Universidad de Canterbury:*** simula una Máquina de Turing que está diseñada para copiar una cadena que contiene N números 1. Se detiene dejando 2N números 1 en la cinta.
- **Simulador de Grafos de la Facultad de Informática de la Universidad Politécnica de Madrid:** consta de dos fases:
 - Construir el grafo al que se quiere aplicar el algoritmo.
 - Resolución del algoritmo.

⁶ 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, 2000).

- **Simulador de diagramas de Venn:** Posibilita realizar operaciones entre conjuntos.

Además (Carballo, 2014) menciona otras aplicaciones vinculadas también a resolver determinados problemas de la MD:

- **LEMAT (Libro Electrónico de Matemáticas):** es un conjunto de materiales multimedia elaborado con el propósito de proporcionar al usuario una gran interactividad en el proceso de aprendizaje de las Matemáticas. Además, las herramientas de autoevaluación y evaluación que incluye permiten al alumno saber cuándo ha superado un hito determinado en su formación.
- **Applets Java de Matemáticas:** Es una colección de *applets* de Matemáticas de distintos niveles, incluyendo algunos en 3 dimensiones. Los temas son de Geometría elemental, Geometría esférica, Trigonometría, Vectores 3D, Análisis y Números Complejos.
- **Cut the knot:** Cientos de *applets* de juegos, misceláneas, curiosidades, pruebas de Matemáticas en general clasificados por temas.
- En el sitio web del Departamento de Matemática Aplicada de España se pueden encontrar algunas aplicaciones en Java sobre MD. Estas abordan los contenidos de Aritmética entera y modular, Grafos y Álgebras de Boole.

Como se puede constatar los sistemas internacionales anteriormente analizados resuelven determinados problemas, pero presentan un bajo nivel de interactividad y experimentación, por lo que es la aplicación la que brinda una serie de informaciones definidas por defecto; además de estar dedicados a determinados temas en específico.

Derivado del análisis anterior de los sistemas tanto nacionales como internacionales, se puede comprobar que, a pesar de resolver determinados problemas, aún les faltan muchos elementos para llegar a ser lo que verdaderamente se espera de un OA-IE acorde con los objetivos que persigue en los diferentes contextos educativos.

1.3 Metodologías y herramientas para el desarrollo del Sistema Web Interactivo - Experimental como apoyo a la docencia.

Se puede concebir a un modelo pedagógico como la base de la metodología para el diseño de un OA; esta idea se deriva de lo expresado por (Mariana, 2008) al definir a un modelo pedagógico como una forma de concebir la práctica de los procesos formativos en una institución de educación superior. Comprende los procesos relativos a las cuestiones pedagógicas de cómo se aprende, cómo se enseña, las metodologías

más adecuadas para la asimilación significativa de los conocimientos, habilidades y valores, las consideraciones epistemológicas⁷ en torno a la pedagogía, las aplicaciones didácticas⁸, el currículo y la evaluación de los aprendizajes.

No obstante, la presente investigación está dirigida al desarrollo de un OA-IE haciendo uso de las TIC, específicamente de tecnologías web, por lo que se puede considerar como un producto de software que responde a un modelo de ciclo de vida del proceso de desarrollo de software. Dichos modelos están basados en la naturaleza del proyecto y de la aplicación, así como en los métodos y herramientas que serán utilizados en el proyecto, los controles y los productos o servicios desarrollados (Pressman, 2010).

Haciendo un uso adecuado de las metodologías de desarrollo de software es posible lograr un software de calidad debido a que las metodologías están presentes en todo el ciclo de vida del software y (García W. Y., 2009) las define como un conjunto de procedimientos, técnicas, herramientas y soporte documental que permite a los desarrolladores realizar un nuevo software.

Existen varias clasificaciones para estas metodologías dentro de las cuales se destacan dos (Figuroa, Solís, & Cabrera, 2007):

- **Metodologías tradicionales:** También conocidas como metodologías formales están enfocadas en una documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.). Esto trae consigo que la utilización de esta metodología no sea una buena solución para proyectos que se desarrollen sobre entornos volátiles, ya que realizar un cambio en el proyecto implica elevados costos. En esta metodología se destacan *RUP (Rational Unified Process)* y *MSF (Microsoft Solution Framework)*.
- **Metodologías ágiles:** Nace para dar respuesta a los problemas derivados de la tradicional. Esta metodología esta principalmente basada en los siguientes aspectos:
 - Asume que los individuos y las interacciones entre ellos son más importantes que las herramientas y procesos empleados

⁷ Parte de la filosofía que estudia los principios, fundamentos, extensión y métodos del conocimiento humano.

⁸ Parte de la pedagogía que estudia las técnicas y métodos de enseñanza.

- Crear un producto de software que funcione es más importante que escribir una documentación exhaustiva.
- Debe existir una estrecha colaboración entre el cliente y los desarrolladores del producto.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Como ejemplo de estas metodologías se destacan *XP (Extreme Programming)*, *AUP (Agil Unified Process)*, *SCRUM*.

Para la presente investigación se asumirá una metodología ágil con el fin de aprovechar las facilidades que brinda en cuanto a la documentación y el entorno cambiante que tiene el proyecto; pues a raíz de que el proyecto se lleva a cabo por un solo desarrollador es más importante centrarse en la correcta obtención del sistema sin importar cuán documentado esté, y además, es seleccionada por el tiempo de desarrollo del sistema y la estrecha relación cliente – desarrollador para lograr un proyecto flexible a los cambios que ocurran durante el proceso de desarrollo de software. Para interpretar mejor lo expresado anteriormente puede observarse la siguiente tabla comparativa entre ambos enfoques.

Tabla 2: Comparación entre las metodologías tradicionales y ágiles. (Figueroa, Solís, & Cabrera, 2007)

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.

Luego de haber seleccionado la metodología ágil para el desarrollo del OA-IE, a continuación, se realiza un análisis de varias de ellas:

EXTREMA PROGRAMMING (XP).

Dentro de los procesos ágiles, este es el que más se destaca. Su mayor diferencia con las metodologías tradicionales es que se enfatiza más en la adaptabilidad que en la previsibilidad, por lo que un cambio de requisitos sobre la marcha es algo natural e inevitable, por lo que el equipo de desarrollo debe ser capaz de adaptarse a los cambios en cualquier punto de la vida del proyecto en vez de intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos mayores en controlarlos a lo largo del ciclo de vida.

Sus principales características son:

- **Desarrollo iterativo e incremental:** se trata de ir realizando pequeñas mejoras una tras otras.
- **Frecuente interacción del equipo de programación con el cliente o usuario:** es recomendado que el cliente o al menos un representante suyo trabaje junto al equipo de desarrollo.
- Deben corregirse todos los errores antes de añadir nuevas funcionalidades y hacer entregas frecuentes.
- **Refactorización del código:** se trata de reescribir ciertas partes del código para aumentar la legibilidad y mantenibilidad sin modificar su comportamiento. Con las pruebas se verifica que las refactorizaciones no introduzcan algún fallo.
- **Propiedad del código compartida:** todo el personal del equipo puede corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores sean 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.

Según (Bolivariana, 2009) las principales desventajas de esta metodología son que es recomendable emplearla solo en proyectos a corto plazo y los altos costos en caso de fallar.

AGILE UNIFIED PROCESS (AUP).

Es un acercamiento aerodinámico al desarrollo del software basado en el Proceso Unificado Racional (*RUP* por sus siglas en inglés) de la *International Business Machines Corp. (IBM)*, basado en disciplinas y entregas incrementales con el tiempo. El ciclo de vida en proyectos grandes es serial mientras que en los pequeños es iterativo. Las disciplinas de *AUP* según (Salvador, 2015) son:

- **Modelado:** Entender el negocio de la organización, el problema de dominio que se aborda en el proyecto, y determinar una solución viable para resolver el problema de dominio.
- **Implementación:** Transformar el modelo en código ejecutable y realizar un nivel básico de pruebas individuales.
- **Prueba:** Realizar una evaluación objetiva para garantizar la calidad. Incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido y verificar que se cumplan los requisitos.
- **Despliegue:** Realizar un plan para la presentación del sistema y ejecutarlo para hacer que el sistema se encuentre a disposición de los usuarios finales.
- **Administración de la configuración:** Realizar la gestión de acceso a los artefactos de su proyecto. Incluye el seguimiento de las versiones del artefacto en el tiempo y la gestión y control de los cambios.
- **Administración o gerencia del Proyecto:** Dirigir las actividades que se llevan a cabo en el proyecto. Incluye la gestión de riesgos, dirección de personas, garantizar la entrega a tiempo y con el presupuesto asignado.
- **Entorno:** Garantizar que las orientaciones y herramientas estén disponibles para el equipo en cualquier momento que ellos lo necesiten.

Sus principales características son:

- Iterativo e incremental.
- Manejo de los casos de uso.
- Centrado en la arquitectura.
- Enfocado en los riesgos.

SCRUM.

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo del software. El desarrollo es iterativo e incremental donde cada ciclo culmina con una pieza de software ejecutable que incorpora nueva funcionalidad. Por lo general cada ciclo tiene una duración de entre 2 y 4 semanas.

Scrum está mayormente enfocado en priorizar el trabajo en función del valor que tenga para el negocio, diseñado para adaptarse a los cambios en los requerimientos. Su objetivo principal es entregar un software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

El equipo de desarrollo se focaliza en construir un software de calidad, para ello se define las características del producto (qué construir, qué no construir y en qué orden construir las), además debe removerse cualquier obstáculo que entorpezca el desarrollo para lograr que el equipo sea lo más efectivo y productivo posible.

Scrum tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa.

A continuación, se muestran las ventajas que posee esta metodología:

- Se obtiene software lo más rápido posible y este cumple con los requerimientos importantes.
- El trabajo es en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.
- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva.
- Las reuniones se dedican a inconvenientes recientes, evitando así el estancamiento.

Metodologías para el desarrollo de OA.

METODOLOGÍA DE DESARROLLO DE SISTEMAS INSTRUCCIONALES (ISD-MeLO).

Fue creada por Lúcia Blondet Baruque y Rubens Nascimento Melo en el año 2004, esta metodología está basada en OA y en el Modelo de Diseño Instruccional ADDIE.

ISD-MeLO posee 5 fases:

- **Análisis:** para determinar el problema y el perfil del aprendiz, incluso considerar la aplicación de modelos de estilos de aprendizaje. Genera los siguientes productos: formularios de análisis de perfil del aprendiz, análisis del problema, de análisis ambiental, así como los OA existentes. De igual forma, esta fase considera los metadatos del OA, como parte fundamental para lograr su catalogación y reutilización.

- **Diseño del OA:** referido al contenido instructivo y el *look and feel*⁹ de la interfaz del OA. Se generan los siguientes productos: documentos de análisis de tarea y de análisis del contenido, secuenciamiento del OA (mapa conceptual), *metadata* y *storyboards*¹⁰ de diseño de la interfaz.
- **Desarrollo del OA:** destinada a producir el OA y almacenarlo en un repositorio.
- **Puesta en práctica:** donde se utiliza el OA desarrollado. En esta fase se debe tener el OA almacenado y poder utilizarlo en un LMS o una página web, tener un plan para la entrega de la instrucción y uso del OA.
- **Evaluación del OA:** destinada a medir la adecuación y la eficacia de la instrucción ofrecida con el OA. Genera los siguientes productos: ajustes o eliminación del OA del repositorio, verificación si la instrucción está satisfaciendo objetivos del aprendizaje.

(Silva, Ponce, & Hernández, 2013) expresan que esta metodología no relaciona aspectos de diseño del OA al análisis de los estilos analizados, por lo que separa el perfil del estudiante desde el punto de vista de su estilo de aprendizaje del diseño desde el aspecto pedagógico del OA, sin embargo, recomienda analizar estilos de aprendizajes de los estudiantes.

METODOLOGÍA AODDEI (Silva, Ponce, & Hernández, 2013).

Es una metodología basada en el Modelo *ADDIE* que sirve de guía en la parte pedagógica del desarrollo de OA. Incorpora a docentes (autores de OA), docentes o alumnos (usuarios que utilizarán el OA ya sea para aprender de ellos o generar nuevos), técnico (diseñadores de páginas web y concedores de cuestiones técnicas computacionales) y grupos de expertos (evaluadores de los OA).

AODDEI integra cinco (5) fases principales:

1. **Análisis y Obtención:** incluye el análisis de necesidades, la obtención de materiales y contenidos, la digitalización de materiales y las competencias a alcanzar. Las competencias consideradas son: habilidades de pensamiento, habilidades comunicacionales, producir textos escritos con redacción clara y estructurada, y construir argumentos lógicos para exponer ideas.

⁹ Metáfora utilizada dentro del entorno de *marketing* para poder dar una imagen única a los productos.

¹⁰ En español significa guión gráfico.

2. **Diseño del OA:** incluye el armado y estructuración del OA (objetivo, contenido informativo, actividades, evaluación del aprendizaje y metadatos).
3. **Desarrollo del OA:** incluye el armado, el empaquetado y el almacenamiento del OA en un repositorio temporal.
4. **Evaluación del OA:** incluye la evaluar propia del OA por parte de los expertos y el almacenamiento en un repositorio permanente.
5. **Implantación:** incluye la integración del OA a un LMS.

Esta metodología permite considerar un Diseño Instruccional, así como la concepción del OA desde la misma necesidad de la instrucción, y posteriormente, en la fase 4, evaluar los contenidos, el diseño y los metadatos del OA, sin considerar la evaluación de la adecuación de los contenidos y actividades a los estudiantes que utilizarán el OA y que aprenderán interactuando con estos recursos.

METODOLOGÍA LEARNING OBJECTS CONSTRUCTION METHODOLOGY (LOCoME) (Silva, Ponce, & Hernández, 2013).

Propuesta por Manuel Medina y María Gertrudis López; basa el diseño en estándares y mecanismos que guíen la construcción de los objetos de software tomando como modelo el uso de la metodología *RUP*.

Utilizar *RUP* permite aprovechar las fortalezas necesarias para el desarrollo de proyectos de software, e incorporar aquellas condiciones adicionales que se requieren para construir OA con altos niveles de calidad sistémica, como son aspectos pedagógicos e instruccionales necesarios.

LOCoME contempla 4 fases dentro del ciclo iterativo del desarrollo, e incluye en cada una de ellas: la descripción de objetivos a alcanzar, los artefactos a utilizar y los criterios de evaluación que deben ser considerados.

1. **Análisis del OA:** fase temprana del desarrollo, donde se establece la visión, pertinencia, metáforas y características detalladas del OA. Esto permite establecer los requerimientos perseguidos con el desarrollo, tanto a nivel conceptual como funcional.
2. **Diseño Conceptual del OA:** donde se establece la “forma” mediante la cual va a ser desarrollado el OA, independientemente de la plataforma de desarrollo a ser empleada. Se diseñan modelos informales de datos y de aplicación, que permiten especificar claramente las características y disposición ideal de cada uno de los sub-objetos que conformarían el OA.
3. **Construcción del OA:** contempla la implementación del OA, y a diferencia de *RUP* abarca dos subfases: desarrollo de los recursos y adecuación al estándar del OA. La adecuación al estándar

incluye: la generación de metadata y del *SCORM CAM (SCORM Content Aggregation Model)*, el empaquetado y la visualización del OA.

4. **Evaluación Pedagógica:** determinar si las características del enfoque educativo seleccionado se están cumpliendo satisfactoriamente. *LOCoME* incorpora el uso de metáforas para los contenidos, y concentra los aspectos pedagógicos a enfoques educativos, sin considerar los estudiantes que luego utilizaran los OA desarrollados.

Por sí solas las metodologías de software analizadas no son factibles para cumplir con los objetivos de esta investigación. Es por esto que como propone (García A. , 2014) es aconsejable utilizar *XP* como metodología de desarrollo de software, que es una metodología fácil de implementar, con una documentación ligera, flexible y eficiente por lo que es conveniente utilizarla para guiar el proceso de desarrollo, al ser los OA-IE pequeños proyectos de desarrollo de software a corto plazo de tiempo. Por otra parte, se decide utilizar la *ISD-MeLO* como metodología para guiar el proceso de desarrollo de los OA-IE desde el punto de vista pedagógico, al ser flexible para aplicarse en diversas situaciones de aprendizaje.

1.3.5 Tecnologías, herramientas y lenguajes a utilizar para el desarrollo del Objeto de Aprendizaje Interactivo - experimental.

LENGUAJE DE MODELADO.

El modelado constituye una simplificación de la realidad donde se define lo esencial para la construcción del software con los objetivos de comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que se está desarrollando y descubrir oportunidades de simplificación y reutilización (Booch, Rumbaugh, & Jacobson, 2004).

Para la presente investigación se propone utilizar el Lenguaje Unificado de Modelado (*UML* por sus siglas en inglés) en su versión 2.0, que consiste en un lenguaje diseñado para visualizar, especificar, construir y documentar software orientados a objetos (Booch, Rumbaugh, & Jacobson, 2004).

Las ventajas del lenguaje *UML* según (López & Ruiz, 2011) son:

- Es estándar, lo que significa que facilita la comunicación.
- Está basado en el meta modelo con una semántica bien definida.
- Se basa en una notación gráfica concisa y fácil de aprender y utilizar.

- Se puede utilizar para modelar software en distintos dominios: sistemas de información empresariales, sistemas Web, sistemas críticos y de tiempo real, etc.
- Es fácilmente extensible.

HERRAMIENTA DE MODELADO.

Dentro de las principales herramientas para el modelado se encuentran herramientas de Ingeniería de Software Asistida por Ordenadores (por sus siglas en inglés *CASE*) definidas por (Terry & Logee, 1990) como herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento).

Visual Paradigm para *UML* es una herramienta para desarrollo de aplicaciones utilizando modelado *UML* ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Se propone utilizar esta herramienta en su versión 8 que brinda las siguientes ventajas según (Galindos, 2013):

- **Dibujo:** Facilita el modelado de *UML*, proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que se ajusta al estándar soportado por la herramienta.
- **Corrección sintáctica:** Controla que el modelado sea correcto
- **Coherencia entre diagramas:** Dispone de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, esto evita las duplicidades.
- **Integración con otras aplicaciones:** Permite integrarse con otras aplicaciones como herramientas ofimáticas, lo cual aumenta la productividad.
- **Trabajo multiusuario:** Permite el trabajo en grupo, proporciona herramientas de compartición de trabajo.
- **Reutilización:** Dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- **Generación de código:** Permite generar código de forma automática, esto reduce los tiempos de desarrollo y evita errores en la codificación del software.
- **Generación de informes:** Permite generar diversos informes a partir de la información introducida en la herramienta.

LENGUAJE DE PROGRAMACIÓN.

Un lenguaje de programación es un conjunto de reglas o normas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador (sin ambigüedades) (Ureña, 2011). Se pueden usar para crear programas que controlen el comportamiento físico y lógico de una máquina. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Peralta & Durán, 2014).

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Para la presente investigación se propone utilizar Python 2.7.11 debido a que según (Dayley, 2007) este lenguaje posee las siguientes ventajas:

- **Propósito general:** Se pueden crear distintos tipos de programas.
- **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible siempre y cuando exista un intérprete programado para este lenguaje.
- **Interpretado:** No se debe compilar el código antes de su ejecución. En realidad, sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador.
- **Interactivo:** Dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables. Además, Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de *strings*¹¹, números, archivos, etc. Además, existen muchas librerías que se pueden importar en los programas para tratar temas específicos.
- **Sintaxis clara:** Tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.
- **Mixto:** Se puede integrar de manera "fácil" con otros lenguajes de programación.
- **Gratuito:** Una ventaja fundamental de Python es la gratuidad de su intérprete, se puede descargar desde la página web: <http://www.python.org>.

MARCO DE TRABAJO.

Un marco de trabajo o *framework* no solo es utilizado en el ámbito de aplicaciones web, se pueden encontrar *frameworks* para el desarrollo de aplicaciones médicas, de visión de computador, para el desarrollo de juegos,...Por lo que al referirse a este término se hace alusión a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación (Gutiérrez, 2006).

Para el desarrollo del OA-IE se propone utilizar Django 1.9 que es un marco de trabajo de desarrollo web totalmente implementado sobre Python, con el que se pueden crear y mantener aplicaciones de alta calidad. Incluye un servidor web ligero que se puede usar mientras se desarrolla. Al mismo tiempo, Django permite trabajar fuera de su ámbito según sea necesario. Django ofrece las siguientes facilidades: (Holovaty & Kaplan-Moss, 2009)

- Sistema de plantillas para separar la presentación de un documento de sus datos.
- Construcción automática de interfaces de administración.
- Vistas genéricas que recogen ciertos estilos y patrones comunes en su desarrollo y los abstraen, de modo que se puede escribir rápidamente vistas comunes de datos sin tener que escribir mucho código.
- Sistema de caché robusto y con un nivel de granularidad ajustable, que permite guardar páginas dinámicas para que no tengan que ser recalculadas cada vez que se piden.

¹¹ Cadenas de caracteres.

- Integración con bases de datos y aplicaciones existentes.
- Construcción de aplicaciones multilenguaje permitiendo especificar cadenas de traducción de más de cuarenta idiomas.

SISTEMA GESTOR DE BASES DE DATOS (SGBD).

PostgreSQL es un SGBD que incorpora el modelo relacional para sus bases de datos y es compatible con el lenguaje de consulta *Structured Query Language (SQL)*, que significa lenguaje de consulta estructurado. Es ampliamente considerado como una de las alternativas de sistemas de BD de código abierto.

Para la gestión de los datos en la presente investigación se propone utilizar PostgreSQL 9.4 debido a las siguientes ventajas que posee sobre otros gestores según (Rodríguez, 2010):

- Instalación ilimitada.
- Es un sistema multiplataforma.
- Diseñado para ambientes de alto volumen de datos.
- Existen varias herramientas gráficas de alta calidad para administrar las BD (*pgAdmin, pgAccess*) y para hacer diseño de bases de datos (*Tora, Data Architect*).

ENTORNO INTEGRADO DE DESARROLLO (IDE).

Para la implementación del OA-IE se propone utilizar PyCharm 4.5.4 que es un IDE basado en IntelliJ IDEA que ofrece las siguientes funciones: (PyCharm, 2015)

- Auto-completamiento de código.
- Señalamiento de errores con soluciones fáciles.
- Posibilita una fácil navegación para proyectos y código.
- Mantiene el código bajo control de chequeos, asistencia de pruebas, refactorizaciones y un conjunto de inspecciones que posibilitan codificar de forma limpia y sostenible.

SERVIDOR WEB.

Se propone utilizar el Servidor Web Apache 2.2. Apache según (Mifsuf, 2013) es un servidor web de software libre desarrollado por *Apache Software Foundation (ASF)*. Desde 1996 es el servidor más utilizado en Internet y es el utilizado en sistemas GNU/Linux. Las ventajas que brinda Apache son:

- Es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores con los que dispone.
- Es estable (impide caídas o cambios en el servidor inesperados), flexible y eficiente (es capaz de trabajar con el estándar HTTP P/1.1 (RFC2616) y con la mayor parte de las extensiones web que existen en la actualidad: PHP, SSL, Python...), extensible (dispone de una elevada cantidad de módulos que amplían su funcionalidad) y multiplataforma (está disponible para diferentes plataformas como GNU/Linux, Window, MacOS).

Conclusiones parciales del capítulo.

En este capítulo se evidenció la importancia que posee el estudio de la MD para la formación de los estudiantes de la ICI como futuros profesionales de la informática; además de cómo a través de la utilización de las TIC, específicamente de los sistemas web interactivos - experimentales se puede apoyar al PEA y brindar un recurso educativo capaz de mantener una adecuada interacción y retroalimentación que posibilite su utilización en distintos contextos educativos.

El estudio realizado a los sistemas homólogos existentes permitió constatar principalmente que los OA que se encuentran publicados en la UCI no posibilitan interacción, experimentación ni evaluación como verdaderamente se espera de un OA-IE.

Luego de haber analizado todos estos elementos, el autor realizó un estudio de las herramientas, tecnologías y metodologías a utilizar que lo llevó a asumir como metodología de desarrollo un híbrido entre *ISD-MeLO* y *XP*, como lenguaje de modelado al Lenguaje Unificado de Modelado con la herramienta *Visual Paradigm*, para la implementación Python como lenguaje de programación sobre el *framework* Django apoyado en el *IDE* PyCharm y para la gestión de los datos PostgreSQL.

CAPÍTULO 2: Análisis y diseño del Sistema Interactivo - Experimental.

El objetivo de este capítulo es presentar los resultados que se obtuvieron una vez cumplidas las fases de Análisis y Diseño que propone la metodología seleccionada. Se realiza un análisis de los contenidos de la MD que serán incluidos en el sistema, así como la estructura del OA-IE según los criterios pedagógicos, se especifican los requisitos funcionales y no funcionales y se presentan los artefactos que se generaron para dar solución al problema planteado.

2.1 Análisis.

En esta fase se estudia el análisis del perfil del aprendiz de (García A. , 2014) para así realizar un estudio de los contenidos que contendrá el OA-IE propuesto y se presenta el modelo de dominio con todos los conceptos relacionados.

2.1.1 Análisis de los contenidos de la MD que contendrá el OA-IE a desarrollar.

Primeramente, para definir los temas de la MD que serán incluidos en el sistema se aplicó una encuesta (Ver [ANEXO #1 y 2](#)) atendiendo a los indicadores definidos por (García A., 2014) (Ver [ANEXO #3](#)), la encuesta fue aplicada a un total de diez (10) profesores de la asignatura de la Facultad Introdutoria en Ciencias Informáticas (FICI), cincuenta (50) estudiantes del actual segundo año y cien (100) de la actual FICI.

Al analizar los resultados de la encuesta (Ver [ANEXO #4](#)) se determinó que los contenidos a incluir serán la Teoría de conjuntos, Relaciones binarias, Máquinas de Turing, Teoría combinatoria, Relaciones de recurrencia y Teoría de grafos.

1. **Teoría de conjuntos:** Los resultados asociados a este tema pudieran ser mejorados si se lograra una mejor comprensión y ejercitación de los contenidos, donde se presenten una variedad de ejercicios en cuanto a forma y complejidad. Además, permite grandes posibilidades de experimentación donde se le permita al usuario experimentar con los siguientes contenidos de la Teoría de conjuntos: resolución de ecuaciones entre conjuntos, intersección, unión, diferencia, diferencia simétrica, conjunto potencia, producto cartesiano, cardinalidad, complemento, partición, subconjunto, subconjunto propio, conjuntos disjuntos, equivalentes, iguales, comparables y comprobar que un elemento pertenece a un conjunto o no.
2. **Relaciones binarias:** Sería factible contar con un sistema donde los estudiantes puedan experimentar con los siguientes contenidos de las relaciones binarias: dominio, imagen, inversa, compuesta, comprobar si una relación es función, función total, inyectiva, biyectiva, sobreyectiva,

reflexiva, irreflexiva, simétrica, asimétrica, antisimétrica, transitiva, de equivalencia, de orden, de orden total, determinar las clases de equivalencia y conjunto cociente en caso que sea un una relación de equivalencia.

3. **Máquinas de Turing:** Es uno de los temas de mayor dificultad para los estudiantes. Para una adecuada comprensión de este tema el principal aspecto es permitir que los estudiantes experimenten con las máquinas de Turing, donde puedan observar su representación gráfica además de comprobar que la máquina reconozca ciertas cadenas o condiciones especificadas por el mismo usuario.
4. **Teoría combinatoria:** Igualmente es un tema complicado dentro de la MD por lo que se debería permitir a los usuarios experimentar con los siguientes contenidos: permutaciones, combinaciones, permutaciones generalizadas, combinaciones generalizadas, determinar el coeficiente de un término, descomponer un polinomio y el principio de las casillas.
5. **Relaciones de recurrencia:** Teniendo en cuenta el tiempo para el desarrollo del sistema y la complejidad para resolver funciones de recurrencia, al menos sería importante que el sistema permitiera a los usuarios determinar si una serie numérica se ajusta a una determinada función de recurrencia y clasificar una función de recurrencia en cuanto a su linealidad, homogeneidad, coeficientes y orden.
6. **Teoría de grafos:** Es un tema donde es importante la representación gráfica por lo que el sistema debe permitirlo, además de experimentar con las siguientes operaciones: determinar el grado de un vértice, si una secuencia de vértices es un camino y/o cadena cerrada de Euler, determinar si un vértice o una arista es de corte, si el grafo es simple, circular, rueda, regular, bipartido, completo, bipartido completo, conexo, contiene una cadena de Euler, cadena cerrada de Euler, un ciclo de Hamilton o un camino de Hamilton.

2.1.2 Modelo de dominio.

Se encarga de representar los conceptos más importantes y significativos en el desarrollo de un sistema, siguiendo como objetivo fundamental definir las interrelaciones de los objetos más importantes, es representado mediante clases.

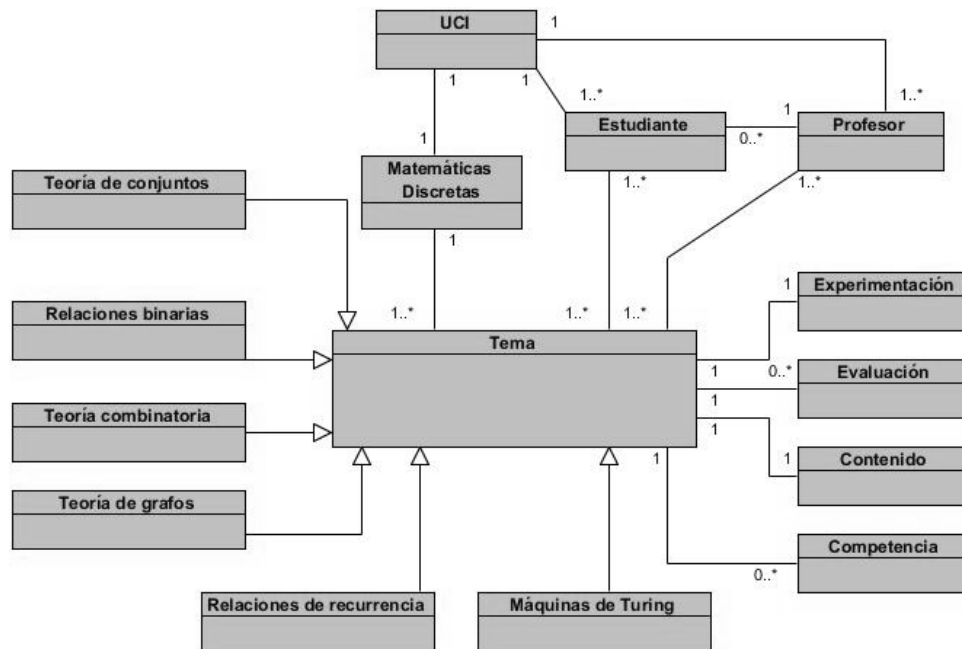


Ilustración 2: Modelo de dominio (Elaboración propia).

2.2 Diseño.

En la presente fase se encuentran definidas la unidad de información, la unidad didáctica y la unidad de evaluación y retroalimentación. Además, se define la lista de funcionalidad y requisitos no funcionales (RNF), las historias de usuario (HU) y las Tarjetas CRC.

2.2.1 Unidad de información.

Esta unidad está conformada por todos los elementos teóricos (definiciones, conceptos, ejemplos resueltos, etc.) relacionados con los temas incluidos en el sistema, de esta forma el usuario podrá informarse y tomar sus apuntes sobre los contenidos aclarando así sus dudas. Viéndolo desde el punto de vista de diseño es el resultado de la elección del contenido y la forma de presentárselos al usuario final por lo que debe cumplir con los criterios de accesibilidad. En el OA-IE que se presenta, esta unidad estará conformada por la propuesta de libro “Matemáticas Discretas para Ingenieros en Ciencias Informáticas” del MSc. Alién García Hernández.

El diseño de esta unidad puede ser común para los OA-IE porque independientemente de que los temas varíen, la forma de presentar la información y navegar en ella puede realizarse de forma similar en todos

los OA-IE ya que se presenta el libro en formato “.pdf” embebido en la misma página dando la opción además de descargarlo.

2.2.2 Unidad de experimentación.

Es una unidad orientada a la experimentación con los contenidos, de esta forma los estudiantes podrán reflexionar acerca de los elementos que ya adquirió en la unidad teórica. Para la confección de esta unidad se permite que los estudiantes sean los gestores de su conocimiento dándole la opción de que él escoja con qué desea experimentar, defina los datos a utilizar y realice las operaciones sobre ellos, una vez que el usuario desee comprobar las operaciones respondidas el sistema debe ser capaz de evaluarlo y emitirle una respuesta donde se le diga cuáles respondió bien, cuáles mal y en el caso de las mal respondidas debe mostrarle la respuesta correcta.

2.2.3 Unidad de evaluación y retroalimentación.

Permite comprobar lo que los estudiantes han aprendido, para ello se utiliza como criterio de evaluación el sistema de puntuación de 2, 3, 4 y 5, tomando una evaluación de dos (2) puntos cuando se haya respondido correctamente a menos del 60%, tres (3) a un total de entre un 60 y 80%, cuatro (4) entre 80 y 90% y cinco (5) puntos por encima del 90% de los ejercicios propuestos.

Las evaluaciones están divididas en dos (2) categorías: auto-evaluaciones y evaluaciones orientadas por el profesor. Las auto-evaluaciones consisten en responder una serie de ejercicios, de los cuales los resultados no serán reflejados en el currículo del estudiante, sino que solo es una forma de que el estudiante pueda medir hasta qué punto se encuentra preparado en los distintos temas de la asignatura y las orientadas por el profesor si reflejan una calificación para el currículo de desempeño del estudiante, los profesores encargados de cada estudiante son los encargados de controlar dicho currículo; atendiendo a que el sistema emite una nota según el porcentaje de respuestas correctas el profesor en correspondencia con las respuestas del estudiante está facultado para modificar su nota. Además, los profesores son los encargados de la gestión de las evaluaciones.

En esta unidad se incluye además las competencias, donde opcionalmente los estudiantes que lo deseen pueden resolver ejercicios con carácter competitivo y se mantendrá un registro con el *ranking* de los usuarios atendiendo a la cantidad de ejercicios respondidos y la puntuación que emita cada uno de acuerdo a su complejidad.

2.2.4 Funcionalidades del sistema.

Para definir las funcionalidades del sistema se tuvo en cuenta la unidad teórica, didáctica y de evaluación y retroalimentación de los cinco (6) contenidos de la MD a tratar, además de determinados elementos de gestión necesarios para un adecuado funcionamiento del sistema y dar cumplimiento al objetivo planteado. A continuación, se expone la relación con las ochenta y seis (86) funcionalidades definidas para el OA-IE.

Tabla 3: Relación de funcionalidades definidas para el sistema (Elaboración propia).

Funcionalidad	
F1: Experimentar con el tema “Teoría de conjuntos”.	F2: Exponer el contenido teórico tema “Teoría de conjuntos”.
F3: Realizar evaluaciones orientadas por el profesor del tema “Teoría de conjuntos”.	F4: Realizar autoevaluaciones del tema “Teoría de conjuntos”.
F5: Listar ejercicios del tema “Teoría de conjuntos”.	F6: Adicionar ejercicios del tema “Teoría de conjuntos”.
F7: Eliminar ejercicios del tema “Teoría de conjuntos”.	F8: Deshabilitar ejercicios del tema “Teoría de conjuntos”.
F9: Habilitar ejercicios del tema “Teoría de conjuntos”.	F10: Experimentar con el tema “Relaciones binarias”.
F11: Exponer el contenido teórico del tema “Relaciones binarias”.	F12: Realizar evaluaciones orientadas por el profesor del tema “Relaciones binarias”.
F13: Realizar autoevaluaciones del tema “Relaciones binarias”.	F14: Listar ejercicios del tema “Relaciones binarias”.
F15: Adicionar ejercicios del tema “Relaciones binarias”.	F16: Eliminar ejercicios del tema “Relaciones binarias”.
F17: Deshabilitar ejercicios del tema “Relaciones binarias”.	F18: Habilitar ejercicios del tema “Relaciones binarias”.
F19: Experimentar con el tema “Teoría combinatoria”.	F20: Exponer el contenido teórico del tema “Teoría combinatoria”.
F21: Realizar evaluaciones orientadas por el profesor del tema “Teoría combinatoria”.	F22: Realizar autoevaluaciones del tema “Teoría combinatoria”.
F23: Listar ejercicios del tema “Teoría combinatoria”.	F24: Adicionar ejercicios del tema “Teoría combinatoria”.

F25: Eliminar ejercicios del tema “Teoría combinatoria”.	F26: Deshabilitar ejercicios del tema “Teoría combinatoria”.
F27: Habilitar ejercicios del tema “Teoría combinatoria”.	F28: Experimentar con el tema “Máquinas de Turing”.
F29: Exponer el contenido teórico del tema “Máquinas de Turing”.	F30: Realizar evaluaciones orientadas por el profesor del tema “Máquinas de Turing”.
F31: Realizar autoevaluaciones del tema “Máquinas de Turing”.	F32: Listar ejercicios del tema “Máquinas de Turing”.
F33: Adicionar ejercicios del tema “Máquinas de Turing”.	F34: Eliminar ejercicios del tema “Máquinas de Turing”.
F35: Deshabilitar ejercicios del tema “Máquinas de Turing”.	F36: Habilitar ejercicios del tema “Máquinas de Turing”.
F37: Representar Máquinas de Turing gráficamente.	F38: Experimentar con el tema “Teoría de grafos”.
F39: Exponer el contenido teórico del tema “Teoría de grafos”.	F40: Realizar evaluaciones orientadas por el profesor del tema “Teoría de grafos”.
F41: Realizar autoevaluaciones del tema “Teoría de grafos”.	F42: Listar ejercicios del tema “Teoría de grafos”.
F43: Adicionar ejercicios del tema “Teoría de grafos”.	F44: Eliminar ejercicios del tema “Teoría de grafos”.
F45: Deshabilitar ejercicios del tema “Teoría de grafos”.	F46: Habilitar ejercicios del tema “Teoría de grafos”.
F47: Representar grafos gráficamente.	F48: Experimentar con el tema “Relaciones de recurrencia”.
F49: Exponer el contenido teórico del tema “Relaciones de recurrencia”.	F50: Realizar evaluaciones orientadas por el profesor del tema “Relaciones de recurrencia”.
F51: Realizar autoevaluaciones del tema “Relaciones de recurrencia”.	F52: Listar ejercicios del tema “Relaciones de recurrencia”.
F53: Adicionar ejercicios del tema “Relaciones de recurrencia”.	F54: Eliminar ejercicios del tema “Relaciones de recurrencia”.
F55: Deshabilitar ejercicios del tema “Relaciones de recurrencia”.	F56: Habilitar ejercicios del tema “Relaciones de recurrencia”.

F57: Mostrar ayuda en las experimentaciones.	F58: Adicionar ejercicios competitivos.
F59: Eliminar ejercicios competitivos.	F60: Listar ejercicios competitivos.
F61: Deshabilitar ejercicios competitivos.	F62: Habilitar ejercicios competitivos.
F63: Resolver ejercicios competitivos.	F64: Mostrar <i>ranking</i> de usuarios según su desempeño en las competencias.
F65: Comparar usuarios según su desempeño en las competencias.	F66: Autenticar usuarios con el dominio uci.cu.
F67: Mostrar perfil de usuario.	F68: Adicionar usuarios locales con privilegios administrativos.
F69: Editar usuarios locales con privilegios administrativos.	F70: Eliminar usuarios locales con privilegios administrativos.
F71: Deshabilitar usuarios.	F72: Habilitar usuarios.
F75: Adicionar grupos docentes.	F76: Editar grupos docentes.
F77: Eliminar grupos docentes.	F78: Listar grupos docentes.
F79: Asignar profesores a grupos docentes.	F80: Listar evaluaciones de los estudiantes.
F81: Modificar evaluaciones de los estudiantes.	F82: Enviar mensajes privados a otros usuarios.
F83: Listar mensajes privados.	F84: Eliminar mensajes privados.
F85: Marcar mensajes privados como no leídos.	F86: Marcar mensajes privados como leídos.

2.2.5 Requisitos no funcionales.

Los RNF como su nombre lo sugiere son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de los datos, así como la protección, disponibilidad, entre otras propiedades emergentes (Olivera & Novelo, 2010).

- **Requerimiento de estándares:**

RNF1: Todas las clases y funciones deben quedar bien documentadas.

RNF2: Implementar códigos robustos donde se comprueben todos los datos, evitando así colisiones inesperadas.

- **Requerimientos de rendimiento:**

RNF3: La velocidad de procesamiento de la información y los tiempos de respuestas deben ser menores a los 3000 milisegundos.

- **Requerimientos de fiabilidad:**

RNF4: El sistema debe estar disponible un 98% del tiempo, dejando el otro 2% para tareas de mantenimiento, actualizaciones y corrección de errores.

RNF5: Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad.

- **Requerimientos de eficiencia:**

RNF6: El tiempo de demora de una petición al servidor debe estar comprendido entre los dos (2) y los cinco (5) segundos aproximadamente.

RNF7: El sistema debe permitir la navegación multiusuarios.

2.2.6 Historias de usuario.

La metodología de desarrollo ágil de software *XP* genera como uno de sus artefactos a las Historias de Usuario (HU) que consiste en una técnica para representar y especificar los requisitos del software, a través de un conjunto de tablas en las cuales el cliente describe brevemente las características que debe poseer el sistema. Estas HU son descompuestas en tareas de programación que serán implementadas durante una iteración (Canós, José H, Leterier, Patricio, & M^a Carmen, 2003).

Para la presente investigación se generaron un total de ochenta y seis (86) HU, a continuación, solo se muestran dos (2) de ellas, tres (3) de las restantes se exponen en el [Anexo #5](#) y se podrán observar todas en el documento “*Historias de Usuario (HU).pdf*” adjunto a este informe como artefacto generado.

Tabla 4: HU_1 Experimentar con la Teoría de conjuntos (Elaboración propia).

Historia de Usuario	
Número: HU_1	Nombre: Experimentar con la Teoría de conjuntos.
Programador responsable: José A. Alvarez Abaira	Iteración asignada: 1
Descripción: Permite experimentar con la teoría de conjuntos, para ello debe permitirse que se gestionen los conjuntos con los que se desea trabajar y posteriormente se le dará al usuario las siguientes opciones sobre los conjuntos especificados: resolver ecuaciones, intersección, resolver las siguiente operaciones:	

unión, diferencia, diferencia simétrica, conjunto potencia, producto cartesiano, cardinalidad, complemento, partición, determinar si un conjunto es subconjunto, subconjunto propio, disjuntos, equivalentes, iguales o comparables con otro y si un elemento pertenece al conjunto.

Observaciones: Tendrán acceso los estudiantes y profesores. Para la gestión de los conjuntos debe tenerse en cuenta que se debe definir un universo y de ahí en adelante todos los conjuntos que se definan tienen que poseer solamente elementos de los definidos en el universo. Los conjuntos no deben poseer elementos repetidos, el vacío se representará con un guión (-) y no pueden existir conjuntos con el mismo nombre.

Al hacer clic en cualquiera de los botones “*Comprobar*” se mostrará un mensaje diciendo si el resultado es correcto o no y en cada operación se indicará también la validez de la respuesta, teniendo en cuenta que solo se califican las operaciones respondidas por el usuario. En caso que la respuesta sea incorrecta se dará la opción de visualizar cuál es la respuesta correcta.

Tabla 5: HU_2 Exponer teoría de la Teoría de conjuntos (Elaboración propia).

Historia de Usuario	
Número: HU_2	Nombre: Exponer teoría de la Teoría de conjuntos.
Programador responsable: José A. Alvarez Abaira	Iteración asignada: 1
Descripción: Permite mostrar en el sistema el contenido de la teoría de conjuntos a través de un archivo PDF.	
Observaciones: Tendrán acceso los estudiantes y profesores. El archivo debe mostrarse embebido en la página y además contener un enlace para descargarlo.	

2.2.7 Plan de entregas.

Tabla 6: Plan de entregas (Elaboración propia).

Entregable	Iteración	Fin de la iteración
Contenido, Experimentación y Evaluación del tema “Teoría de Conjuntos”.	1	Enero 2016
Contenido, Experimentación y Evaluación del tema “Relaciones binarias”.	2	Enero 2016
Contenido, Experimentación y Evaluación del tema “Teoría combinatoria”.	3	Enero 2016
Gestión de los ejercicios competitivos.	4	Febrero 2016

Gestión de usuarios.	5	Febrero 2016
Contenido, Experimentación y Evaluación del tema “Teoría de Grafos”.	6	Marzo 2016
Contenido, Experimentación y Evaluación del tema “Máquinas de Turing”.	7	Abril 2016
Contenido, Experimentación y Evaluación del tema “Relaciones de Recurrencia”.	8	Mayo 2016

2.2.8 Plan de iteraciones.

Un plan de iteraciones consiste en generar una planificación donde los desarrolladores en conjunto con el cliente definen los tiempos ideales para la implementación de las HU, la prioridad de estas y cuáles serán implementadas en cada versión del programa (Carballo, 2014).

Para la implementación del OA-IE se definieron 8 iteraciones distribuidas como puede observarse en la siguiente tabla:

Tabla 7: Plan de iteraciones (Elaboración propia).

Iteración	Duración (semanas)	Historias de usuario
1	6	De la 1 a la 9
2	2	De la 10 a la 18
3	2	De la 19 a la 27
4	2	De la 58 a la 65
5	4	De la 66 a la 86
6	3	De la 28 a la 37
7	3	De la 38 a la 47
8	2	De la 48 a la 57

2.2.9 Tarjetas CRC.

La metodología XP en lugar de utilizar diagramas de clases utilizando la notación UML se pueden utilizar otras técnicas, una de ellas son las tarjetas *CRC* (*Class-Responsability-Collaboration*) (Beck, 2000).

La utilización de estas tarjetas permite al desarrollador centrarse en el desarrollo orientado a objetos. La estructura de una tarjeta *CRC* está dividida en tres (3) partes: nombre de la clase, sus responsabilidades y colaboradores. Una clase puede describirse como cualquier objeto o evento mediante sus atributos y métodos, las responsabilidades consisten en las tareas que realiza dicho objeto o evento y los colaboradores son otras clases que trabajan en conjunto con ella para cumplir determinado objetivo. A continuación, se muestran solo dos (2) tarjetas *CRC*, en el [Anexo #6](#) podrán observarse tres (3) más, y las restantes se encuentran en el documento "*Tarjetas CRC.pdf*" adjunto a este informe como artefacto generado.

Tabla 8: Tarjeta CRC "Contenido" (Elaboración propia).

Contenido	
Responsabilidades	Colaboraciones
Se encarga de mostrar el contenido de todos los temas de la MD incluidos en el sistema.	----

Tabla 9: Tarjeta CRC "Experimentacion.Conjuntos" (Elaboración propia).

Experimentacion.Conjuntos	
Responsabilidades	Colaboraciones
Se encarga de realizar la experimentación de la Teoría de Conjuntos, para ello comprueba que las respuestas del usuario sean correctas.	Experimentación.Validaciones

2.2.10 Arquitectura del software.

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución (ASPP, 2015).

PATRÓN MODELO – VISTA – PLANTILLA.

Según (Infante, 2012) Django es un *framework* MTV del inglés *Model-Template-View* que es una modificación de MVC (Modelo-Vista-Controlador) debido a que los desarrolladores del *framework* no tuvieron la intención de seguir algún patrón de desarrollo sino hacerlo lo más funcional posible. Para comenzar a entender Django se debe tener en cuenta su analogía con MVC de la siguiente forma:

- El modelo en Django sigue siendo Modelo (M).
- La vista en Django pasa a llamarse *Template* (T).
- El controlador en Django pasa a llamarse Vista (V).

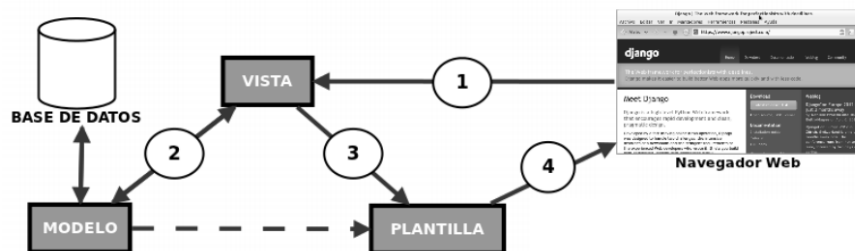


Ilustración 3: Funcionamiento del MTV de Django (Infante, 2012).

A continuación, se explica detalladamente el funcionamiento del MTV de Django representado en la figura anterior:

1. El navegador envía una solicitud.
2. La vista interactúa con el modelo para obtener datos.
3. La vista llama a la plantilla.
4. La plantilla renderiza la respuesta a la solicitud del navegador.

De lo visto anteriormente acerca de cómo trabaja el MTV de Django se derivan los siguientes elementos:

- **El modelo:** define los datos almacenados, es representado en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.
- **La vista:** su propósito es determinar qué datos serán visualizados, es representado en forma de funciones. El ORM (*Object Relational Mapping*) de Django permite escribir código Python en lugar de SQL (*Structured Query Language*) para hacer las consultas. Además, se encarga de tareas

como el envío de correo electrónico, autenticación con servicios externos y la validación de datos a través de formularios.

- **La plantilla:** recibe los datos de la vista y luego los organiza para la presentación al navegador web. Básicamente es una página HTML (*HyperText Markup Language*) con algunas etiquetas extras que son propias del Django, dichas etiquetas permiten flexibilidad para los desarrolladores del *frontend*¹².

De manera general cada una de las aplicaciones del sistema cuenta con un fichero `models.py` donde se definen todos los modelos, `views.py` que es donde se definen los controladores y una serie de archivos HTML que representan las vistas. A continuación, se muestra un ejemplo del patrón en la implementación del sistema:

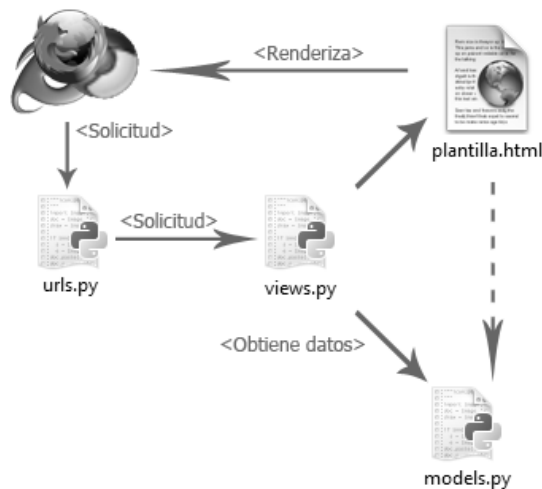


Ilustración 4: Ejemplo de utilización del patrón MVT en el sistema (Elaboración propia).

PATRONES DE DISEÑO.

¹² Es la parte del desarrollo web que se dedica de la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos.

- **Patrones GRASP (Responsability Assignment Software Patterns):** Según (Grosso, 2011) son patrones basados en la asignación de responsabilidades a objetos. Es una buena práctica para el desarrollo eficaz de la Programación Orientada a Objetos (POO).
 - **Experto en información:** consiste en asignar una responsabilidad al experto en información, es decir a la clase que contiene toda la información necesaria para desempeñar una responsabilidad (Informáticos, 2005).

```
def get_ranking_5(self):
    estudiantes = Estudiante.objects.filter(competitivo=True).order_by('-ranking')
    if len(estudiantes) >= 5:
        cantidad = 5
    else:
        cantidad = len(estudiantes)
    lista = []
    for x in range(0, cantidad, 1):
        lista.append(estudiantes[x])
    return lista
```

Ilustración 5: Código para obtener los cinco (5) primeros estudiantes del ranking de las competencias en línea (Elaboración propia).

- **Creador:** consiste en asignar a una determinada clase B la responsabilidad de crear una instancia de la clase A al ocurrir alguna de las siguientes circunstancias: B agrega a A, B tiene los datos de inicialización de A, B registra a A o B utiliza estrechamente a A (Informáticos, 2005).

```
@login_required(login_url='/')
@permission_required(perm='change_grupodocente', login_url='/denegado')
def grupos_docentes eliminados(request):
    grupos = GrupoDocente.objects.filter(eliminado=True).order_by('nombre')
    datos = {'grupos': grupos}
    return render_to_response('seguridad/lista_grupos_docentes eliminados.html', {'datos': datos},
                              RequestContext(request))
```

Ilustración 6: Código para mostrar el listado de grupos docentes eliminados (Elaboración propia).

- **Bajo acoplamiento:** plantea que debe existir una alta reutilización entre las funcionalidades de las clases con una mínima dependencia, contribuyendo así al mantenimiento de las mismas (Peralta & Durán, 2014). Este patrón ya viene incluido con Django que permite un bajo acoplamiento entre las piezas, lo que evita las dependencias,

por lo que por ejemplo a la hora de realizar cambios en las configuraciones de las *URL*¹³, en la BD, plantillas HTML, etc., basta solo con realizarlo una sola vez.

- **Alta cohesión:** consiste en asignar las responsabilidades teniendo en cuenta que permanezcan altamente cohesionados, es decir, que su utilización facilite la comprensión del diseño y el incremento de las capacidades de reutilización. Una alta cohesión permite que las clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme.

```
def validar_cardinalidad(conjunto, respuesta):
    cardinalidad = calcular_cardinalidad(conjunto)
    if unicode(respuesta) == unicode(cardinalidad):
        return True
    return False

def calcular_cardinalidad(conjunto):
    if conjunto == u'':
        return 0
    else:
        return len(conjunto)
```

Ilustración 7: Código para validar la cardinalidad de un conjunto (Elaboración propia).

- **Controlador:** Asigna la responsabilidad de gestionar un mensaje de un evento del sistema a una clase controladora. Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

```
@login_required(login_url='/')
@permission_required('perm=change_user', login_url='/denegado')
def incluir_estudiante(request, id):
    estudiante = get_object_or_404(Estududiante, pk=id, is_staff=False)
    estudiante.is_staff = True
    estudiante.save()
    messages.success(request, 'Estudiante incluido satisfactoriamente.')
    return HttpResponseRedirect('/usuarios/estudiantes/eliminados')
```

Ilustración 8: Código para incluir a un estudiante una vez que haya sido eliminado (Elaboración propia).

- **Patrones GoF (Gang of Four):** Representan patrones que representan soluciones técnicas basadas en POO que favorecen la reutilización del código.

¹³ Del inglés *Uniform Resource Locator*.

- **Decorador:** Es un patrón estructural que extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase (Peralta & Durán, 2014). Como ejemplo de este patrón se evidencia la utilización de los decoradores de Django (`@login_required`, `@permission_required`, etc).

```
@login_required(login_url='/')
@permission_required(perm='change_user', login_url='/denegado')
def deshabilitar_administrador(request, id):
    deshabilitar_usuario(request, id)
    return HttpResponseRedirect('/usuarios/administradores')
```

Ilustración 9: Código para deshabilitar a un administrador del sistema (Elaboración propia).

2.2.11 Modelo de datos.

Un modelo de datos consiste en un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones (Zorrilla, 2011).

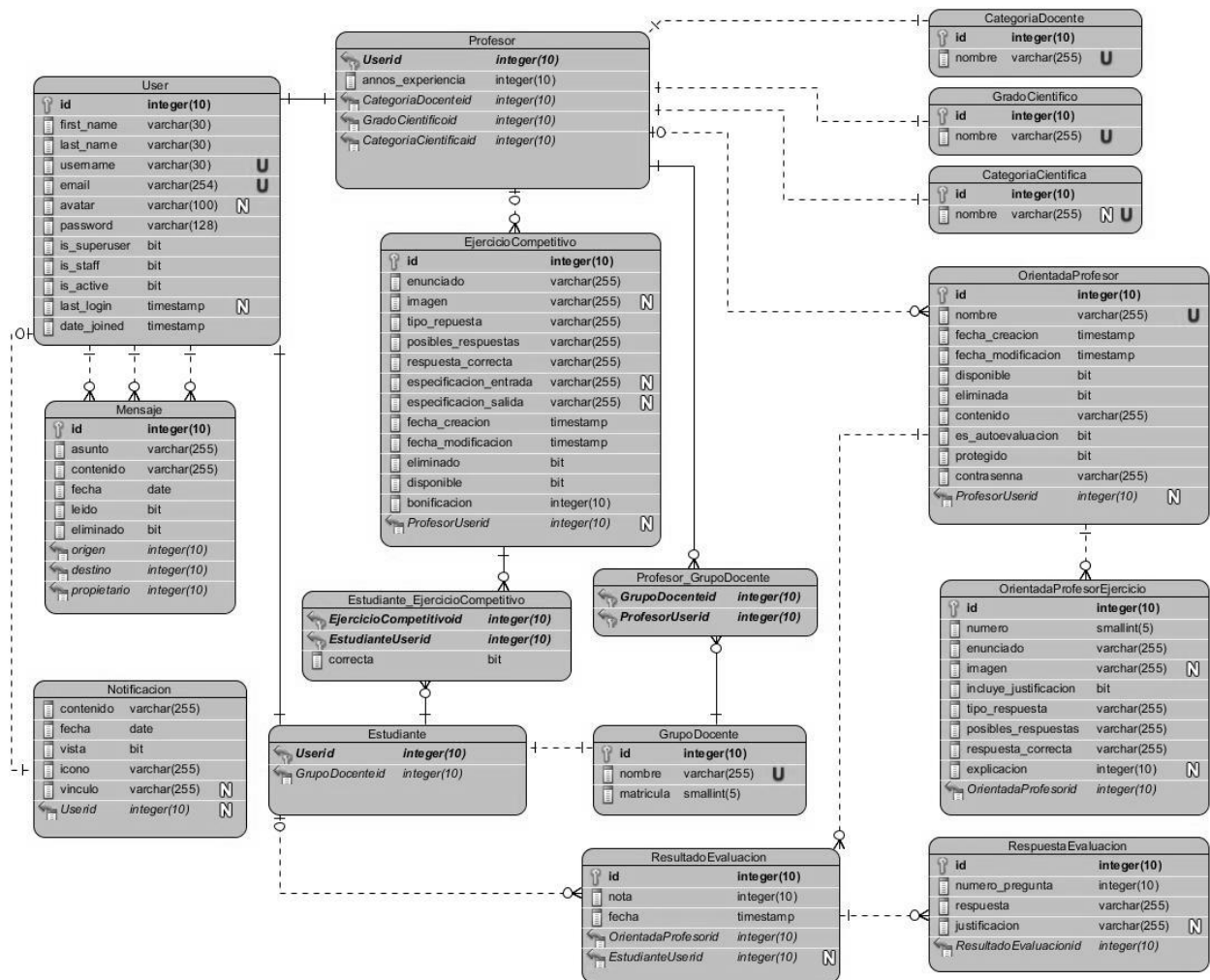


Ilustración 10: Modelo de datos (Elaboración propia).

2.2.12 Diagrama de despliegue.

El autor de la presente investigación coincide con (Sarmiento, 2016) cuando plantea que un diagrama de despliegue consiste en un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue o distribución de los artefactos del *software* en los destinos de despliegue; definiendo a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son archivos ejecutables, bibliotecas, archivos, esquemas de BD, archivos de configuración, etc. Cuando se refiere a destino de despliegue se hace referencia a un nodo que es o bien un dispositivo de hardware o bien un entorno de ejecución de *software*.

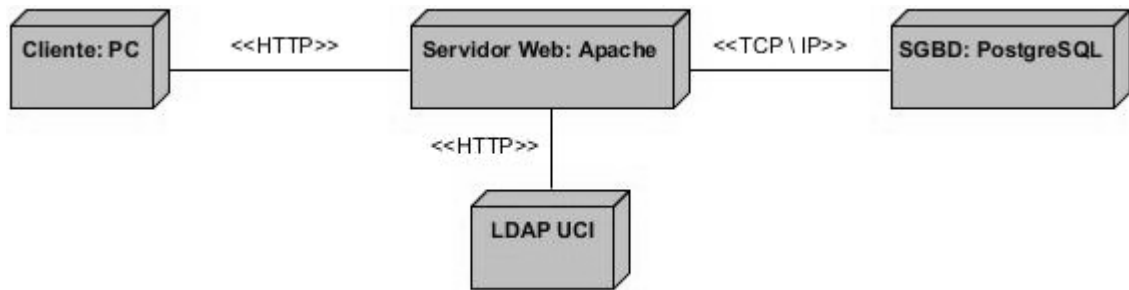


Ilustración 11: Diagrama de despliegue (Elaboración propia).

Conclusiones parciales del capítulo.

Luego de realizar la encuesta a profesores que imparten la asignatura y estudiantes que ya la cursaron se pudo analizar el estado actual de los temas propuestos para el OA-IE donde se evidenciaron los problemas que tienen los estudiantes del primer año de la UCI en la comprensión de la MD, lo que permitió dar una propuesta de solución donde se definieron las unidades: teórica, de experimentación y de evaluación y retroalimentación.

El levantamiento de las funcionalidades y sus respectivas historias de usuario, en conjunto con los requisitos no funcionales y demás artefactos permitió diseñar una propuesta de solución que incluye una serie de patrones GRASP y GoF a utilizar en la implementación para lograr una adecuada reutilización del código para futuras versiones.

Capítulo 3: Implementación y pruebas de la propuesta de solución.

3.1 Estándar de codificación.

Cada programador tiene su propia forma de escribir los códigos y puede ser completamente diferente a la de otros programadores, pero de la forma que se use depende la facilidad de que otros programadores entiendan el código y se les facilite su reutilización, de ahí se desprende la importancia de los estilos de programación, también conocidos como estándares o convenciones de código los cuales definen un grupo de convenciones para escribir código fuente en ciertos lenguajes de programación. A continuación, se define la relación de estándares de codificación a utilizar en la implementación del sistema.

Tabla 10: Estándares de codificación a utilizar en la implementación del sistema (Guía de estilo para el código Python - PEP 8 en Español, 2013).

Tipo de estándar	Descripción del estándar
Indentación	<ul style="list-style-type: none"> Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes). Utilizar una indentación de una tabulación para cada línea con excepción de la primera. La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios.
Máxima longitud de las líneas	<ul style="list-style-type: none"> Todas las líneas deben estar limitadas a un máximo de setenta y nueve (79) caracteres. Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas. En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas.
Líneas en blanco	<ul style="list-style-type: none"> Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco. Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco. Se puede utilizar líneas en blanco escasamente para separar secciones lógicas.
Codificaciones	<ul style="list-style-type: none"> Utilizar la codificación UTF-8. Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”.
Importaciones	<ul style="list-style-type: none"> Las importaciones deben estar en líneas separadas. Siempre deben colocarse al comienzo del archivo. Deben quedar agrupadas de la siguiente forma:

1. Importaciones de la librería estándar.
 2. Importaciones terceras relacionadas.
 3. Importaciones locales de la aplicación / librerías.
- Cada grupo de importaciones debe estar separado por una línea en blanco.
 - Evitar utilizar espacios en blanco en las siguientes situaciones:
 - Inmediatamente dentro de paréntesis, corchetes y llaves.
 - Inmediatamente antes de una coma, un punto y coma o dos puntos.
 - Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función.
 - Inmediatamente antes de un corchete que empieza una indexación.
 - Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro.
 - Deben rodearse con exactamente un espacio los siguientes operadores binarios:
 - Asignación (=).
 - Asignación de aumentación (+=, -=, etc.).
 - Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not).
 - Expresiones lógicas (and, or, not).
 - Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad.
 - No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.
 - Los comentarios deben ser oraciones completas.
 - Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
 - Nunca cambiar las minúsculas y mayúsculas en los identificadores de clases, objetos, funciones, etc.
 - Si un comentario es corto el punto final puede omitirse.
 - Deben estar indentados al mismo nivel que el código a comentar.
 - Cada línea de un comentario en bloque comienza con un numeral (#) y un espacio en blanco.
 - Se recomienda utilizarlos escasamente.
 - Se debe definir comenzando por un numeral (#) seguido de un espacio en blanco.
 - Deben ubicarse en la misma línea que se desea comentar.

Espacios en blanco en expresiones y sentencias

Comentarios

Comentarios en bloque

Comentarios en la misma línea

Cadenas de documentación

- Deben quedar documentados todos los módulos, funciones, clases y métodos públicos.
- Para definir una cadena de documentación debe quedar encerrada dentro de (“”).
- Los (“”) que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea.

Convenciones de nombramiento

- Nunca se deben utilizar como simple caracteres para nombres de variables los caracteres ele minúscula “l”, o mayúscula “O”, ele mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0).
- Los módulos deben tener un nombre corto y en minúscula.
- Los nombres de clases deben utilizar la convención “*CapWords*” (palabras que comienzan con mayúsculas).
- Los nombres de las excepciones deben estar escrito también en la convención “*CapWords*” utilizando el sufijo “Error”.
- Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guión bajo “_”.
- Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guión bajo (_).

3.2 Pruebas funcionales al sistema.

Las pruebas funcionales son realizadas al sistema con el objetivo de validar cuándo el comportamiento observado del software probado cumple o no con sus especificaciones; este tipo de prueba se toma desde el punto de vista del usuario. Las funciones son probadas ingresando las entradas y examinando las salidas (Pérez, 2007).

3.2.1 Pruebas de aceptación.

Este tipo de prueba tienen como propósito demostrar al cliente el cumplimiento de un requisito de software describiendo un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del cliente (Letelier, 2007).

A continuación, se muestran los primeros dos (2) casos de pruebas de aceptación, tres (3) de los restantes se encuentran en el [Anexo #7](#) y se podrán visualizar todas en el documento “*Pruebas de aceptación.pdf*” generado como artefacto.

Tabla 11: Caso de prueba de aceptación “Exponer contenido de la Teoría de conjuntos”.

Caso de prueba de aceptación

Código del caso de prueba: MD_UCI_CPA_2	Nombre de Historia de Usuario: Exponer contenido de la Teoría de conjuntos.	
Persona que realiza la prueba: José Angel Alvarez Abraira.		
Descripción de la prueba: Prueba funcional a la unidad teórica de la Teoría de conjuntos.		
Condiciones de ejecución: Estar autenticado en el sistema.		
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Autenticarse en el sistema. 2. Acceder al menú CONTENIDOS / Teoría de conjuntos / Teoría. 3. Se muestra la teoría del tema Teoría de conjuntos brindando la opción de descargarlo. 		
Escenarios	Resultados esperados	Evaluación de la prueba
SC1: Presentar documento “.pdf”.	Se muestra un archivo “.pdf” embebido a partir de la página cinco (5).	Satisfactoria
SC2: Descargar archivo	Se brinda la opción de descargar el archivo “.pdf”.	Satisfactoria

Tabla 12: Caso de prueba de aceptación "Experimentar con las Máquinas de Turing".

Caso de prueba de aceptación		
Código del caso de prueba: MD_UCI_CPA_3	Nombre de Historia de Usuario: Experimentar con las Máquinas de Turing.	
Persona que realiza la prueba: José Angel Alvarez Abraira.		
Descripción de la prueba: Prueba funcional a la experimentación con las Máquinas de Turing.		
Condiciones de ejecución: Estar autenticado en el sistema.		
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Autenticarse en el sistema. 2. Acceder al menú CONTENIDOS / Máquinas de Turing / Experimentación. 3. Se muestra un área para definir el alfabeto, otra para los estados y una para definir las aristas. 		
Escenarios	Resultados esperados	Evaluación de la prueba

<p>SC1: Definir la Máquina de Turing.</p>	<p>Primeramente, debe definirse el alfabeto atendiendo a las siguientes restricciones:</p> <ul style="list-style-type: none"> • Deben introducirse los elementos separados por coma. • No deben existir elementos repetidos. <p>Luego deben definirse los estados de la máquina para ello, se cuenta con un campo de texto para introducir el identificador del estado y dos (2) <i>checkbox</i> para denotar al estado como inicial y/o final. Para definir un estado se debe tener en cuenta que:</p> <ul style="list-style-type: none"> • No deben existir estados con el mismo identificador. • Debe existir uno y solo un estado inicial. <p>Una vez definidos los estados se deben introducir las aristas, para ello se cuenta con dos listas desplegables donde se deben escoger el origen y destino, además, en se muestran dos campos de texto donde debe introducirse el carácter que reconocerá y escribirá dicha arista y de escoger de una lista desplegable el tipo de movimiento (Derecha, Izquierda, Ninguno). Para definir una arista debe tenerse en cuenta las siguientes restricciones:</p> <ul style="list-style-type: none"> • El carácter de lectura y escritura debe pertenecer al alfabeto. • No debe existir más de una arista con el origen y el carácter de lectura iguales. • Para especificar el carácter vacío se hace con el numeral (#). 	<p>Satisfactoria</p>
<p>SC2: Representar gráficamente.</p>	<p>Al dar clic en el botón ACEPTAR del escenario 1 aparece un área donde se visualiza una representación gráfica de la Máquina de Turing introducida. Además, deben brindarse las siguientes opciones:</p>	<p>Satisfactoria</p>

	<ul style="list-style-type: none"> • EDITAR DATOS: Permite cambiar los datos de la máquina. • VER IMAGEN: Permite visualizar la máquina como una imagen en una pestaña nueva del navegador. • EXPORTAR COMO PNG: Permite descargar la máquina como una imagen en formato “.png”. • RE-DIBUJAR: En caso que la vista actual no sea conveniente puede volver a dibujar la máquina con los mismos datos introducidos. <p>La representación gráfica debe desplazarse por toda el área y permitir organizar los estados convenientemente.</p>	
SC3: Comprobar cadena.	Al dar clic en el botón ACEPTAR del escenario 1 se muestra un área con un campo de texto donde el usuario puede introducir una cadena para comprobar si la cadena es reconocida por la máquina o no y verificar que se cumpla la respuesta del usuario escogida de la lista desplegable.	Satisfactoria
SC4: Comprobar cadenas aleatorias.	Al dar clic en el botón ACEPTAR del escenario 1 se muestra un área donde el usuario pueda elegir entre las siguientes condiciones: <ul style="list-style-type: none"> • Comienza con... • No comienza con... • Contiene la subcadena... • No contiene la subcadena... • Termina en... • No termina en... <p>El sistema debe generar una serie de cadenas aleatorias que cumplan con las condiciones especificadas para comprobar que son reconocidas por la máquina y una serie de cadenas aleatorias que no cumplan con las condiciones para comprobar que no son reconocidas por la máquina. Luego de</p>	Satisfactoria

	comprobarla deben mostrarle al usuario las cadenas que fueron probadas.	
--	---	--

3.2.2 Evaluación de la calidad desde el punto de vista pedagógico.

Debido a las características de los OA-IE se pueden considerar desde el punto de vista metodológico como un producto de software que a su vez constituye un producto pedagógico, por lo que además de probar cada una de las funcionalidades del sistema debe ser evaluada también la calidad desde el punto de vista pedagógico. Para ello se decide utilizar como estrategia de evaluación de la calidad la Guía propuesta por Toll Palma en el año 2011 (Ver [ANEXO #8](#)) para evaluar la calidad de los OA-IE.

El primer apartado está conformado por los indicadores de evaluación agrupados en tres (3) aspectos: formativo, de diseño y presentación y tecnológico; definiendo un conjunto de treinta y cuatro (34) indicadores de evaluación de cumplimiento básico para cualquier OA. Cada uno de los indicadores estará evaluado por la escala de Excelente (3 puntos), Bien (2 puntos), Regular (1 punto) y Mal (0 puntos).

El objetivo principal de esta puntuación es emitir una evaluación final que permita evaluar el OA-IE según los rangos de la escala definida en Muy Adecuado, Adecuado, Poco Adecuado y No Adecuado, de acuerdo al nivel de adecuación del OA-IE será el nivel de calidad alcanzado por el sistema como recurso pedagógico.

Para la evaluación del OA-IE se entregó los aspectos de la guía a seis (6) profesores de MD de la actual FICI para que emitieran su criterio de forma anónima acerca de cada uno de los seis (6) temas incluidos y las competencias en línea del OA-IE para luego con la suma de ellos otorgar una evaluación final de acuerdo a la siguiente escala definida por (García A. , 2014):

Tabla 13: Escala para medir la calidad del OA-IE (García A. , 2014).

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

A partir de los resultados obtenidos en la evaluación (Ver ANEXO #9) se puede constatar que los seis (6) profesores coinciden en una calificación de Muy Adecuado para el OA-IE, lo que demuestra el cumplimiento de los objetivos planteados para la investigación validando así la importancia y valor del OA-IE implementado.

Conclusiones parciales del capítulo.

Los estándares de codificación definidos permitieron implementar códigos completamente legibles para el buen entendimiento de otros programadores facilitando así su recodificación.

La validación del sistema como producto de software demostró el cumplimiento de las ochenta y seis (86) HU definidas por el cliente sin quedar no conformidades sin resolver.

La validación del sistema como recurso pedagógico mediante la Guía de Toll Palma arrojó una evaluación de Muy Adecuado lo cual demostró la validez del sistema para ser utilizado en el PEA de la MD.

CONCLUSIONES

La presente investigación tuvo como base el desarrollo de un Sistema interactivo - experimental para el apoyo al proceso de enseñanza – aprendizaje de la Matemática Discreta en la carrera Ingeniería en Ciencias Informáticas, para ello se dio cumplimiento a una serie de tareas de la investigación, las cuales fueron cumplidas satisfactoriamente, por lo que se puede arribar a las siguientes conclusiones:

- La profundización en los principales referentes teóricos que sustentan el desarrollo y utilización de un sistema web interactivo - experimental en el proceso de enseñanza – aprendizaje de la MD en la Ingeniería en Ciencias Informáticas permitió identificar a la MD como una asignatura esencial en la formación de profesionales de la informática y las deficiencias que presenta actualmente en la UCI.
- El análisis y diseño del OA-IE permitió identificar los contenidos a incluir en el sistema a través de una encuesta realizada a una serie de estudiantes y profesores teniendo en cuenta el grado de complejidad de impartición y asimilación, la posibilidad de representación gráfica y la posibilidad de experimentar. Lo anterior se materializó en una propuesta de solución donde se define la unidad de información, didáctica y de evaluación y retroalimentación para lo cual se generaron una serie de artefactos ingenieriles.
- Durante la implementación del sistema se utilizaron una serie de estándares de codificación que permiten una futura reutilización del código, además, fueron utilizados un conjunto de algoritmos matemáticos complejos que llevaron a la implementación de los requisitos exigidos por el cliente.
- Al validar el OA-IE se comprobó el correcto cumplimiento de cada una de las historias de usuario definidas, así como la correctitud del sistema como recurso pedagógico obteniendo una calificación de Muy Adecuado, lo cual permitió comprobar que el objetivo propuesto para la presente investigación finalmente fue cumplido.

RECOMENDACIONES

El autor de la presente investigación propone las siguientes recomendaciones:

- Realizar versiones posteriores donde se incluya el tema de Lógica y se enriquezcan los temas implementados con otras operaciones que puedan ser añadidas en aras de fortalecer aún más el PEA de la MD.
- Implementar una versión del sistema adaptable a dispositivos móviles.

REFERENCIAS BIBLIOGRÁFICAS

- Algieri, R. D., Mazzoglio, M., & Castro, F. (2012). Espacios Virtuales de Enseñanza y Aprendizaje (EVEA) en Esplacnología: Especificidades Pedagógicas en su Enseñanza Topográfica. *International Journal of Morphology*.
- Álvarez, M. A. (2009). *Matemática Discreta (LADE+) ITIG*. Obtenido de <http://www.escet.urjc.es/~rmunoz/discreta.html>
- Area, M. (2009). Introducción a la Tecnología Educativa. *Manual electrónico*. España.
- ASPP, I. (2015). Multimedia Takes on Societal Challenges. *EIC's Message*.
- Báez, S. (2012). *Sistemas Web*. Obtenido de KnowDo: <http://www.knowdo.org/knowledge/39-sistemas-web>
- Beck, K. (2000). *Extreme Programming Explained. Embrace Change*. Pearson Education.
- Benvenuto, A. (2003). *LAS TECNOLOGIAS DE INFORMACION Y COMUNICACIONES (TIC) EN LA DOCENCIA UNIVERSITARIA*. Obtenido de <http://www.redalyc.org/resumen.oa?id=29901210>
- Bolivariana, U. U. (2009). Programación Extrema (XP).
- Booch, G., Rumbaugh, J., & Jacobson, I. (2004). *UML. El Lenguaje Unificado de Modelado*.
- Cabero, J. (2006). *Tecnología educativa: su evolución histórica y su conceptualización*. España.
- Canós, José H, Leterier, Patricio, & M^a Carmen. (2003). *Método logías Ágiles en el Desarrollo de Software*. Obtenido de No Quality Inside - Cursos: http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf
- Carballo, Y. A. (2014). Sistemas de Objetos de Aprendizaje Interactivos y Experimentales para la Matemática Discreta. *Trabajo de Diploma en opción al título de Ingeniero en Ciencias Informáticas*. La Habana, Cuba.
- Carnoy, M. (2004). Las TIC en la enseñanza: posibilidades y retos. *Lección inaugural del curso académico 2004-2005*.
- Chan, M. E. (2001). *Objetos de Aprendizaje: una herramienta para la innovación educativa*.
- Chiappe, A. (2009). Evolución conceptual de los objetos de aprendizaje.
- Corrales, A. R. (Diciembre de 2009). La integración de las Tecnologías de la Información y Comunicación (TIC) en el área de la Educación Física. *Hekademos*.
- Dayley, B. (2007). *Python Phrasebook*.

- de Kereki, I. F. (Diciembre de 2003). Modelo para la Creación de Entornos de Aprendizaje basados en técnicas de Gestión del Conocimiento. *Tesis Doctoral*. España.
- De la Rosa, R. (2006). *La interactividad del software educativo en el aprendizaje: Multimedia interactiva*. Obtenido de Ilustrados: <http://www.ilustrados.com/tema/9870/interactividad-software-educativo-aprendizaje-Multimedia-interactiva.html>
- Figuroa, R., Solís, C., & Cabrera, A. (2007). Metodologías tradicionales vs. metodologías ágiles.
- Galindos, R. (2013). Guión Visual Paradigm for UML.
- García, A. (2014). Estrategia metodológica para la elaboración y utilización de objetos de aprendizaje interactivos y experimentales en el proceso de enseñanza - aprendizaje de la Matemática Discreta en la UCI. *Tesis presentada en opción al título académico de Máster en Ciencias Matemáticas. Mención: Enseñanza de la Matemática*. La Habana, Cuba.
- García, W. Y. (2009). Propuesta de una herramienta gráfica para la reutilización de componentes en el análisis y diseño de software educativo.
- Grosso, A. (2011). *Prácticas de software, Experiencias sobre la Ingeniería y Management del Software*. Obtenido de <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>
- Guía de estilo para el código Python - PEP 8 en Español*. (2013). Obtenido de www.recursopython.com
- Gutiérrez, J. (2006). Framework.
- Herrera, I. (2012). Conjunto de Objetos de Aprendizajes para la Matemática Discreta. *Trabajo de Diploma en opción al título de Ingeniero en Ciencias Informáticas*. La Habana, Cuba.
- Holovaty, A., & Kaplan-Moss. (2009). *The Definitive Guide to Django: Web Development Done Right*.
- Huidobro. (2007). *Tecnologías de información y comunicación*. Obtenido de Monografías: <http://www.monografias.com/trabajos37/tecnologias-comunicacion/tecnologias-comunicacion.shtml>
- Infante, S. (2012). *Curso Django para perfeccionistas con deadlines*.
- Informáticos, D. d. (2005). *Patrones de asignación de responsabilidades (GRASP)*. Universidad de Sevilla.
- Letelier, P. (2007). Pruebas de aceptación como conductor del proceso de software. Universidad Politécnica de Valencia, Valencia.

- López, C. (2010). Los repositorios de objetos de aprendizaje como soporte para entornos e-learning.
- López, P., & Ruiz, F. (2011). Lenguaje Unificado de Modelado - UML.
- Mariana, G. P. (2008). Modelo Pedagógico. Colombia.
- Método experimental.* (2012). Obtenido de http://fdhusacursopsicologia.webpin.com/1145949_Metodo-Experimental.html
- Mifsuf, E. (2013). Apache. España.
- MÓDULO IV: TIC'S Y ENTORNOS VIRTUALES.* (2008). Obtenido de DIPLOMADO EN: EDUCACIÓN A DISTANCIA Y FORMACIÓN VIRTUAL: http://www.academia.edu/9901135/Modulo_IV_-_TIC_s_Entornos_Virtuales
- Moreno, A. J. (2012). *La web 2.0. Recurso educativo.* Obtenido de Observatorio Tecnológico: <http://recursostic.educacion.es/observatorio/web/es/internet/web-20/1060-la-web-20-recursos-educativos>
- Moya, A. M. (2009). Las nuevas tecnologías en la educación.
- Olivera, A. G., & Novelo, C. (2010). *Reporte de instalación de apache.* Escárcega.
- O'Reilly, T. (2003). *Web 2.0.* Obtenido de <http://www.peremarques.net/web20.htm>
- Peralta, C., & Durán, D. (2014). Módulos de edición de plantillas y recepción de órdenes de impresión para el Sistema de Personalización de Documentos de Identidad basado en tecnologías libres. *Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.* La Habana, Cuba.
- Pérez, B. (4 de Julio de 2007). Gestión de las pruebas funcionales. Universidad de la República, Montevideo, Uruguay.
- Pressman, R. S. (2010). *Software Engineering a Practitioner's Approach.*
- PyCharm.* (16 de Diciembre de 2015). Obtenido de <http://www.jetbrains.com/pycharm/>
- Salvador, J. (2015). *Metodologías Ágiles AUP.* Obtenido de http://www.academia.edu/7894130/METODOLOGIAS_AGILES_AUP
- Sarmiento, J. (16 de Febrero de 2016). *UML: Diagrama de despliegue.* Obtenido de Visión general de los diagrama de despliegue: <http://umlidiagramadespliegue.blogspot.com/>
- Serrano, M. d. (2010). Objetos de Aprendizaje. *e-Formadores.*

- Silva, A., Ponce, J. C., & Hernández, Y. (2013). Estado del Arte de las Metodologías para el Desarrollo de Objetos de Aprendizaje. México.
- Sommerville, I. (2005). Ingeniería de Software. 2, *Séptima Edición*, 712. Madrid, España: Pearson Education.
- Terry, & Logee. (1990). *Terminology for Software Engineering and Computer-aided Software Engineering*.
- Toll, Y. d. (2011). Guía de evaluación de la calidad de Objetos de Aprendizaje producidos en la Universidad de Ciencias Informáticas.
- UCI. (2010). *Portal UCI*. Obtenido de <http://www.uci.cu/?q=mision>
- UCI. (2011). *Caribbean Online Judge*. Obtenido de <http://coj.uci.cu/general/about.xhtml>
- UCI. (2012). *Caribbean Mind Forge*. Obtenido de <http://cmf.uci.cu/mindPro/about>
- UCI. (2013). Plan de Estudio "D" del Ingeniero en Ciencias Informáticas.
- UCI. (2014). *Entorno Virtual de Enseñanza Aprendizaje*. Obtenido de <http://eva.uci.cu/mod/resource/view.php?id=30904>
- Ureña, C. (2011). Lenguajes de Programación.
- Van, G. (Septiembre de 2009). El tutorial de Python.
- Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.
- Zorrilla, M. E. (2011). *Modelos de datos*. Universidad de Cantabria.

GLOSARIO DE TÉRMINOS

Didáctica: parte de la pedagogía que estudia las técnicas y métodos de enseñanza.

Educación a distancia: forma de enseñanza en la que los estudiantes no necesitan asistir físicamente al lugar de estudio.

E-learning: educación y capacitación a través de Internet.

Metadatos: datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos.

Metodología: se refiere a una forma de trabajo o un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas informáticos.

Multiplataforma: se refiere a los programas informáticos que pueden funcionar en diversas plataformas.

Objeto de aprendizaje: cualquier recurso digital que puede ser reusado como soporte para el aprendizaje.

Pedagogía: ciencia que tiene como objeto de estudio a la educación.

Sistema Web: son aquellos que se alojan en un servidor web. Su aspecto es similar al de las páginas web, pero poseen funcionalidades potentes que brindan respuestas a casos particulares.

Software interactivo: software que permite la interacción usuario – máquina y usuario – usuario.

Software experimental: programa basado en la observación y la manipulación sistemática de la realidad para evolucionar el conocimiento que se tiene sobre un fenómeno concreto.

Tecnología educativa: se refiere a las TIC aplicadas a la educación.

ACRÓNIMOS

Inicial	Acrónimo
Numérico	3D: 3 Dimensiones.
A	ACM-ICPC: <i>Association for Computing Machinery – International Collegiate Programming Contest.</i> ADDIE: Análisis, Diseño, Desarrollo, Implementación y Evaluación. ASF: <i>Apache Software Foundation.</i> AUP: <i>Agil Unified Process.</i>
B	BD: Base de Datos. CASE: <i>Computer Aided Software Engineering.</i>
C	CD-ROM: <i>Compact Disc Read Only Memory.</i> CMF: <i>Caribbean Mind Forge.</i> COJ: <i>Caribbean Online Judge.</i>
D	DVD: <i>Digital Versatile Disk.</i>
E	EVEA: Entorno Virtual de Enseñanza – Aprendizaje. IBM: <i>International Business Machines.</i>
I	ICI: Ingeniería en Ciencias Informáticas. ISD-MeLO: Metodología de Desarrollo de Sistemas Instruccionales.
L	LEMAT: Libro Electrónico de Matemática. LMS: <i>Learning Management System.</i>
M	MD: Matemática Discreta. MSF: <i>Microsoft Solution Framework.</i>
O	OA: Objeto de Aprendizaje. OA-IE: Objeto de Aprendizaje Interactivo – Experimental. PEA: Proceso de Enseñanza – Aprendizaje.
P	PHP: <i>Hypertext Preprocessor.</i> ROA: Repositorio de Objetos de Aprendizaje.
R	RUP: <i>Rational Unified Process.</i> SGBD: Sistema Gestor de Base de Datos.
S	SQL: <i>Structured Query Language.</i> SSL: <i>Secure Sockets Layer.</i>
T	TE: Tecnología Educativa.

U**TIC:** Tecnologías de la Información y las Comunicaciones.**UCI:** Universidad de las Ciencias Informáticas.**UML:** *Unified Modeling Language*.**X****XP:** *Extreme Programming*.