



Universidad de las Ciencias  
Informáticas

Facultad 1

# Módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*

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

**Autor:**

Carlos Yordan González Herrera

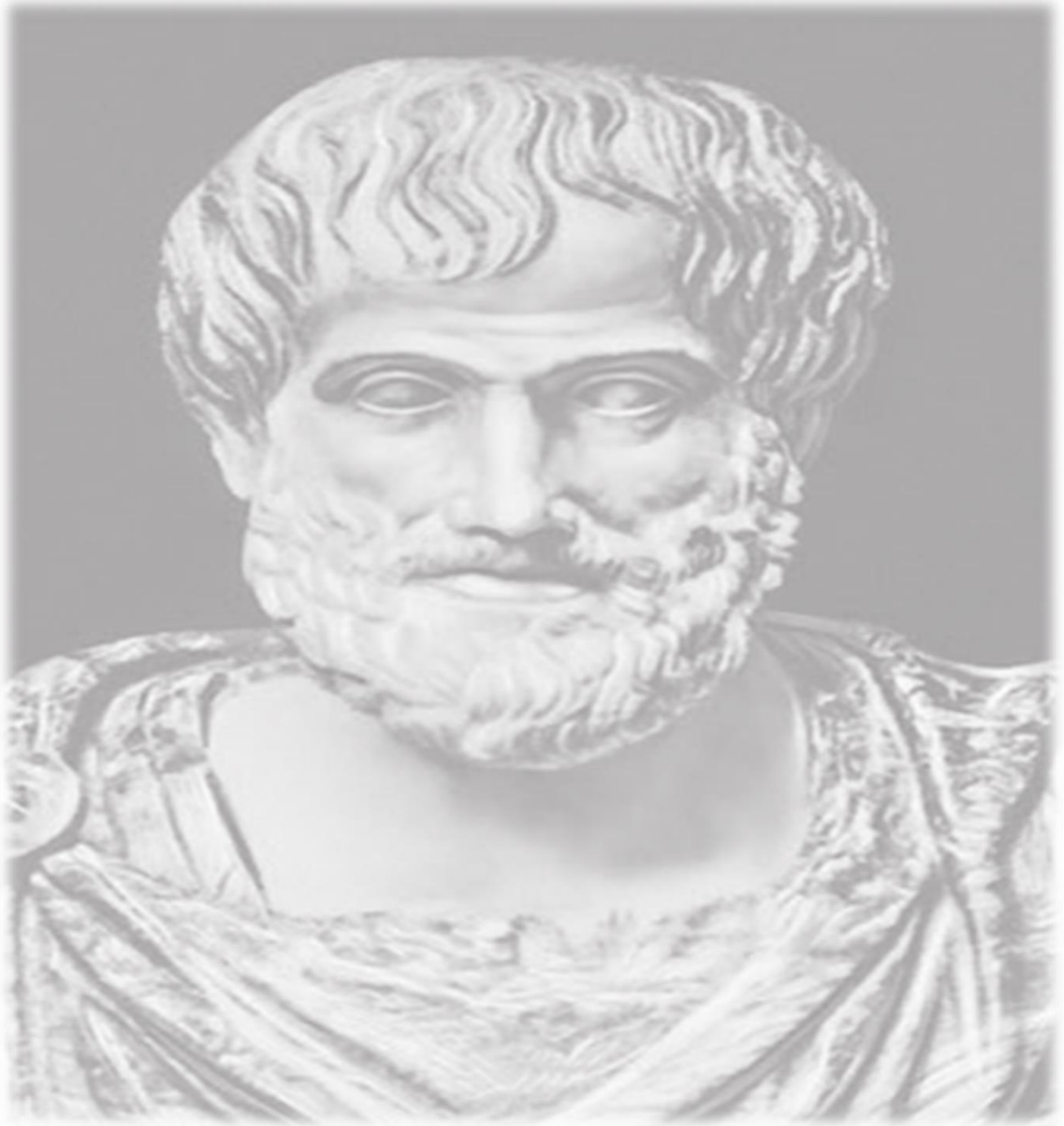
**Tutores:**

Ing. Yaniel Lázaro Aragón Barreda

MSc. Eylín Hernández Luque

Ing. Yarileidis Barcena Calzado

La Habana, junio 2017



*“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica”.*

*Aristóteles*

## Declaración de autoría

Declaro por este medio que yo: Carlos Yordan González Herrera, con carné de identidad 93092906681, soy el autor principal del trabajo final de tesis de pregrado que se titula: “Módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*”. El cual ha sido desarrollado como parte del trabajo en el Departamento de Ingeniería y Gestión de Software, específicamente en la asignatura Sistemas de Bases de Datos de la Facultad 1. Autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2017.

---

**Autor**

Carlos Yordan González Herrera

---

**Tutor**

Ing. Yaniel Lázaro Aragón Barreda

---

**Tutor**

MSc. Eylín Hernández Luque

---

**Tutor**

Ing. Yarileidis Barcena Calzado

## **Dedicatoria**

*A mi familia, y principalmente a los que han estado siempre conmigo y no han dejado, ni un solo día, de ocuparse de mi crecimiento y mi bienestar.*

*Muy especialmente, a mis abuelos y mis padres, que donde quiera que estén, sé que están orgullosos de cómo me he formado y de la persona que soy hoy.*

*A ustedes, con todo mi corazón.*

*Carlos Yordan González Herrera.*

## Agradecimientos

*Quiero agradecer, en un principio, a mi país, a la Revolución cubana y al siempre eterno Comandante en Jefe Fidel Castro Ruz, quién creó esta Universidad que me recibió y me dio la posibilidad de formarme como una persona de bien y como el profesional que debo ser en un futuro inmediato.*

*En ese sentido, también va mi agradecimiento a todas aquellas personas que, de una forma u otra, pusieron su grano de arena en mi formación durante estos cinco años de carrera. A todos mis profesores, alumnos ayudantes y hasta compañeros de año que me ayudaron a obtener buenos resultados docentes y a optar hoy por un título de Oro.*

*A mi decana, quien me brindó su apoyo para con mi investigación, por su ayuda en cuanto problema le planteaba y su cariñosa forma de ser.*

*Mis compañeros de aula, la primera familia que encontré al llegar a la UCI. De todos y cada uno me llevo algo. Especialmente de las niñas que me brindaron su cariño y la dulzura que las caracteriza. A Ortelio, mi cuasi-compañero de tesis, que también estuvo apoyándome con sus conocimientos para lograr buenos resultados.*

*A mi Organización, la FEU, de la cual además fui dirigente, y al mismo tiempo que fomentó en mí, virtudes como el sacrificio y la responsabilidad, me permitió conocer a personas muy especiales que fueron desde mis compañeros de trabajo, hasta muy buenos amigos. A toda mi familia de la FEU, gracias.*

*A otra familia, la del baile, quienes me acompañaron en esa pasión que descubrí al subirme por primera vez a un escenario. A todos los Cumbiamberos, los que seguimos y los que no, gracias por ser parte de algo tan bonito.*

*A todas mis amistades, las que he conocido por una u otra razón, pero que no dudan en detenerse a saludarme y preguntar cómo estoy. A todos y cada uno de ellos, mi más sincero agradecimiento.*

*A mis amigos, mis hermanos de la UCI: Sandra, Bia, Yesenia, Dayana, Arlene, Esteban, Vladis y Abel. Los quiero un montón y siempre los voy a llevar conmigo, aun cuando por las circunstancias de la vida, ya no estemos juntos.*

*Por otra parte, agradezco a todos los tribunales de talleres y eventos, que, con sus criterios, ayudaron a perfeccionar el resultado de mi trabajo de diploma.*

*A mis tutores, por todo su apoyo: Yari, Eylín y especialmente a Yaniel, por confiar en mí y proponerme trabajar con él en este proyecto, por ser mi guía en esta ardua tarea.*

*Agradezco también a una persona que conocí hace aproximadamente un año y gracias a la cual mi vida ha cambiado completamente y para bien. Gracias Y, por tu cariño, apoyo, preocupación y comprensión. Gracias por existir en mi vida.*

*A mi familia, por ayudarme a crecer y ser mejor persona cada día, por su incondicional apoyo en todo lo que he tenido que enfrentar y en todo lo que me he propuesto hacer. Por el amor que siempre me han dado, y por simplemente ser mi familia.*

*A todos, mi eterno agradecimiento y cariño.*

*Carlos Yordan González Herrera.*

## Resumen

La Universidad de las Ciencias Informáticas, es un centro de estudios superiores, el cual dispone de tecnología y se caracteriza por el uso de las Tecnologías del Aprendizaje y el Conocimiento en el proceso docente-educativo. Dentro del Plan de estudios de la Universidad, se encuentra la asignatura Sistemas de Bases de Datos I, la cual cuenta con dos habilidades genéricas necesarias para cumplir con los objetivos de la misma. Una de ellas es diseñar sistemas correctos para el almacenamiento de información a partir de la teoría de los Sistemas de Bases de Datos Relacionales. En la actualidad, los estudiantes presentan problemas en la adquisición de esta habilidad, siendo una de las principales causas, la falta de tecnologías que faciliten la práctica. La herramienta utilizada en la asignatura, así como la plataforma educativa *RDB-Learning*, no permiten el diseño de modelos entidad relación basado en la teoría de diseño impartida en clases; por lo cual, no brindan apoyo a los estudiantes para desarrollar dicha práctica. La presente investigación presenta un módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*, el cual permite solucionar ejercicios de diseño a través de la representación de dichos modelos. Este módulo basa su funcionamiento en la tecnología *e-learning*, al permitir la retroalimentación entre sus usuarios. Se muestran los resultados de un conjunto de pruebas realizadas al módulo, con el fin de entregar al cliente una solución libre y confiable, que pueda ser utilizada como apoyo en la asignatura.

**Palabras clave:** bases de datos relacionales, diseño, *e-learning*, modelo entidad relación, retroalimentación

## Índice de contenidos

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1. INTRODUCCIÓN.....	6
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DE LA INVESTIGACIÓN .....	6
1.3. TEORÍA PARA EL DISEÑO DEL MODELO ENTIDAD RELACIÓN.....	8
1.4. RESULTADOS DEL ANÁLISIS DE HERRAMIENTAS INFORMÁTICAS PARA DISEÑAR MER .....	9
1.4.1. <i>Aplicaciones a nivel internacional</i> .....	9
1.4.2. <i>Aplicaciones a nivel nacional</i> .....	11
1.5. ENTORNO DE DESARROLLO DE LA PROPUESTA DE SOLUCIÓN.....	12
1.6. CONCLUSIONES DEL CAPÍTULO .....	18
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN</b> .....	<b>19</b>
2.1. INTRODUCCIÓN.....	19
2.2. ANÁLISIS.....	19
2.2.1 <i>Características de la propuesta de solución</i> .....	19
2.2.2 <i>Modelo Conceptual</i> .....	20
2.3. DISEÑO.....	21
2.3.1. <i>Requisitos funcionales</i> .....	22
2.3.2. <i>Requisitos no funcionales</i> .....	23
2.3.3. <i>Historias de usuario</i> .....	25
2.3.4. <i>Diagramas de clases del diseño</i> .....	26
2.3.5. <i>Diagramas de secuencia</i> .....	28
2.3.6. <i>Estilo arquitectónico</i> .....	30
2.3.7. <i>Patrones de diseño</i> .....	32
2.3.8. <i>Modelo de datos</i> .....	34
2.3.9. <i>Modelo de despliegue</i> .....	37
2.4. CONCLUSIONES DEL CAPÍTULO .....	38
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN</b> .....	<b>39</b>



3.1.	INTRODUCCIÓN.....	39
3.2.	DIAGRAMA DE COMPONENTES .....	39
3.3.	ESTÁNDARES DE CODIFICACIÓN.....	41
3.4.	VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN .....	45
3.4.1.	<i>Pruebas funcionales</i> .....	46
3.4.2.	<i>Pruebas de seguridad</i> .....	51
3.4.3.	<i>Pruebas de usabilidad</i> .....	52
3.4.4.	<i>Pruebas de aceptación</i> .....	53
3.4.5.	<i>Pruebas de carga y estrés</i> .....	56
3.5.	INTERFACES PRINCIPALES DEL MDMER .....	58
3.6.	CONCLUSIONES DEL CAPÍTULO .....	59
	<b>CONCLUSIONES GENERALES .....</b>	<b>60</b>
	<b>RECOMENDACIONES.....</b>	<b>61</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>62</b>
	<b>ANEXOS .....</b>	<b>67</b>
	ANEXO 1. ENTREVISTA AL CLIENTE PARA CONOCER LA NECESIDAD DEL DESARROLLO DE LA PROPUESTA DE SOLUCIÓN Y DEFINIR LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES .....	67
	ANEXO 2. ENCUESTA AL PROFESOR PARA EVALUAR EL NIVEL DE ACEPTACIÓN DEL MDMER .....	68

## Índice de figuras

Figura 1. Representación de un MER .....	9
Figura 2. Modelo Conceptual .....	21
Figura 3. Diagrama de clases de la HU Mostrar ejercicios de diseño.....	27
Figura 4. Diagrama de clases de la HU Crear respuestas a ejercicios de diseño.....	27
Figura 5. Diagrama de secuencia de la HU Mostrar ejercicios .....	28
Figura 6. Diagrama de secuencia de la HU Crear respuesta .....	29
Figura 7. Funcionamiento del MPV en Django aprehendido de Infante (2012).....	30
Figura 8. Organización del MDMER.....	31
Figura 9. Modelo de datos del MDMER .....	36
Figura 10. Modelo de despliegue del MDMER .....	37
Figura 11. Diagrama de componentes del MDMER .....	40
Figura 12. Resultados de las pruebas funcionales .....	50
Figura 13. Resultados de las pruebas de seguridad .....	51
Figura 14. Resultados de la encuesta de aceptación del MDMER .....	55
Figura 15. Interfaz de la página para mostrar un ejercicio.....	58
Figura 16. Interfaz de la página para crear una respuesta .....	59

## Índice de tablas

Tabla 1. Resumen del análisis de los sistemas homólogos.....	11
Tabla 2. Notación de los elementos del MER en el MDMER.....	20
Tabla 3. Relación de requisitos funcionales del módulo.....	22
Tabla 4. HU_5 Mostrar ejercicios de diseño.....	25
Tabla 5. HU_6 Crear respuestas a ejercicios de diseño.....	26
Tabla 6. Descripción de los elementos del diagrama de componentes.....	40
Tabla 7. Estándares de codificación para la implementación del MDMER.....	42
Tabla 8. Estrategia de prueba para el MDBDR.....	45
Tabla 9. Descripción de las variables del caso de prueba 1.....	46
Tabla 10. Caso de prueba del RF1_Crear ejercicios de diseño.....	47
Tabla 11. Resultado de las pruebas de usabilidad.....	52
Tabla 12. Resultado de las encuestas realizadas para pruebas de aceptación.....	55
Tabla 13. Resultado de las pruebas de carga y estrés.....	57

### 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. Según la UNESCO (2017), estas desempeñan un papel cada vez más importante en la forma de comunicarse, aprender y vivir. En la actualidad, las TIC se han desarrollado aceleradamente, lo que ha traído consigo un elevado auge en el desarrollo de la sociedad y de los procesos que en ella se suceden.

Según Granados y otros (2014), el entorno educativo es uno de los que mayores transformaciones ha sufrido con este acelerado desarrollo de las TIC. Es allí donde aparece el concepto de tecnologías del aprendizaje y el conocimiento (TAC), las cuales son la conjunción entre tecnología y metodología, y según Lozano (2011), tratan de orientar hacia usos más formativos de la tecnología, tanto para el estudiante como para el profesor, con el objetivo de aprender más y mejor. Estas van más allá de aprender meramente a usar las TIC, y se apuesta por explotar estas herramientas tecnológicas al servicio del aprendizaje y de la adquisición de conocimiento, por medio de recursos virtuales de aprendizaje (*e-learning*, en inglés).

La introducción de la tecnología *e-learning* en el proceso de enseñanza-aprendizaje, si bien es considerada un reto para los docentes, desde el punto de vista de los estudiantes resulta más atractiva. En Cuba, con el uso de las TAC, se ha aplicado la tecnología *e-learning* en el proceso educativo. Esta vinculación, permite mejorar e innovar los métodos tradicionales de la enseñanza presencial, al brindar diversas modalidades para realizar estudios superiores, ya sea totalmente a distancia o mixto. La Educación Superior en Cuba, juega un destacado papel en la formación de profesionales y la superación permanente, y apuesta por el uso de la tecnología *e-learning* en el proceso educativo.

La Universidad de las Ciencias Informáticas (UCI), es un centro de estudios superiores, el cual dispone de tecnología y se caracteriza por el uso de las TAC en el proceso docente-educativo. En la UCI se emplean diversas herramientas que mediante la práctica facilitan la adquisición de nuevos conocimientos. Entre ellas destacan la plataforma educativa ZERA 2.0<sup>1</sup> y el Repositorio de Objetos de Aprendizaje RHODA.

---

<sup>1</sup> Desarrollada en la UCI, por el centro productivo FORTES, ZERA 2.0 es una plataforma educativa innovadora, interactiva y adaptable basada en híper-entornos de aprendizaje y que reúne características de Sistemas Gestión de Contenidos y del Aprendizaje enriqueciendo y guiando el proceso de enseñanza-aprendizaje en empresas, organizaciones, comunidades e instituciones de cualquier nivel de enseñanza (FORTES, 2015).

En función de cumplir con la misión encomendada, la UCI cuenta con un Plan de Estudio de Ingeniería en Ciencias Informáticas (ICI), dentro del cual se incluye Sistemas de Bases de Datos I (SBD I) como una de las asignaturas esenciales en el segundo año de la carrera. Esta asignatura cuenta con dos habilidades genéricas necesarias para vencer los objetivos que tiene la misma. Una de ellas, es diseñar sistemas correctos para el almacenamiento de información, a través de la representación de modelos entidad relación (MER) basados en la teoría de los Sistemas de Bases de Datos Relacionales (SBDR), reflejada en el plan de estudios “D” de la carrera (Ministerio de Educación Superior, 2014).

Se debe tener en cuenta que con el avance de las TIC y el cúmulo de información circulante que estas generan, el uso de las bases de datos (BD) es cada vez mayor. En la actualidad, cualquier organización que se precie, por pequeña que sea, debe contar con una BD eficaz y bien definida desde sus inicios. Por esta razón, es de suma importancia realizar un correcto diseño de la misma, pues de ello dependerá que los datos estén correctamente actualizados, la información sea exacta y se puedan obtener reportes efectivos y eficientes que permitan a las instituciones tomar decisiones futuras.

En la UCI, dentro de los centros de desarrollo, está presente el rol de diseñador de base de datos y es SBD I, como se menciona con anterioridad, la asignatura encargada de garantizar estos conocimientos. Sin embargo, se evidencia que en el proceso de enseñanza-aprendizaje de la misma, los estudiantes presentan dificultades para diseñar MER, resultado que se muestra en los exámenes de la asignatura donde se mide este tema. Respecto a los tres últimos años académicos (2014-2017), en la Facultad 1 de la UCI, se tiene como resultado que aproximadamente el 62% de los estudiantes cometen errores al responder la pregunta de diseño en dichos exámenes. Entre las principales causas se encuentra la ausencia de tecnologías que faciliten la auto-preparación.

A pesar de existir un espacio para la asignatura en la plataforma ZERA, en ocasiones los profesores dejan a un lado la educación virtual y se enfocan más en la presencial. Ello se debe en gran medida, a que esta plataforma carece de herramientas que permitan conjugar los beneficios de ambas modalidades educativas, en pos de formar la habilidad diseñar MER desde un aprendizaje mezclado. Esto trae como consecuencia la necesidad de emplear una herramienta externa a la plataforma.

La herramienta *DB-Designer*, empleada en la asignatura durante una clase práctica de laboratorio, realiza el diseño de BD a través del modelo relacional (MR) y no a partir de un MER; por lo cual no contribuye al desarrollo de esta habilidad en los estudiantes. Otro de los inconvenientes de esta herramienta es que, al

no ser una aplicación web, no hace uso de las potencialidades de la tecnología *e-learning*. Como consecuencia, no es posible la retroalimentación directa entre sus usuarios, la auto-preparación de los estudiantes guiada, controlada y evaluada por el profesor en espacios-tiempos diferentes, así como poder flexibilizar el ritmo de aprendizaje y el trabajo diferenciado con los estudiantes.

Como posible recurso, se encuentra en desarrollo hoy, en la Facultad 1 de la UCI, la plataforma educativa *RDB-Learning*, un soporte tecnológico para apoyar el proceso de enseñanza-aprendizaje de la asignatura SBD I en la Universidad. Basada en las TAC, la misma debe contribuir, mediante su uso, al desarrollo de las habilidades genéricas de la asignatura a través de las actividades de auto-preparación que realicen los estudiantes. En estos momentos, la plataforma no permite el diseño de MER, lo cual trae consigo, que los estudiantes no puedan usarla como herramienta de apoyo a la práctica de esta materia.

Teniendo en cuenta que el diseño de MER es considerado como la base para el entendimiento de los SBD a partir de su modelación inicial y la importancia que tiene un correcto diseño de una BD, surge el siguiente **problema de investigación**: ¿Cómo contribuir, mediante el uso de las tecnologías del aprendizaje y del conocimiento, al diseño de modelos entidad relación en la asignatura Sistemas de Bases de Datos I en la Facultad 1 de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de la presente investigación va dirigido a las herramientas informáticas para el diseño de modelos entidad relación, y el **campo de acción** a las tecnologías del aprendizaje y el conocimiento para el diseño de modelos entidad relación en la asignatura Sistemas de Bases de Datos I en la Facultad 1 de la Universidad de las Ciencias Informáticas.

Para dar solución al problema mencionado se plantea como **objetivo general**: desarrollar un módulo como tecnología del aprendizaje y el conocimiento para la plataforma *RDB-Learning*, que permita el diseño de modelos entidad relación en la asignatura Sistemas de Bases de Datos I en la Facultad 1 de la Universidad de las Ciencias Informáticas.

En función de cumplir con el objetivo planteado, se formulan las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos que sustentan la investigación, relacionados con el uso de las herramientas informáticas, en función del diseño de modelos entidad relación?
2. ¿Cuál es el estado actual de las herramientas informáticas para el diseño de modelos entidad relación?

3. ¿Qué elementos deben tenerse en cuenta para llevar a cabo el análisis y diseño del módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*?
4. ¿Cómo materializar, en términos de componentes y código fuente, los diseños especificados para el módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*?
5. ¿Qué resultados se obtendrán al validar, a través de una estrategia de pruebas de software, el módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*?

Para responder estas preguntas científicas, es necesario dar cumplimiento a las siguientes **tareas de investigación**:

1. Sistematización de los referentes teóricos que sustentan la investigación, relacionados con el uso de las herramientas informáticas en función del diseño de modelos entidad relación.
2. Análisis del estado actual de las herramientas informáticas para el diseño de modelos entidad relación.
3. Análisis y diseño de las funcionalidades del módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*.
4. Implementación de las funcionalidades del módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*.
5. Validación, a través de una estrategia de pruebas, del módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*.

Los **métodos de investigación** utilizados para dar solución a la presente investigación son:

### **Teóricos:**

**Histórico-Lógico:** Permite una mayor comprensión de la evolución de las herramientas informáticas para el diseño de MER.

**Analítico-Sintético:** Se utiliza con el objetivo de realizar un análisis bibliográfico para establecer las bases teóricas en relación al desarrollo de tecnologías del aprendizaje y el conocimiento para el apoyo del proceso de enseñanza-aprendizaje.

**Modelación:** Es empleada en la representación mediante diagramas de las características, procesos y componentes de la solución propuesta, así como la relación existente entre ellos.

### **Empíricos:**

**Entrevista:** Se emplea en encuentros con el cliente para conocer la necesidad del desarrollo de la propuesta de solución, definir sus funcionalidades e identificar a la vez particulares de cada usuario y las restricciones que se imponen.

**Observación:** Posibilita obtener conocimiento acerca del funcionamiento de los sistemas existentes en la actualidad para el diseño de Modelos Entidad Relación.

La presente investigación está estructurada de la siguiente forma:

**Capítulo1: “Fundamentación teórica”:** En este capítulo se exponen los conceptos asociados al dominio de la investigación y se realiza un análisis de la teoría para el diseño de MER. Además, se analizan algunas herramientas para el diseño de MER reconocidos nacional e internacionalmente y de define el entorno de desarrollo que se utilizará para dar solución al problema planteado en la investigación.

**Capítulo2. “Análisis y diseño de la propuesta de solución”:** En este capítulo se documenta todo el proceso de elaboración del módulo de manera más detallada. Se describen los requerimientos de acuerdo a lo establecido en la metodología utilizada, diagramas de clases del diseño, diagramas de despliegue, lo referente al estilo arquitectónico del módulo, así como los patrones de diseño utilizados y el entorno de despliegue propuesto para desplegarlo.

**Capítulo3: “Implementación y pruebas de la propuesta de solución”:** En este capítulo se detalla la propuesta de solución al problema planteado. Se describe la organización del módulo en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Además, se realiza la estrategia de pruebas definida para el módulo y se muestran interfaces como parte del resultado final.

Con la realización de la presente investigación, se pretende obtener un módulo que permita a los estudiantes realizar actividades prácticas sobre ejercicios de diseño de MER, y a los profesores controlar y evaluar el trabajo de sus educandos, para así contribuir desde las TAC al tema de diseño de MER en la asignatura de SBD I.



# Capítulo 1: Fundamentación Teórica

## 1.1. Introducción

Las herramientas de modelado de datos son muy útiles a la hora de realizar un diseño de datos nuevo o acercarse a la comprensión y estudio de un modelo existente. La capacidad de realizar el diseño de un modelo, visualizándolo de forma gráfica a través de los MER, o de obtener un diagrama y documentación de una base de datos existente, aumenta la eficiencia del trabajo al reducir el tiempo necesario con respecto a una realización manual de dichas tareas de diseño, análisis y estudio. Con el objetivo de lograr una mayor comprensión del alcance de la investigación y esclarecer su objeto de estudio, se exponen en el presente capítulo, los conceptos asociados al dominio de la investigación y se realiza un análisis del estado del arte que la precede. Además, se presenta el entorno de desarrollo a utilizar para dar solución al problema planteado.

## 1.2. Conceptos asociados al dominio de la investigación

A continuación, se relacionan los principales conceptos que ayudan a entender el desarrollo de la investigación.

En la Ingeniería de Software, se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener estas aplicaciones sin distribuir e instalar software a miles de usuarios potenciales (Alegsa, 2015).

En la fase de diseño del ciclo de vida de una aplicación web, dado un problema a resolver, en primer lugar, se debe estudiar la posibilidad de dividirlo en otros más pequeños, llamados sub-problemas. Del mismo modo, también puede ser conveniente fragmentar estos sub-problemas obtenidos, hasta llegar a otros realmente sencillos. A cada sub-problema se le considera parte o **módulo** del problema global (Rodríguez, 2012).

En otro sentido, el aprendizaje electrónico, conocido como **e-learning**, es una modalidad de enseñanza-aprendizaje que consiste en el diseño, puesta en práctica y evaluación de un curso o plan formativo desarrollado a través de redes de ordenadores. Esta puede definirse como una educación o formación ofrecida a individuos que están geográficamente dispersos o separados, o que interactúan en tiempos diferidos del docente, mediante el empleo de recursos informáticos y de telecomunicaciones. Lo característico del *e-learning* es que el proceso formativo tiene lugar, totalmente o en parte, a través de una especie de aula o entorno virtual en el cual tiene lugar la interacción profesor-alumnos, así como las actividades de los estudiantes con los materiales de aprendizaje (e-ABC, 2017).

Por su parte, según Guevara (2015), se conoce como **plataforma de e-learning** o plataforma educativa, a la herramienta usada para distribuir el conocimiento, donde se agrupan un conjunto de equipos y software, y debe funcionar el sistema que se desea desarrollar para las actividades del proceso de enseñanza-aprendizaje. En ella se pueden llevar las estadísticas de las evaluaciones y se incluyen foros y debates, que permiten la retroalimentación entre sus usuarios.

La **retroalimentación**, que se puede lograr a través de la tecnología *e-learning*, debe acompañar al proceso educativo a lo largo de toda su extensión. La intención es dar inmediata y oportuna información al alumno acerca de la calidad de su desempeño, para que realice las correcciones necesarias con el objetivo de mejorar su competencia. La retroalimentación no está restringida sólo a corregir los errores y omisiones que cometa el aprendiz para que se haga cada vez más competente (retroalimentación negativa), si no, también, para que pueda identificar sus aciertos (retroalimentación positiva), como una forma de estimularlo para hacer las cosas cada vez mejor. Es una forma de aumentar su motivación intrínseca para aprender y/o trabajar con mayor eficiencia (Fonseca, 2009).

Otro de los aspectos a tener en cuenta para un mejor entendimiento del problema planteado, es el concepto de **entidad**, la cual puede definirse como un objeto del mundo real que se puede distinguir del resto de los objetos y del que interesan algunas propiedades. En BD, una entidad es la representación de un objeto o concepto del mundo real que se describe en una BD. Las propiedades de los objetos que interesan se denominan **atributos** (Costal, 2014).

**Relación:** En Bases de Datos Relacionales (BDR), una relación o vínculo entre dos o más entidades describe alguna interacción entre las mismas. No siempre las condiciones para establecer vínculos entre dos tablas son iguales, la manera en que se relacionan las tablas entre sí da lugar a comportamientos

diferentes. En la estructura de cualquier BD se pueden encontrar tres tipos de relaciones, las cuales se describen del siguiente modo: uno a muchos, uno a uno, muchos a muchos (Gómez, y otros, 2013).

Un **Modelo Entidad-Relación** se define como un gráfico que representa visualmente las relaciones entre entidades de BDR. Los MER modelan los requisitos de almacenamiento de datos de una organización con tres componentes principales: entidades, atributos y relaciones (Costal, 2014). Estos modelos, siguen una teoría para su diseño, que establece las formas de representación de los elementos que lo componen.

### 1.3. Teoría para el diseño del modelo entidad relación

Dentro del sistema de habilidades (Gómez, 2016) que debe desarrollar el estudiante, este debe ser capaz de: diseñar esquemas conceptuales y lógicos eficientes para representar un fenómeno de la realidad objetiva, lo cual, en términos de la asignatura, es llamado: diseño del MER. Para realizar este diseño, a partir de un problema dado, el estudiante debe ser capaz de identificar los principales elementos que intervienen, sus características y las relaciones existentes entre ellos. Según Mato (2007), los principales elementos que se tienen en cuenta a la hora de realizar un MER son: Entidades (fuerte, débil), atributos (llave, simple, compuesto, multivaluado, derivado), relaciones (binaria, ternaria, recursiva, débil), cardinalidad de las relaciones, generalización/ especialización (con su cubrimiento), agregación.

En la Figura 1, se muestra un MER donde se observa la representación visual de cada uno de estos elementos. El diseño de MER puede ser de gran ayuda para el diseño de BDR. Una de las ventajas de los MER radica en que estos pueden ser comprendidos por personas que no sean especialistas o se inician en el aprendizaje de esta materia. El número de entidades en una BD es, por lo general, considerablemente menor que el número total de elementos de datos. Así, usando el concepto de una entidad, se simplifican las fases de análisis de los requerimientos y el diseño conceptual. Es por ello que los conocimientos sobre este tema deben ser profundizados por los estudiantes durante el aprendizaje de la asignatura SBD I.

Para lograr este fin, se pueden emplear herramientas informáticas que permitan el diseño del MER, las cuales puedan servir a los estudiantes como instrumento para la profundización y consolidación de los conocimientos. A continuación, se realiza un análisis de este tipo de herramientas, empleadas para el diseño de MER.

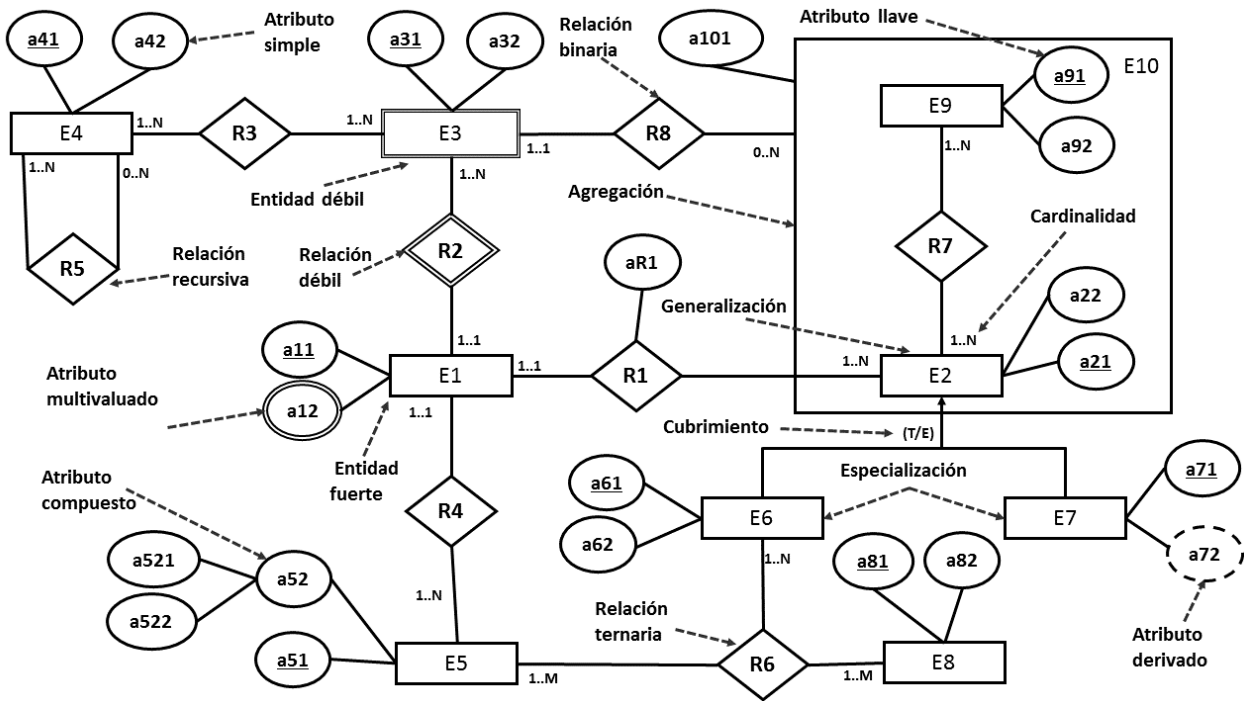


Figura 1. Representación de un MER

#### 1.4. Resultados del análisis de herramientas informáticas para diseñar MER

En la actualidad existen aplicaciones para el diseño de MER. Con el objetivo de identificar ventajas y desventajas de estas, se realiza el siguiente análisis haciendo énfasis en las características y funcionalidades de este tipo de herramientas.

##### 1.4.1. Aplicaciones a nivel internacional

###### DBCASE

Es una herramienta que permite diseñar BD relacionales mediante una interfaz gráfica. El objetivo de la misma es facilitar la creación de las BD mediante los diagramas entidad-relación, para lo que permite comprobar posteriormente su corrección y generar el código asociado a las tablas creadas. La herramienta está orientada principalmente al entorno académico, aunque también es útil en otros ámbitos. La interfaz gráfica se ha diseñado de manera que siga los estándares utilizados en los diagramas entidad-relación de los principales libros de la bibliografía de diseño de BD, de forma que la notación utilizada resulte familiar al

usuario (Mateos, y otros, 2009). Esta herramienta, permite la representación del MER, pero no tiene en cuenta la agregación, como uno de los elementos del mismo, además de que no es una aplicación web, y a pesar de que es utilizada en apoyo a la docencia, no está basada en la tecnología *e-learning*.

### **SUITEBD**

El sistema de software SUITEBD, desarrollado en la Escuela Superior de Cómputo de México D.F., tiene como finalidad apoyar la impartición de BD, en particular de los MER y MR, en alumnos de nivel superior. Este sistema está basado en Internet (es una aplicación web), lo que significa que es accesible desde cualquier computadora conectada a la red, y brinda las ventajas que ello conlleva, como son la disponibilidad y la flexibilidad, entre otras. SUITEBD no sólo es un editor de diagramas, sino que guía a los alumnos en la creación de los mismos a través de su función de Verificación que indica en dónde se pueden encontrar errores (Báez, y otros, 2016). Esta herramienta no tiene en cuenta la agregación, dentro de los elementos del MER, además, se emplea bajo licencia privativa, sin embargo, es una plataforma web *e-learning*, por lo que las tecnologías empleadas en su desarrollo, así como el análisis de su arquitectura, pueden servir de apoyo para la presente investigación.

### **ACME-DB**

Según Soler (2010), ACME-DB es un entorno utilizado como complemento a las clases presenciales en la asignatura de BD de segundo curso de Ingeniería Técnica en Informática de Gestión de la Universidad de Girona. Entre otras características, esta herramienta, es de licencia privativa, soporta SQL como SGBD y facilita el aprendizaje y evaluación de los principales temas de una asignatura de BD. Es una aplicación web, con corrección y *feedback*<sup>2</sup> automáticos, puntuación automática de los ejercicios, así como evaluación automática de los alumnos. Está desarrollada bajo licencia privativa. Permite crear el MER a partir de la teoría de los SBDR, pero sin tener en cuenta la agregación como un elemento del mismo, de igual forma, es una plataforma web *e-learning*, por lo que las tecnologías empleadas en su desarrollo, así como el análisis de su arquitectura, pueden servir de apoyo para la presente investigación.

---

<sup>2</sup> Devolución de una señal modificada a su emisor.

1.4.2. Aplicaciones a nivel nacional

**ERECASE**

ERECASE se basa en el MER Extendido. Para la creación de los esquemas conceptuales la herramienta posee un conjunto amplio de construcciones, entre ellas: entidades fuertes y débiles, interrelaciones de asociación recursivas, binarias y ternarias, interrelaciones ISA<sup>3</sup>, interrelaciones débiles, jerarquías de generalización/especialización y agregación. La agregación es algo poco común en herramientas para la creación de diagramas Entidad - Relación, lo que hace que ERECASE sea singular en este sentido (García, y otros, 2008). Esta aplicación permite la representación del MER, y tiene en cuenta todos sus elementos, excepto la representación de los tipos de atributos, los cuales, además de que no los representa visualmente en el diagrama, solo pueden ser insertados como atributos simples o llave. Otra desventaja de la aplicación, es que la misma no es web, por lo que no es posible hacer uso de la tecnología *e-learning*.

El análisis realizado sobre las aplicaciones para diseño de MER, permitió realizar un resumen (ver Tabla 1) teniendo en cuenta los siguientes parámetros:

- **Licencia:** se refiere al estado jurídico de la aplicación en cuanto a su uso, modificación y distribución, esta puede ser pública, en aquellas que no necesitan un pago para ser utilizadas; o privativa, en aquellas que si lo requieren.
- **Web:** se refiere a si la herramienta es una aplicación web o no.
- **E-Learning:** especifica si la aplicación es o forma parte de una plataforma educativa.

Tabla 1. Resumen del análisis de los sistemas homólogos

Herramienta	Licencia	Web	<i>E-Learning</i>
DBCASE	pública	no	no
SUITEDB	privativa	si	si
ACME-DB	privativa	si	si
ERECASE	privativa	no	no

<sup>3</sup> Es una relación de dependencia entre abstracciones (tipos, clases), donde una clase A es una subclase de otra clase B (y así B es una superclase de A) (Koopman, 2016).

A partir del análisis realizado, y del resumen antes presentado, el autor arriba a las siguientes conclusiones:

- A pesar de que todas permiten la representación del MER, la herramienta cubana ERECASE es la única que permite representar la agregación, como otro elemento del MER.
- La herramienta bajo licencia pública, no es una herramienta web, por lo que no hace uso de las potencialidades de la tecnología *e-learning*.
- Las herramientas SUITEDB Y ACME-DB, que son aplicaciones web *e-learning*, son privativas.
- La herramienta cubana analizada, además de que contempla la necesidad de respetar los derechos de autor y por tanto no está disponible para su actualización atendiendo a otras investigaciones, no hace uso de las potencialidades de la tecnología *e-learning*, al ser una aplicación de escritorio.
- Ninguna de estas herramientas puede ser integrada a la plataforma *RDB-Learning*.

Por todo lo antes expuesto, se puede resumir que las herramientas analizadas no cumplen con el objetivo de la presente investigación, pues ninguna reúne todos los requisitos de ser una aplicación web, *e-learning*, gratuita y que permita el diseño de MER, a partir de la teoría de los SBDR, donde se tengan en cuenta todos los elementos del modelo que propone Mato (2007). Sin embargo, el análisis realizado posibilitó la identificación de funcionalidades y tecnologías que pueden contribuir al desarrollo de la propuesta de solución que se propone en la investigación.

### 1.5. Entorno de desarrollo de la propuesta de solución

Teniendo en cuenta que el sistema a desarrollar es un módulo, este debe ser implementado en el mismo entorno de desarrollo del sistema al cual se debe integrar. Luego de analizar la documentación relacionada a la plataforma *RDB-Learning*, la cual documenta las herramientas, así como la metodología de desarrollo empleada, se especifica el uso del siguiente entorno de desarrollo:

#### LENGUAJE DE MODELADO

El lenguaje de modelado es cualquier lenguaje informático gráfico o textual que provee el diseño y construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y marcos (Techopedia, 2017). Como lenguaje de modelado se utiliza el Lenguaje Unificado de Modelado (*UML* por sus siglas en inglés) en su versión 2.1, que consiste en un lenguaje diseñado para visualizar, especificar, construir y documentar software orientados a objetos (Martínez, y otros, 2014).

### HERRAMIENTA DE MODELADO

Dentro de las principales herramientas para el modelado se encuentran herramientas *CASE* definidas por Araujo y otros (2013) como herramientas informáticas que asisten al diseñador en algunas de las actividades relacionadas con el desarrollo de un sistema (requerimientos, análisis, diseño, codificación y pruebas).

*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 la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Para la presente investigación, se utiliza esta herramienta en su versión 8.0.

### HERRAMIENTA PARA CONTROL DE VERSIONES

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante (Saldaña, 2017). Para garantizar el control de versiones en el sistema, y a su vez, la continua integración del módulo a la plataforma *RDB-Learning*, se emplea el repositorio GitLab, disponible en la UCI en la dirección <http://codecomunidades.prod.uci>, para esto, se propone el uso del cliente Git en su versión 2.8, el cual, según Saldaña (2017):

- Es un sistema de control de versiones distribuido
- No depende de acceso a la red o un repositorio central
- Está enfocado a la velocidad, uso práctico y manejo de proyectos grandes

### LENGUAJE DE PROGRAMACIÓN (LADO DEL SERVIDOR)

Un lenguaje de programación 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, y otros, 2014). Para el desarrollo del módulo se emplea el lenguaje *Python* en su versión 2.7.11, el cual 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 (Python, 2017).

### MARCO DE TRABAJO

Un marco de trabajo o *framework* puede definirse, según *TeachTerms* (2013) como una plataforma para desarrollar aplicaciones de software. Proporciona una base en la que los desarrolladores de software



pueden crear programas para una plataforma específica. Esto agiliza el proceso de desarrollo ya que los programadores no necesitan reinventar la rueda cada vez que desarrollan una nueva aplicación.

Para el desarrollo del módulo se utiliza el marco de trabajo *Django* en su versión 1.9.5, 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 e incluye un servidor web ligero que se puede usar mientras se desarrolla (Django, 2014).

### LENGUAJE DE PROGRAMACIÓN (LADO DEL CLIENTE)

El lenguaje para el lado del cliente, empleado en el desarrollo de la plataforma *RDB-Learning*, y por tanto propuesto para el desarrollo del módulo, es *JavaScript*, en su versión 1.8.5. Este, es un lenguaje interpretado que permite incluir macros en páginas web. Estas macros se ejecutan en el ordenador del visitante de dichas páginas, y no en el servidor, algo muy interesante, pues los servidores web suelen estar sobrecargados, mientras que las computadoras de los usuarios no suelen estarlo (Gutiérrez, 2009).

### LIBRERÍAS

Una librería es un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones. Las declaraciones de las funciones utilizadas en estas librerías, junto con algunas macros y constantes predefinidas que facilitan su utilización, se agrupan en ficheros de nombres conocidos que suelen tener las extensiones lib, bpl, a, dll, js, etc. Las librerías utilizadas en el desarrollo del módulo son:

#### JQuery

*JQuery* es una biblioteca multiplataforma de *JavaScript*, que permite simplificar la manera de interactuar con los documentos *HTML*<sup>4</sup>, manipular el árbol *DOM*<sup>5</sup>, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX*<sup>6</sup> a páginas web. Es además un software libre y de código abierto, permitiendo su uso en proyectos tanto libres como privados. *JQuery*, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en *JavaScript* que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y

---

<sup>4</sup> Del inglés *HyperText Markup Language* (Lenguaje de marcas de hipertexto).

<sup>5</sup> Del inglés *Document Object Model* (Modelo de Objetos del Documento o Modelo en Objetos para la Representación de Documentos)

<sup>6</sup> Del inglés, acrónimo de JavaScript asíncrono y XML.

espacio (Gutiérrez, 2009). Para la implementación del módulo se emplea la librería *jQuery* en su versión 3.1.1.

### **GoJS**

También es empleada la librería *GoJS* en su versión 1.7.5. Esta, es una biblioteca de *JavaScript* con múltiples funciones para implementar diagramas interactivos personalizados y visualizaciones complejas en navegadores y plataformas web modernos. Facilita la construcción de diagramas con nodos, enlaces y grupos complejos con plantillas y diseños personalizables. Ofrece, además, muchas características avanzadas para la interactividad del usuario, tales como arrastrar y soltar, copiar y pegar, edición de texto en el lugar, información sobre herramientas, menús contextuales, diseños automáticos, plantillas, vinculación y modelos de datos, administración de estados transaccionales y deshacer, paletas, vistas generales, controladores de eventos, comandos y un sistema de herramientas extensible para operaciones personalizadas (GoJS, 2017).

### **FORMATO DE INTERCAMBIO DE DATOS**

*JSON*, acrónimo de *JavaScript Object Notation*, es un formato de texto ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de *JavaScript*, aunque hoy, debido a su amplia adopción como alternativa a *XML*, se considera un formato de lenguaje independiente. Una de las ventajas de *JSON* como formato de intercambio de datos es la sencillez para escribir un analizador sintáctico (*parser*) de *JSON*. En *JavaScript*, un texto *JSON* se puede analizar fácilmente usando la función *eval()*, lo cual ha sido fundamental para que *JSON* haya sido aceptado por parte de la comunidad de desarrolladores *AJAX*, debido a la ubicuidad de *JavaScript* en casi cualquier navegador web (Gutiérrez, 2009). Por lo antes expuesto, se propone el uso del formato *JSON 3* para el desarrollo del módulo.

### **ENTORNO INTEGRADO DE DESARROLLO (por sus siglas en inglés *IDE*)**

*JetBrains PyCharm* es un entorno de desarrollo integrado de código abierto y multiplataforma desarrollado por *JetBrains*, que se utiliza para programar en *Python*. Proporciona un análisis de código, depuración gráfica, probador de unidad integrada y apoya el desarrollo web con el marco de trabajo *Django*. El mismo incluye heurísticas sofisticadas para determinar lo que cada tipo de variable es, y proporcionar sugerencias de autocompletado para dichas variables. También se puede aprovechar la información en tiempo de

ejecución cuando se ejecuta la aplicación con el depurador integrado (PyCharm, 2015). Para la implementación del módulo se utiliza el *IDE PyCharm* en su versión 2016.2.3.

### **SERVIDOR DE BASES DE DATOS**

*PostgreSQL* es un servidor de bases de datos que incorpora el modelo relacional para sus BD 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 (PostgreSQL, 2014). Para la gestión de la BD del sistema se emplea *PostgreSQL* en su versión 9.4.

### **SERVIDOR WEB**

Para el alojamiento del sistema se emplea el servidor web *Apache 2.2*. Según Mifsuf (2013), *Apache* 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*. Es un servidor robusto y con un ciclo de desarrollo muy rápido, gracias a la gran cantidad de colaboradores con los que dispone.

### **MÓDULO DE APACHE PARA PYTHON**

#### ***Web Server Gateway Interface (WSGI)***

*WSGI* es una interface simple y universal entre los servidores web y las aplicaciones web o *frameworks*. Es similar a la especificación *Java Servlet* o *ASP/ASP.NET*. En general, es mucho más simple que dichas especificaciones, y se basa en el estándar *CGI* con mejoras para hacerla reentrante y persistente (Python, 2017). Para la propuesta de solución se propone usar esta interfaz en su versión 4.5.10.

### **HERRAMIENTAS PARA PRUEBAS DE SOFTWARE**

Las pruebas de software, son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada (Pressman, 2010). En el caso de la propuesta de solución de la investigación, este será validado a través de varios tipos y técnicas de pruebas, algunas de ellas de forma manual y otras mediante el uso de herramientas que permiten la realización de dicha tarea. Entre las herramientas existentes, se pretende utilizar las herramientas *Acunetix Web Vulnerability Scanner* y *Apache JMeter*, para las pruebas de seguridad y de carga y estrés, respectivamente.

*Acunetix* realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección SQL, *Cross site scripting* y otras vulnerabilidades que puedan ser explotadas por *hackers*.

*Apache JMeter* se caracteriza por ser libre de uso y mostrar los resultados de las pruebas en una amplia variedad de informes y gráficas, con gran cantidad de variables que permiten interpretar los resultados desde diferentes puntos de vista.

### METODOLOGÍA DE DESARROLLO

La metodología ágil *AUP*, es una versión simplificada del Proceso Unificado de *Rational (RUP)*. Esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en *RUP*. El *AUP* aplica técnicas ágiles incluyendo (Sánchez, 2015):

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad

Según Sánchez (2015), al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.), exigiéndose así que el proceso sea configurable, se decide hacer una variación de la metodología *AUP*, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Esta variación está unida al modelo CMMI-DEV v1.3<sup>7</sup>, para garantizar las buenas prácticas en función de un software de calidad. La plataforma *RDB-Learning*, a la cual debe integrarse el módulo propuesto, fue desarrollada bajo esta metodología. Para el desarrollo de la propuesta de solución, se decide usar la metodología *AUP*, en su variación para la UCI, teniendo en cuenta todo lo antes expuesto.

---

<sup>7</sup> Colección de buenas prácticas de desarrollo procedentes de la industria y del gobierno. Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software (SEI, 2010).

### 1.6. Conclusiones del capítulo

En este capítulo se han abordado los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, en tal sentido se puede arribar a las siguientes conclusiones:

1. La definición de los principales conceptos asociados al dominio de la presente investigación y las relaciones entre estos, permite alcanzar una mayor comprensión de la propuesta de solución.
2. El análisis de los principales elementos de la teoría para el diseño de modelos entidad relación, permite definir cómo deben ser tratados los mismos desde la propuesta de solución.
3. El análisis de los sistemas homólogos, permite identificar las tendencias en cuanto al desarrollo de herramientas informáticas para el diseño de MER, e identificar las deficiencias que impiden sean utilizadas para el estudio de la asignatura SBD I en la UCI.
4. El análisis de la documentación de la plataforma *RDB-Learning*, sobre la metodología de desarrollo, así como las herramientas, tecnologías y lenguajes de programación utilizados en su implementación, permite especificar el ambiente de desarrollo para la propuesta de solución.

## Capítulo 2: Análisis y diseño de la propuesta de solución

### 2.1. Introducción

Una de las prioridades cuando se desea desarrollar un software, es establecer un entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr (Labrada, y otros, 2013), realizando un correcto análisis y diseño de dicho sistema. 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 *AUP-UCI*. Se detallan las características del módulo, se especifican los requisitos funcionales y no funcionales del mismo y se presentan los artefactos generados para dar solución al problema planteado en la investigación.

### 2.2. Análisis

En esta fase se presentan las características generales del módulo y se realiza un análisis de los conceptos fundamentales asociados al proceso de enseñanza – aprendizaje del diseño de MER en la asignatura SBD I en la UCI los cuales se relacionan a través de un modelo conceptual.

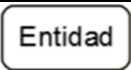
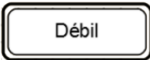




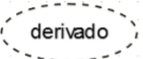



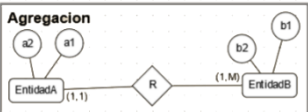
#### 2.2.1 Características de la propuesta de solución

El módulo para diseño de modelos entidad relación (MDMER) en la plataforma *RDB-Learning*, se enmarca dentro de las TAC para el diseño de MER. Una de las características del módulo, es que el mismo permite a los usuarios (estudiantes) resolver ejercicios de diseño de MER.

Para lograr esto, el módulo estará integrado a una plataforma web educativa que permite la selección de los ejercicios disponibles, la visualización de los mismos y su posterior realización. Una vez creada la respuesta, los estudiantes tienen la posibilidad de mandar su respuesta de forma privada a ser revisada por el profesor o de enviarla de forma pública a un foro y que se emitan comentarios sobre la misma, siempre y cuando el ejercicio haya estado asociado a dicho foro. Una vez revisada la respuesta por el profesor, este puede emitir una evaluación y un criterio y enviarla al estudiante, también a través de la plataforma.

Dicho módulo cuenta con un área de trabajo que permite representar en un lienzo los elementos que conforman el diseño de un MER. En la Tabla 2, se muestra la notación de los elementos del MER en el MDMER, basado en la notación de Elmasri y Navathe (Ramez, y otros, 2015), con adaptaciones por la notación que se imparte en las clases de SBD I en la UCI a partir de la teoría de los SBDR:

Tabla 2. Notación de los elementos del MER en el MDMER

Símbolo	Significado
	Entidad fuerte
	Entidad débil
	Atributo llave
	Atributo simple
	Atributo compuesto
	Atributo multivaluado
	Atributo derivado
	Relación
	Relación débil
	Generalización/Especialización
	Agregación

En los siguientes epígrafes y sub-epígrafes se describen artefactos que permiten conceptualizar y entender con claridad el funcionamiento del MDMER.

### 2.2.2 Modelo Conceptual

Un modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos. Puede ser visto, también como una representación de las cosas, entidades, ideas, conceptos u objetos del “mundo real” o dominio de interés (Sommerville, 2011).

Para lograr un mejor entendimiento de los procesos que requieren informatización, se realizó un modelo conceptual (ver Figura 2), con un total de doce (12) clases y doce (12) relaciones, el cual recoge y describe los conceptos más importantes dentro del contexto del módulo, así como las relaciones entre ellos.

Como parte del modelo se aprecian un **estudiante** y un **profesor**, como actores principales del proceso. El profesor imparte la asignatura **SBD I**, que puede ser cursada por varios estudiantes e impartida por varios profesores, un estudiante en la asignatura es atendido por un solo profesor, el cual puede atender a varios estudiantes. Dentro de la asignatura SBD I se encuentra el diseño del **MER**, el cual está compuesto por varios **elementos**: **entidades**, **atributos**, **relaciones**, las cuales tienen una **cardinalidad**, la **generalización/especialización**, con su respectivo **cubrimiento**, y la **agregación**.

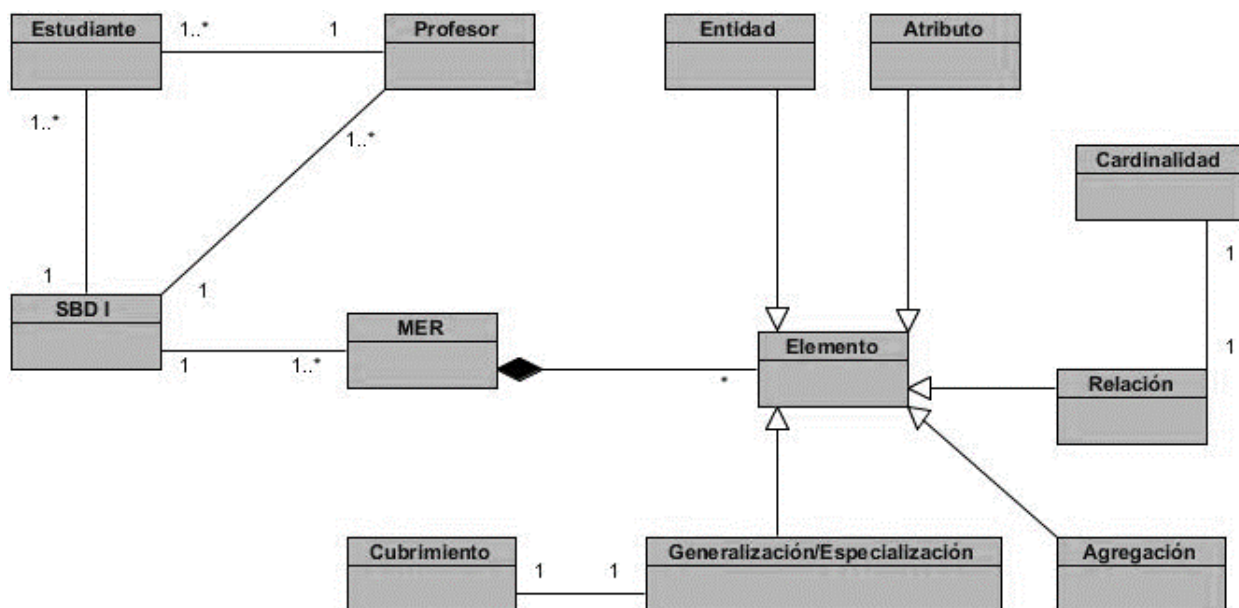


Figura 2. Modelo Conceptual

### 2.3. Diseño

Modelar el sistema posibilita reflejar las propiedades o restricciones que este debe cumplir, a fin de satisfacer las necesidades del cliente (Labrada, y otros, 2013). En esta fase se presentan los requisitos funcionales y no funcionales del módulo, además de los respectivos artefactos generados, a partir de la metodología seleccionada.



### 2.3.1. Requisitos funcionales

Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que el sistema no debe hacer (Sommerville, 2011).

Después del encuentro con el cliente, y a través de una encuesta (ver Anexo 1) aplicada al mismo, se obtuvo un total de dieciocho (18) requisitos funcionales, a los cuales se les asignó una prioridad teniendo en cuenta la importancia fijada por el cliente a partir de sus necesidades. Los mismos se especifican en la siguiente tabla:

Tabla 3. Relación de requisitos funcionales del módulo

Requisito	Prioridad	Requisito	Prioridad
<b>RF1:</b> Crear ejercicios de diseño	Alta	<b>RF2:</b> Eliminar ejercicios de diseño	Media
<b>RF3:</b> Actualizar ejercicios de diseño	Baja	<b>RF4:</b> Listar ejercicios de diseño	Media
<b>RF5:</b> Mostrar ejercicios de diseño	Alta	<b>RF6:</b> Crear respuestas a ejercicios de diseño	Alta
<b>RF7:</b> Eliminar respuestas a ejercicios de diseño	Media	<b>RF8:</b> Actualizar respuestas a ejercicios de diseño	Baja
<b>RF9:</b> Mostrar respuestas a ejercicios de diseño	Alta	<b>RF10:</b> Guardar respuesta a ejercicios de diseño	Alta
<b>RF11:</b> Enviar respuesta a ejercicios de diseño al profesor	Alta	<b>RF12:</b> Publicar respuesta a ejercicios de diseño en el foro	Media

Requisito	Prioridad	Requisito	Prioridad
<b>RF13:</b> Calificar respuesta a ejercicios de diseño enviadas al profesor	Alta	<b>RF14:</b> Crear comentarios a respuestas publicadas en un foro	Alta
<b>RF15:</b> Eliminar comentarios a respuestas publicadas en un foro	Media	<b>RF16:</b> Actualizar comentarios a respuestas publicadas en un foro	Baja
<b>RF17:</b> Mostrar comentarios a respuestas publicadas en un foro	Media	<b>RF18:</b> Responder a comentarios a respuestas publicadas en un foro	Media

### 2.3.2. Requisitos no funcionales

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del mismo (Sommerville, 2011).

El autor considera a los Requisitos no funcionales (RnF) como requerimientos de calidad y para ellos se contemplarán las características del estándar de calidad internacional ISO/IEC 9126-1 (Rojo, y otros, 2012). Distribuidos en especificaciones de usabilidad, confiabilidad, portabilidad, eficiencia, funcionalidad y mantenibilidad, se obtuvo un total de dieciséis (16) requisitos no funcionales, los cuales se relacionan a continuación:

- **Usabilidad:**

**RnF 1:** El MDMER en la plataforma *RDB-Learning* deberá ser una aplicación web

**RnF 2:** El módulo deberá contar con una ayuda para los usuarios, la cual explicará en detalles cómo usar el sistema y los conceptos relacionados con el diseño del MER

**RnF 3:** La aplicación debe adaptarse a las estrategias marcarias de la UCI

**RnF 4:** La aplicación debe presentar una interfaz agradable e intuitiva para el usuario

- **Confiabilidad:**

**RnF 5:** El sistema debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario

- **Portabilidad:**

**RnF 6:** La propuesta de desarrollo debe ser capaz de ejecutarse en los navegadores empleados en la UCI, así como adaptar su interfaz a cualquier dispositivo, ya sea una computadora, tableta o teléfono celular

- **Eficiencia:**

**RnF 7:** El módulo debe permitir que los usuarios interactúen con él de manera concurrente

**RnF 8:** El tiempo de demora de una petición al servidor debe ser menor de cinco (5) segundos aproximadamente

- **Funcionalidad:**

**RnF 9:** La aplicación debe gestionar y requerir información de usuarios para su uso

**RnF 10:** La información manejada por el módulo estará protegida de accesos no autorizados

**RnF 11:** Ante los errores que puedan ocasionarse en el módulo no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad

**RnF 12:** El módulo deberá ser utilizado por los usuarios con dominio uci.cu

**RnF 13:** El módulo deberá ser capaz de cerrar la sesión del usuario una vez pasado diez (10) minutos de inactividad

- **Mantenibilidad:**

**RnF 14:** Se debe hacer uso de los estándares de codificación definidos para la plataforma *RDB-Learning*

**RnF 15:** Se permitirá realizar modificaciones posteriores para adaptar mejoras al módulo o en caso que cambien las necesidades de los clientes

**RnF 16:** El software estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse

### 2.3.3. Historias de usuario

La metodología *AUP-UCI*, en su escenario 4 para la disciplina Requisitos, genera como uno de sus artefactos a las Historias de Usuario (HU) (Sánchez, 2015), que consiste en una técnica para encapsular 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 (Abraira, 2016).

Para la presente investigación se generaron un total de dieciocho (18) HU, a continuación, solo se muestran dos (2) de ellas, pertenecientes a los RF 5 y 6 respectivamente, especificados en el sub-epígrafe 2.3.1.

Tabla 4. HU\_5 Mostrar ejercicios de diseño

Historia de Usuario	
<b>Número:</b> HU_5	<b>Nombre:</b> Mostrar ejercicios de diseño
<b>Programador responsable:</b> Carlos Y. González Herrera	<b>Iteración asignada:</b> 1
<b>Prioridad:</b> Alta	
<b>Descripción:</b> Permite que, a partir de una lista de ejercicios disponibles, se muestren los detalles del que sea seleccionado.	
<b>Observaciones:</b> Esta funcionalidad podrá ser accedida por estudiantes y profesores. El contenido del ejercicio se mostrará en los detalles del mismo y en un panel mientras el usuario se encuentre en el área de diseño.	

Tabla 5. HU\_6 Crear respuestas a ejercicios de diseño

Historia de Usuario	
<b>Número:</b> HU_6	<b>Nombre:</b> Crear respuestas a ejercicios de diseño
<b>Programador responsable:</b> Carlos Y. González Herrera	<b>Iteración asignada:</b> 1
<b>Prioridad:</b> Alta	
<b>Descripción:</b> Consiste en responder los ejercicios disponibles para diseño de MER y la respuesta será la representación del MER correspondiente.	
<b>Observaciones:</b> Tendrán acceso los estudiantes y profesores. Para la representación del MER debe tenerse en cuenta todos los elementos que lo conforman, por lo que el diseño debe permitir la representación de cada uno de ellos. Dígase entidades, atributos (simples, compuestos, derivados y multivaluados), atributos llave, relaciones, cardinalidad de las mismas, entidades débiles, relaciones recursivas, generalizaciones con su respectivo cubrimiento, especializaciones y agregaciones. Durante la realización de la respuesta, se debe permitir insertar todos estos elementos en el área de trabajo, y una vez insertados podrán ser modificados o eliminados.	

#### 2.3.4. Diagramas de clases del diseño

Los diagramas de clase (DC) pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre dichas clases (Sommerville, 2011). Los diagramas de clases del diseño con estereotipos web, describen gráficamente las especificaciones del modelo, la vista y la plantilla de las historias de usuario descritas en el sub-epígrafe 2.3.3. Estas representaciones contienen información acerca de las clases, asociaciones, atributos, métodos y dependencias.

Para la presente investigación se generaron un total de dieciocho (18) DC, a continuación, se muestran dos (2) de ellos, pertenecientes a los RF 5 y 6 respectivamente, especificados en el sub-epígrafe 2.3.1.

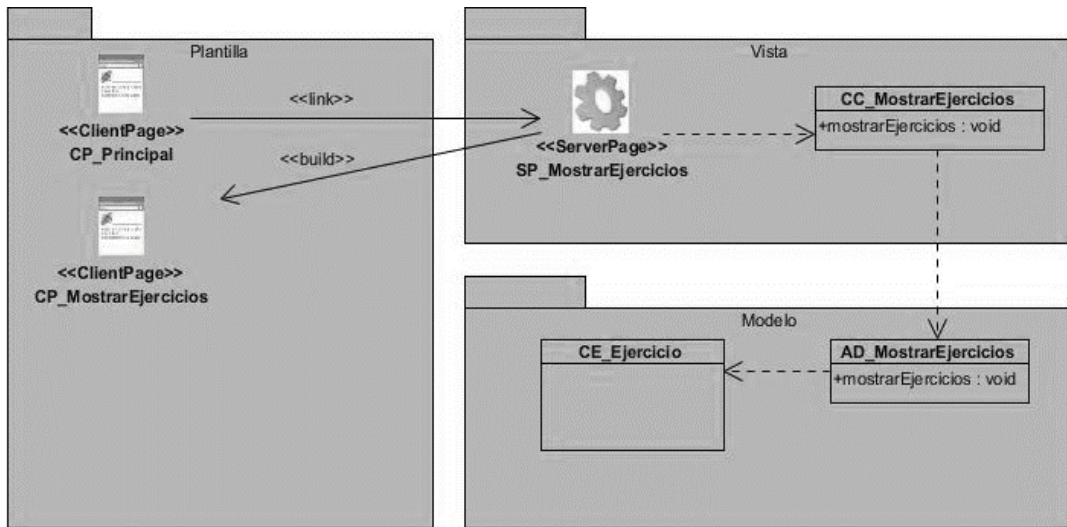


Figura 3. Diagrama de clases de la HU Mostrar ejercicios de diseño

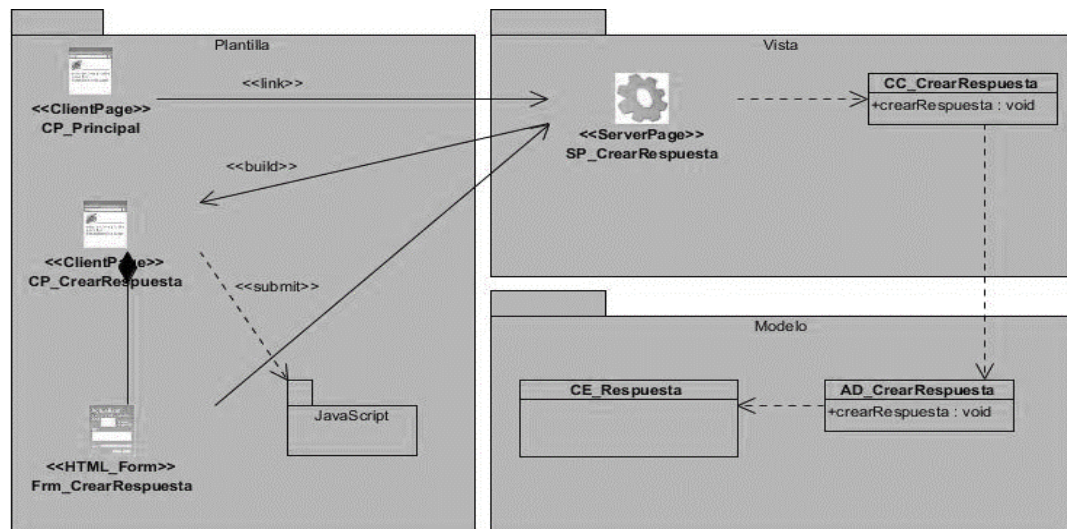


Figura 4. Diagrama de clases de la HU Crear respuestas a ejercicios de diseño

2.3.5. Diagramas de secuencia

Los diagramas de secuencia (DS) en el UML se usan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí (Sommerville, 2011). Para la presente investigación se generaron un total de ocho (8) DS, relacionados con los requisitos de prioridad alta especificados en el sub-epígrafe 2.3.1. A continuación, solo se muestran dos (2) de ellos, pertenecientes a los RF 5 y 6 respectivamente, especificados en el sub-epígrafe 2.3.1.

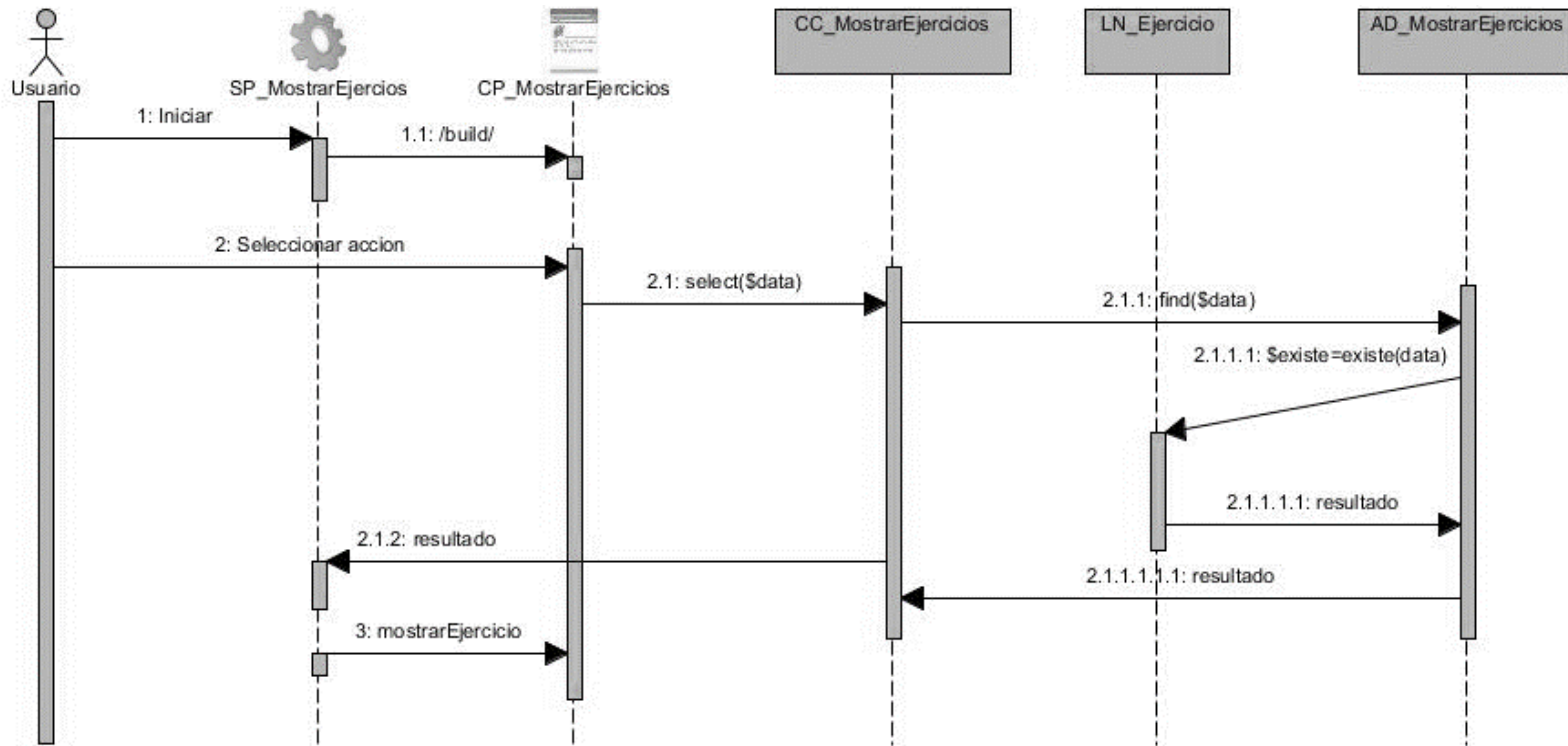


Figura 5. Diagrama de secuencia de la HU Mostrar ejercicios

## Capítulo 2: Análisis y diseño de la propuesta de solución

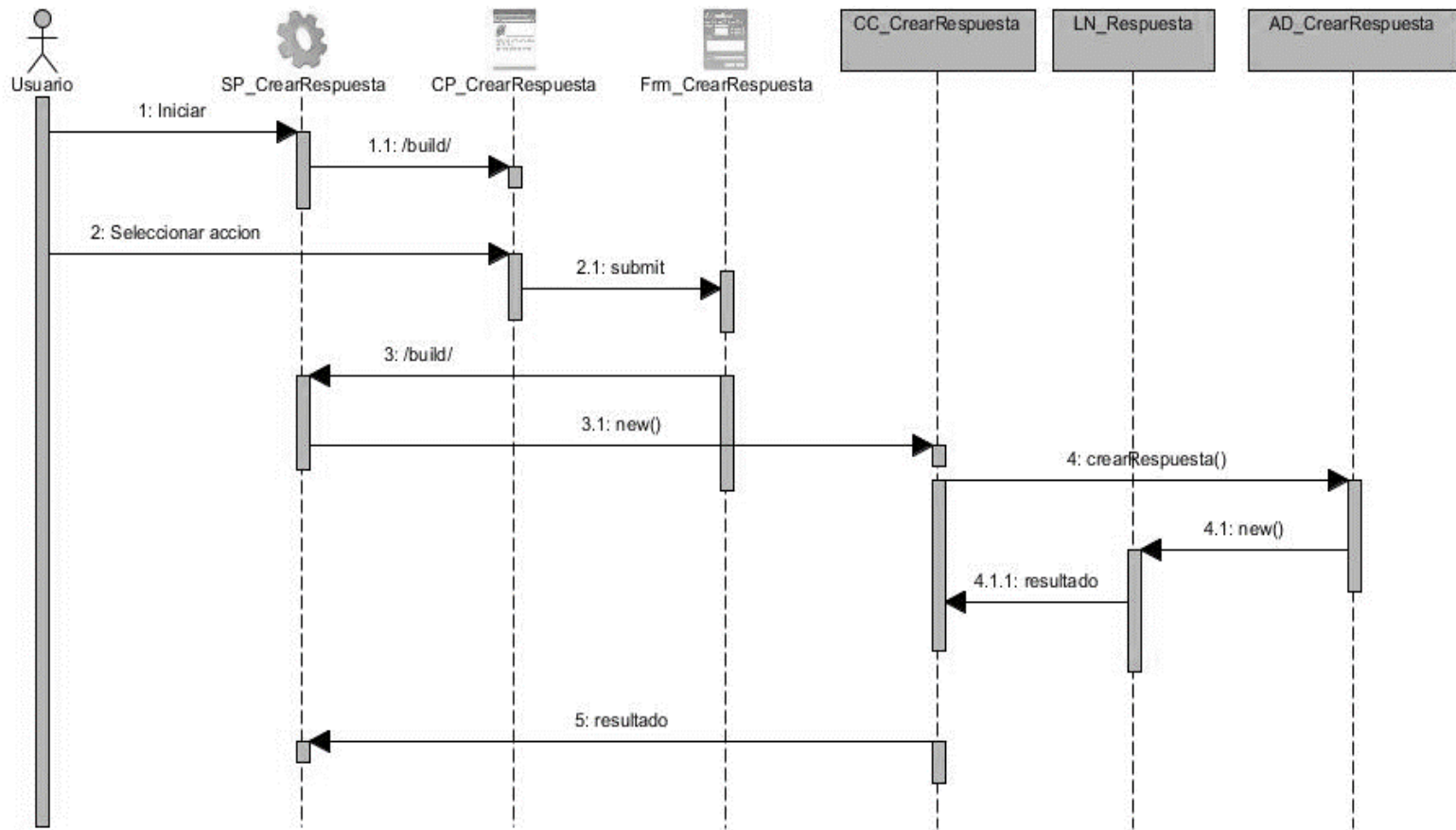


Figura 6. Diagrama de secuencia de la HU Crear respuesta



### 2.3.6. Estilo arquitectónico

El estilo arquitectónico de un sistema puede ser visto como las estructuras del sistema que comprenden a los elementos de software, las propiedades visibles externamente de dichos elementos, y las relaciones entre ellos (Carignano, 2016).

#### Modelo – Plantilla – Vista

Según Infante (2012), *Django* es un marco de trabajo MTV del inglés *Model-Template-View* que es una modificación de MVC (Modelo-Vista-Controlador) debido a que los desarrolladores del marco de trabajo 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 sigue siendo Modelo (M) en *Django*
- La vista pasa a llamarse Plantilla (P) en *Django*
- El controlador pasa a llamarse Vista (V) en *Django*

A partir de lo anterior, se asumirá la arquitectura Modelo-Plantilla-Vista (MPV) para *Django*, de la cual se explica su funcionamiento a continuación, el cual se puede observar en la Figura 7:

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

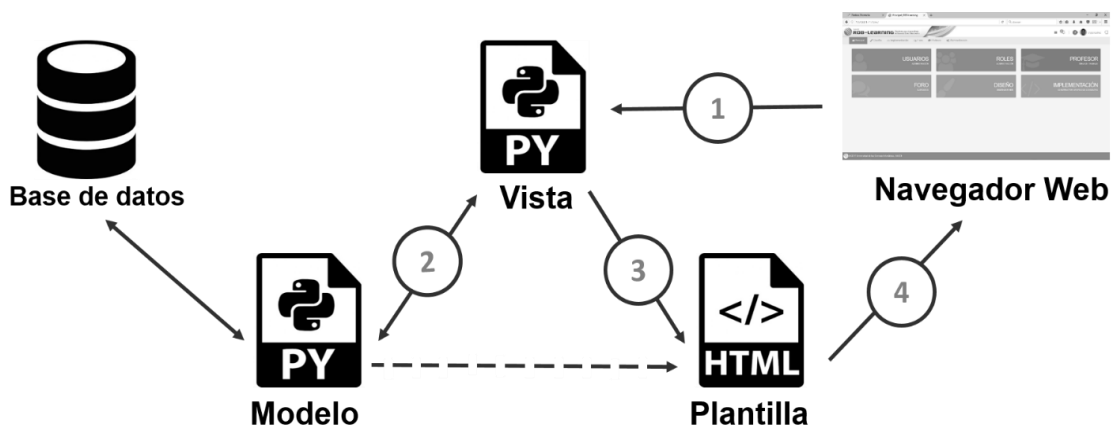


Figura 7. Funcionamiento del MPV en Django aprehendido de Infante (2012)

A partir de lo planteado anteriormente, se deriva lo siguiente (Infante, 2012):

- **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 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*<sup>8</sup>.
- **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.

La figura 8, muestra los elementos del MDMER organizados a partir del estilo arquitectónico antes mencionado y la utilización del marco de trabajo *Django*.

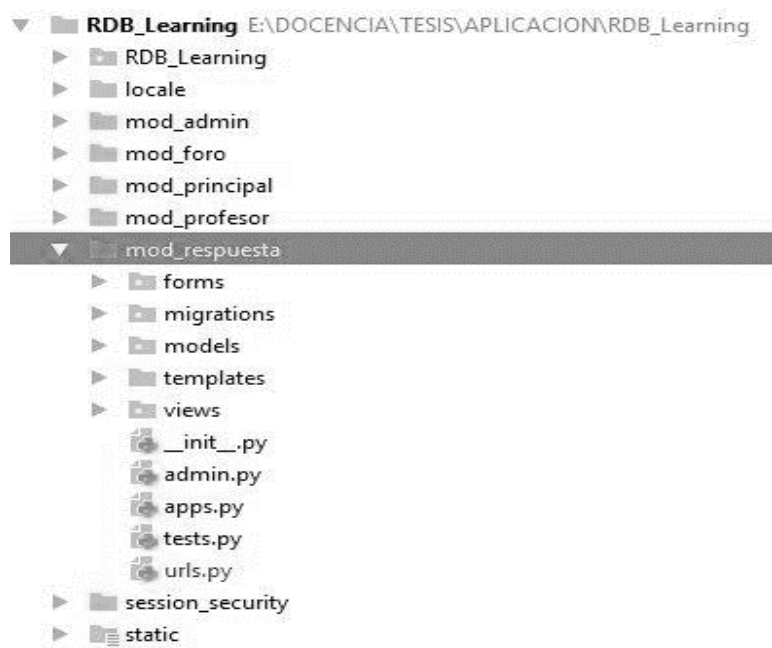


Figura 8. Organización del MDMER

<sup>8</sup> Es la parte del desarrollo web que se dedica de la parte frontal de un sitio web, en pocas palabras, se encarga 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 (Abraira, 2016).

### 2.3.7. Patrones de diseño

Los diseñadores expertos en orientación a objetos (y también otros diseñadores de software), van formando un amplio repertorio de principios generales y de expresiones que los guían a crear el software. A unos y a otras podemos asignarles el nombre de patrones, si se codifican en un formato estructurado que describe el problema y su solución, y si se les asigna un nombre (Larman, 2004). A continuación, se presentan los patrones utilizados en el desarrollo del MDMER en la plataforma *RDB-Learning*.

#### **Patrones Generales de Software para la Asignación de Responsabilidades (GRASP):**

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:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Con el uso de este patrón se alientan las definiciones de clase “sencillas” y más cohesivas que son más fáciles de comprender y de mantener. Brinda soporte a una alta cohesión (Visconti, y otros, 2011).

En el módulo, este patrón se utiliza, por ejemplo, para determinar los estudiantes que han resuelto un problema determinado. La clase entidad **Respuesta**, es la que tiene la información necesaria para brindar esta información, por tanto, es la clase experta en información.

- **Creador:** Se encarga de darle a la clase A la responsabilidad de crear objetos de la clase B. En este caso A es creador de los objetos B (Visconti, y otros, 2011). El propósito fundamental de este patrón, es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento (Larman, 2004).

Este patrón se puede evidenciar en el uso de la función **Profile** la cual se encarga de mostrar los datos del usuario cuyo identificador coincide con el pasado por parámetro para esto crea una instancia del modelo *User* y luego es devuelto a la plantilla.

- **Bajo acoplamiento:** Es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras clases.

Este patrón ya viene incluido con Django (Django, 2014) que permite un bajo acoplamiento entre las piezas, lo que evita las dependencias, por ejemplo, a la hora de realizar cambios en las configuraciones de las *URL*<sup>9</sup>, en la BD, plantillas HTML, etc., basta solo con realizarlo una sola vez.

- **Alta cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Las mismas poseen un número relativamente pequeño de responsabilidades, definiendo así que cada clase realice solo las funcionalidades para las cuales fueron creadas, generando un bajo acoplamiento y fomentando la reutilización (Visconti, y otros, 2011).

Una de las características de *Django* es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, se puede observar en el módulo que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona. Esto hace posible que el sistema sea flexible a cambios sustanciales con efecto mínimo.

- **Controlador:** Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. El mismo asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente el “sistema” global (Visconti, y otros, 2011).

Este patrón se pone de manifiesto en todo el módulo debido a que cada uno de los eventos generados por el usuario es redirigido a una clase controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión.

### **Patrones *Gang of Four* (GOF)**

El catálogo de patrones más famoso es el contenido en el libro “*Design Patterns: Elements of Reusable Object-Oriented Software*”, también conocido como: El libro GOF (*Gang-Of-Four Book*). Según este documento (Gamma, y otros, 1997), estos patrones se clasifican por su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos. El patrón GOF utilizado en el módulo es el siguiente:

---

<sup>9</sup> Del inglés *Uniform Resource Locator*, Localizador De Recursos Uniforme.

- **Decorator (Decorador):** Permite añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. En el MDMER se emplean los decoradores *user\_active\_required* y *user\_admin\_required*.

### 2.3.8. Modelo de datos

Este modelo está compuesto por las entidades (tablas) de la base de datos de *PostgreSQL* que garantiza la persistencia y manejo de los datos desde las clases controladoras del módulo, que fueron representadas en los epígrafes anteriores. El modelo de datos generado, como se observa en la Figura 9, está compuesto por dieciséis (16) relaciones y once (11) tablas, las cuales se explican a continuación, para un mejor entendimiento del modelo (de arriba abajo y de izquierda a derecha):

**Perfil:** almacena la información adicional referente a un usuario en el módulo, proporcionada por el sistema de identificación de la UCI. Se almacena avatar, credencial, carné de identidad, sexo, categoría, expediente, área, teléfono de la residencia, edificio y apartamento, así como municipio y provincia de procedencia. Contiene también la referencia al usuario al cual pertenece el perfil (*id\_usuario*).

**Usuario:** registra, de un usuario del sistema, el atributo que lo identifica (*id\_usuario*), su usuario uci, la contraseña, el correo, su(s) nombre(s) y apellidos, la fecha en que accedió por primera vez al sistema, así como la fecha de su última conexión.

**Comentario:** contiene los comentarios que son realizados en los foros, referentes a las respuestas de ejercicios publicadas en los mismos. Contiene además las referencias al usuario que realiza el comentario (*id\_usuario*) y a la respuesta publicada que se comenta (*id\_respuesta\_publica*).

**Estudiante:** esta entidad, es una especialización de la entidad usuario, por lo que hereda todos sus atributos. Del estudiante se desea conocer además su identificador (*id\_estudiante*) y la referencia al grupo docente al que pertenece (*id\_grupo\_docente*).

**Profesor:** también se especializa de la entidad usuario. Se conoce de este el atributo que lo identifica (*id\_profesor*).

**Ejercicio:** almacena la información referente a los ejercicios que son creados en el sistema, su identificador (*id\_ejercicio*), el nombre, la descripción del ejercicio, su tipo, fecha de creación, fecha de modificación, los modelos *JSON* e imagen del ejercicio, si ha sido eliminado y si está disponible o no para ser resuelto. Se

conoce además el usuario que lo crea (*id\_usuario*) y el foro al que se asocia el ejercicio (*id\_foro*), atributo que puede ser nulo, si el ejercicio no se asocia a ningún foro.

**Respuesta:** de las respuestas se conoce su identificador (*id\_respuesta*), la fecha de envío, tipo de respuesta (privada o pública), la fecha de creación y fecha de modificación, el estado de finalización (terminado o pendiente), y si ha sido revisada. Se almacena también, la referencia al ejercicio al cual pertenece la respuesta (*id\_ejercicio*) y al estudiante que la realizó (*id\_estudiante*).

**RespuestaRevision:** Almacena los datos referentes a la revisión de las respuestas, y se conoce la respuesta que ha sido revisada, el profesor que la revisó, la evaluación emitida y el criterio del mismo.

**Disenno:** esta entidad, hereda de la entidad respuesta, se conoce su identificador (*id\_respuesta\_disenno*) y se almacena además el diagrama diseñado en formato *JSON* y en formato de imagen.

**GrupoDocente:** contiene la información referente a los grupos docentes que son atendidos por los profesores y a los cuales pertenecen los estudiantes usuarios del sistema. Se conoce su identificador (*id\_grupo\_docente*), el nombre del grupo, el profesor que lo atiende (*id\_profesor*), la fecha de creación, si ha sido eliminado y si está disponible o no.

**Foro:** almacena la información de los foros que son creados en el sistema, su identificador (*id\_foro*), el tema con el que está relacionado, la descripción del foro, la fecha de creación, la forma de evaluación, el tipo de foro y el nivel de dificultad sobre el que estarán los ejercicios que se asocian a él. Se conoce además el profesor que crea el foro (*id\_profesor*).

## Capítulo 2: Análisis y diseño de la propuesta de solución

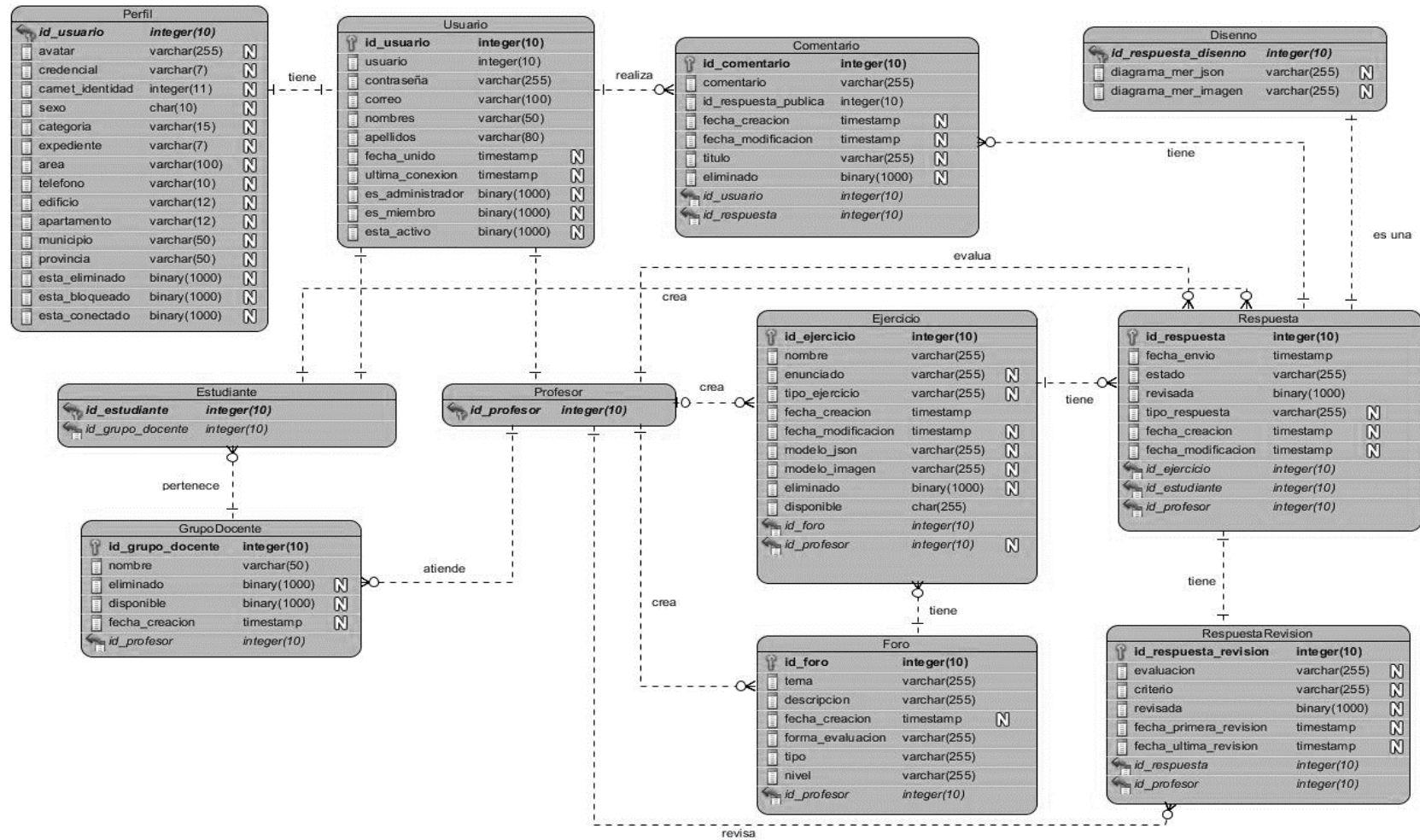


Figura 9. Modelo de datos del MDMER

### 2.3.9. Modelo de despliegue

Un modelo de despliegue consiste en una representación estructural de la arquitectura del sistema desde el punto de vista de la 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 (Sarmiento, 2016).

A continuación, se muestra el diagrama de despliegue propuesto para el MDMER. El mismo, muestra la disposición física de los nodos que componen el sistema y el reparto de los componentes en dichos nodos.

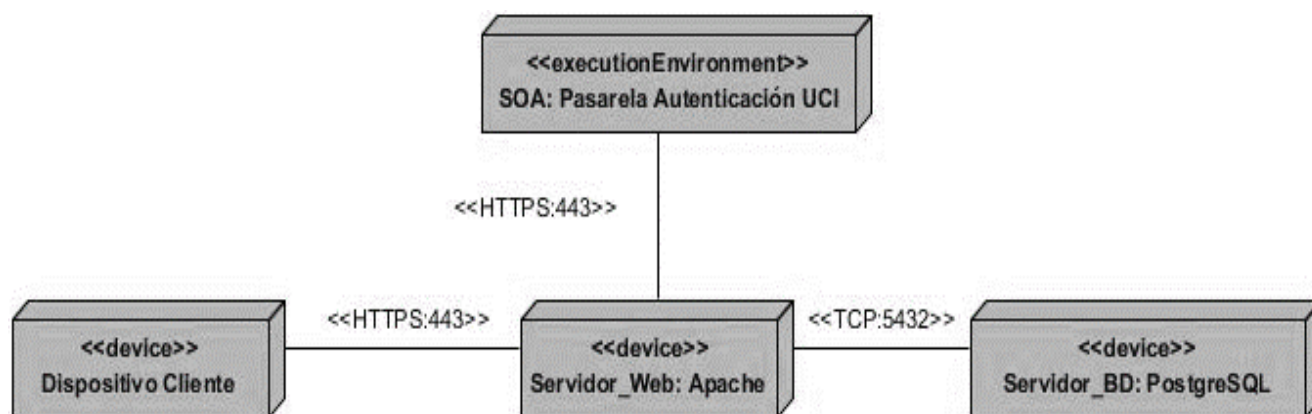


Figura 10. Modelo de despliegue del MDMER

Este diagrama se considera importante para lograr un despliegue exitoso de la aplicación. En él, se definen las estaciones de trabajo (Dispositivo Cliente: computadora personal, tableta, teléfono celular) que el usuario utilizará para conectarse, vía *HTTPS*<sup>10</sup>, con el servidor de aplicaciones web (*Apache*) en el que se encuentra alojado el módulo. Este servidor debe disponer de un ordenador con sistema operativo *GNU/Linux* y cuyas propiedades mínimas sean un procesador *Intel Dual Core* con velocidad de 2.10 GHz, 2 GB de memoria RAM y 500 MB de disco duro. Además, se requiere de la instalación del módulo *WSGI*, *pyhton 2.7.11*, *python-suds 0.4*, *python-Pillow 2.7.0*, *python-psycopg 2* y *python-django 1.9.5*.

Se observa, además, la conexión vía *HTTPS* del servidor web con el servidor *SOA*<sup>11</sup>, necesario para la obtención de la pasarela de autenticación UCI, que garantiza la identificación y acceso de los usuarios al

<sup>10</sup> Del inglés *HyperText Transfer Protocol Secure*, Protocolo Seguro de Transferencia de Hipertexto (Significado, 2014).

<sup>11</sup> Del inglés *Start of Authority*, Inicio de autoridad, para servidores DNS (CódigoMaestro, 2008).



sistema, y el uso del protocolo *TCP*<sup>12</sup> para conectarse con la BD del mismo, la cual estará en servidor de BD *PostgreSQL* con capacidad mínima de 500 MB de disco duro.

### 2.4. Conclusiones del capítulo

Después de realizado el análisis y diseño de la propuesta de solución y haber generado los diferentes artefactos que dispone la metodología *AUP*, se puede concluir lo siguiente:

1. El análisis de las características del módulo y la modelación del dominio, permiten identificar los principales requisitos funcionales y no funcionales del MDMER, los cuales fueron agrupados y categorizados por historias de usuarios.
2. La identificación de los patrones de diseño y el estilo arquitectónico del módulo, evidencia que la solución propuesta cuenta con un alto grado de resistencia ante posibles modificaciones.
3. El diseño de los diagramas de clases y de secuencia, facilita la visión en cuanto a composición física y lógica del módulo.
4. La generación de todos los artefactos requeridos por el modelo de desarrollo, documentan la solución propuesta, lo cual facilita su posterior mantenimiento (actualización o adición de funcionalidades).

---

<sup>12</sup> Del inglés *Transmission Control Protocol*, Protocolo de Control de Transmisión (Significado, 2014).

### Capítulo 3: Implementación y validación de la propuesta de solución

#### 3.1. Introducción

Antes de escribir una sola línea de código, es fundamental haber comprendido bien el problema que se pretende resolver y haber aplicado principios básicos de diseño que permitan construir un sistema de calidad. Una vez que se sabe qué funciones debe desempeñar el sistema (análisis) y se ha decidido cómo organizar sus distintos componentes (diseño), es el momento de pasar a la etapa de implementación. En esta fase, según Russo (2011), se toma como punto de partida el modelo de la fase anterior y se procede a programar los diseños especificados, los cuales son implementados en términos de componentes, ficheros de código fuente y ejecutables.

Por otra parte, el desarrollo de un software es algo complejo y son innumerables las posibilidades de cometer errores. Por esta razón todo proceso de implementación debe ir acompañado de alguna actividad que garantice la calidad. Las pruebas de validación constituyen una base para garantizar la aceptación favorable de una aplicación informática por parte del usuario. Con la realización de las mismas se pretende encontrar y documentar los errores que tiene un sistema, validar los requisitos y comprobar que estos fueron implementados correctamente. Este capítulo tiene como objetivo, documentar los resultados de las fases de implementación del módulo y de la estrategia de pruebas desarrollada.

#### 3.2. Diagrama de componentes

El diagrama de componentes muestra un conjunto de elementos de un modelo, tales como: componentes, subsistemas de implementación y sus relaciones, el cual se utilizan para modelar la vista estática de un sistema. En éste se muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema (Urquiza, 2013).

A continuación, la Figura 11, se muestra el diagrama de componentes del MDMER; cuya organización se encuentra acorde con el estilo arquitectónico MPV propuesto por el marco de trabajo *Django* y descrita en el capítulo anterior de este trabajo.

### Capítulo 3: Implementación y validación de la propuesta de solución

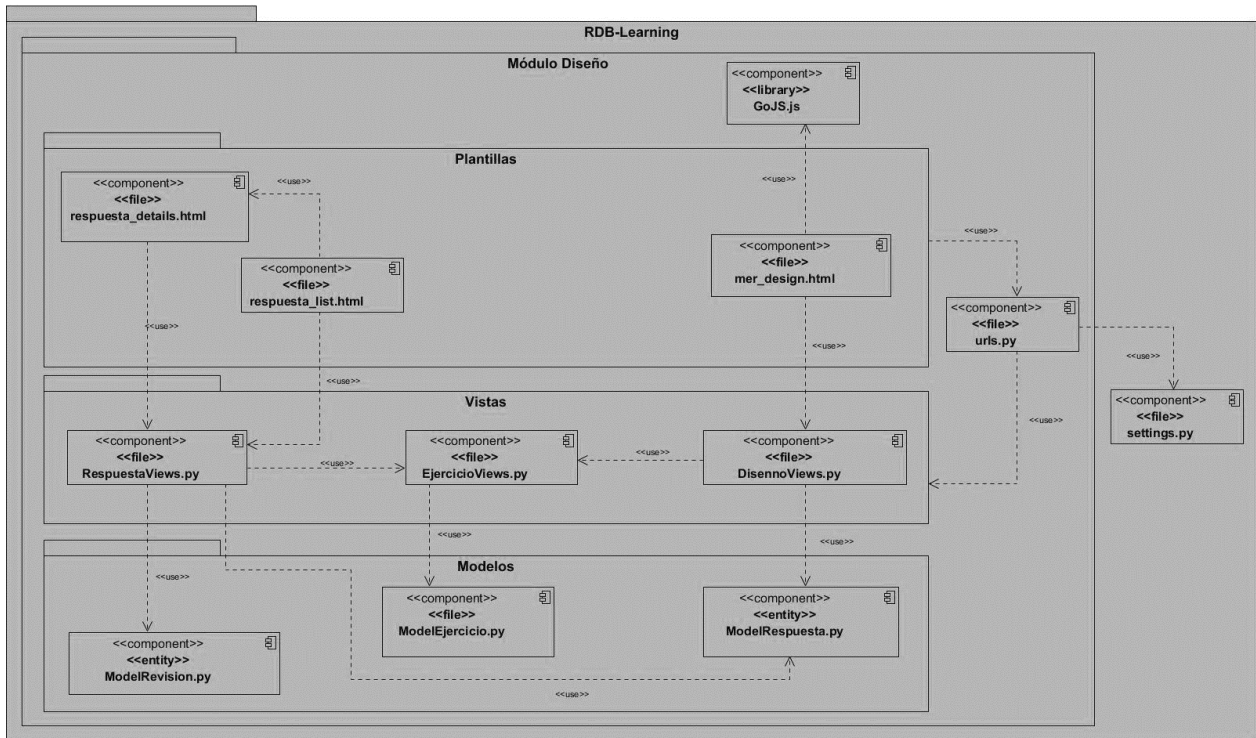


Figura 11. Diagrama de componentes del MDMER

A continuación, se describen los elementos que componen el diagrama de componentes mostrado.

Tabla 6. Descripción de los elementos del diagrama de componentes

Componentes		Descripción
Plantillas	mer_design.html	Este componente es el que muestra el área de trabajo para el diseño de MER. Utiliza la librería <i>GoJS</i> para el diseño de diagramas
	respuesta_list.html	Este componente muestra un listado con todas las respuestas enviadas al sistema permitiendo filtrarla para ser mostrada tanto por ejercicio, por estudiante y las privadas (profesores), así como las públicas (foro del problema), o todas de manera general (solo usuario con permisos)

### Capítulo 3: Implementación y validación de la propuesta de solución

	respuesta_details.html	Este componente muestra los detalles de una respuesta en que si esta es privada el profesor podrá evaluarla y enviar sus consideraciones al estudiante, en caso de ser publica se mostraría los comentarios hecho por todos los usuarios
Vistas	EjercicioViews.py	Este componente se encarga de gestionar los ejercicios en el sistema
	RespuestaViews.py	Este componente se encarga de guardar (área personal), enviar al profesor (privada), publicar en el foro (pública) al que pertenece el ejercicio, evaluar la respuesta, eliminarla, mostrar los detalles de la misma, así como devolver un listado de estas
	DisennoViews.py	Este componente se encarga de listar los ejercicios de diseño en el sistema, así como crear y editar las respuestas creadas
Modelos	ModelEjercicio.py	Clase entidad que almacena los ejercicio disponibles en el sistema
	ModelRespuesta.py	Clase entidad que almacena las respuestas a los ejercicios que son enviadas al sistema
	ModelRevision.py	Clase entidad que almacena los datos de las respuestas privadas: la evaluación emitida por el profesor y su criterio

#### 3.3. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera

### Capítulo 3: Implementación y validación de la propuesta de solución

escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento (Microsoft, 2017).

A continuación, se definen los estándares de codificación a utilizar en la implementación del módulo, basados en los que fueron utilizados en el desarrollo de la plataforma *RDB-Learning*.

Tabla 7. Estándares de codificación para la implementación del MDMER

Tipo de estándar	Descripción
Organización del código	<ul style="list-style-type: none"><li>• El código en una página se organizará por bloques</li><li>• El formato del código se ajustará por la opción Reformat Code del IDE de desarrollo PyCharm</li><li>• La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios</li></ul>
Máxima longitud de las líneas	<ul style="list-style-type: none"><li>• Todas las líneas deben estar limitadas a un máximo de setenta y nueve (79) caracteres</li><li>• Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas</li><li>• En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas</li></ul>
Líneas en blanco	<ul style="list-style-type: none"><li>• Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco</li><li>• Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco</li><li>• Se puede utilizar líneas en blanco escasamente para separar secciones lógicas</li></ul>

### Capítulo 3: Implementación y validación de la propuesta de solución

Tipo de estándar	Descripción
Importaciones	<ul style="list-style-type: none"><li>• Las importaciones deben estar en líneas separadas</li><li>• Siempre deben colocarse al comienzo del archivo</li><li>• Deben quedar agrupadas de la siguiente forma:<ol style="list-style-type: none"><li>1. Importaciones de la librería estándar</li><li>2. Importaciones terceras relacionadas</li><li>3. Importaciones locales de la aplicación / librerías</li></ol></li><li>• Cada grupo de importaciones debe estar separado por una línea en blanco</li></ul>
Codificaciones	<ul style="list-style-type: none"><li>• Utilizar la codificación UTF-8</li><li>• Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”</li></ul>
Comentarios	<ul style="list-style-type: none"><li>• Los comentarios deben ser oraciones completas</li><li>• 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</li><li>• Si un comentario es corto el punto final puede omitirse</li><li>• Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código JavaScript mientras que en Python por el carácter “#” y deben ubicarse en la misma línea que se desea comentar</li><li>• Los comentarios de varias línea para organización del código aparecerán dentro de los caracteres “/** ... */” en caso de código JavaScript, mientras que en Python se hará con los caracteres “...”</li></ul>

### Capítulo 3: Implementación y validación de la propuesta de solución

Tipo de estándar	Descripción
Espacios en blanco en expresiones y sentencias	<ul style="list-style-type: none"><li>• Evitar utilizar espacios en blanco en las siguientes situaciones:<ul style="list-style-type: none"><li>- Inmediatamente dentro de paréntesis, corchetes y llaves</li><li>- Inmediatamente antes de una coma, un punto y coma o dos puntos</li><li>- Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función</li></ul></li><li>• Deben rodearse con exactamente un espacio los siguientes operadores binarios:<ul style="list-style-type: none"><li>- Asignación (=)</li><li>- Asignación de aumentación (+=, -=, etc.)</li><li>- Comparación (==, &lt;, &gt;, &gt;=, &lt;=, !=, &lt;&gt;, in, not in, is, is not)</li><li>- Expresiones lógicas (and, or, not)</li></ul></li><li>• No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto</li></ul>
Convenciones de nombramiento	<ul style="list-style-type: none"><li>• Nunca se deben utilizar como simple caracteres para nombres de variables los caracteres en minúscula "l", o mayúscula "O", o la mayúscula "L" ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0)</li><li>• Los nombres de clases deben utilizar la convención "CapWords" (palabras que comienzan con mayúsculas)</li><li>• Los nombres de las excepciones deben estar escrito también en la convención "CapWords" utilizando el sufijo "error"</li></ul>

### 3.4. Validación de la propuesta de solución

Durante la etapa de implementación, pueden cometerse algunos errores y pueden pasarse por alto algunos elementos que son importantes para el correcto funcionamiento del sistema. Por tal motivo, es esencial llevar a cabo la fase de validación, en la cual, a través de varios tipos y métodos de pruebas de software (estrategia de pruebas), se pretende comprobar el cumplimiento de las especificaciones del diseño y de la codificación, identificar los posibles errores cometidos y validar la solución propuesta en los capítulos anteriores (Labrada, y otros, 2013). En este epígrafe se muestran los resultados de la estrategia de prueba diseñada para el MDMER (ver Tabla 8), en función de garantizar y validar su calidad.

Tabla 8. Estrategia de prueba para el MDBDR

Tipo de prueba	Método (técnica) de prueba	Validación
Funcional	Casos de prueba ( Caja Negra)	Valida las funcionalidades diseñadas para el sistema
Usabilidad	Lista de Chequeo	Valida, a partir de sus características, la capacidad del sistema de cumplir con el propósito para el que fue diseñado
Seguridad	<i>Software Acunetix</i>	Valida la confidencialidad, integridad y disponibilidad de los datos en el sistema
Aceptación	Carta de aceptación del cliente	Valida la funcionalidad del sistema a partir de la aceptación determinada por el usuario (cliente) del mismo
Carga y estrés	<i>Software Apache JMeter</i>	Valida el comportamiento del sistema con distintos niveles de usuarios concurrentes y el consumo excesivo de sus recursos



### 3.4.1. Pruebas funcionales

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, pues los probadores o analistas de pruebas, no enfocan su atención a cómo se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas (Oré, 2009).

Con el objetivo de realizar este tipo de pruebas al MDMER, se diseñó un conjunto de casos de pruebas, seis (6) en total, referentes a varias de las historias de usuarios obtenidas en la fase de diseño del capítulo anterior, pertenecientes a requisitos funcionales de prioridad alta, también especificados en dicho capítulo. A continuación, se muestra uno (1) de los casos de prueba mencionados. En las celdas de la tabla del caso de prueba (ver tabla 10) se pueden encontrar los valores **V**, para datos válidos, **I**, para datos inválidos, y **N/A**, para datos a los que no es necesario proporcionarles un valor.

Descripción de las variables:

Tabla 9. Descripción de las variables del caso de prueba 1

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Permite todos los caracteres
2	Enunciado	Campo de texto	No	Permite todos los caracteres
3	Tipo	Campo de selección múltiple	No	Puede tomar los valores Diseño e Implementación
4	Publicado	Casilla de verificación	No	Este campo puede estar activado o desactivado
5	Foro	Campo de selección múltiple	Si	Puede estar vacío o tomar cualquiera de los valores disponibles

### Capítulo 3: Implementación y validación de la propuesta de solución

Tabla 10. Caso de prueba del RF1\_Crear ejercicios de diseño

Caso de prueba 1: SC RF1_ Crear ejercicios de diseño								
Condiciones de ejecución: El usuario debe tener los permisos necesarios								
Escenario	Descripción	1	2	3	4	5	Respuesta del sistema	Flujo Central
EC 1.1 Crear ejercicio asociado a un foro de forma correcta	Interfaz con el formulario para llenar los datos del ejercicio, si todos son correctos, se agrega el ejercicio al sistema	V	V	V	V	V	Agrega el ejercicio y muestra un mensaje de notificación	1. Seleccionar, en el módulo profesor, la opción "Ejercicios"
		Ejercicio_1	Esto es una prueba	Diseño	Activado	Foro#1		
EC 1.1 Crear ejercicio privado de forma correcta	Interfaz con el formulario para llenar los datos del ejercicio, si todos son correctos, se agrega el ejercicio al sistema	V	V	V	V	V	Agrega el ejercicio y muestra un mensaje de notificación	3. Llenar los campos correspondientes en el formulario y seleccionar la opción "Guardar"
		Ejercicio_2	Esto es otra prueba	Diseño	Activado			

### Capítulo 3: Implementación y validación de la propuesta de solución

Escenario	Descripción	1	2	3	4	5	Respuesta del sistema
EC 1.2 Crear ejercicio con campos vacíos	Interfaz con el formulario para llenar los datos del ejercicio, si existe algún campo vacío, se muestra un mensaje pidiendo llenar el campo	I	I	I	V	V	Comprueba si los campos están vacíos, si lo están, muestra un mensaje que solicita llenarlos
					Desactivado		
EC 1.3 Crear ejercicio con el mismo nombre a otro existente	Interfaz con el formulario para llenar los datos del ejercicio, si existe algún ejercicio con el mismo nombre, se muestra un mensaje de error	V	V	V	V	V	Comprueba si existe algún ejercicio con el mismo nombre, si existe, muestra un mensaje de notificación
		Ejercicio_1	Esto es otra prueba	Diseño	Activado	Foro#1	

### Capítulo 3: Implementación y validación de la propuesta de solución

Escenario	Descripción	1	2	3	4	5	Respuesta del sistema
EC 1.4	Crear ejercicio con tipo distinto al del foro al que se pretende asociar	V	V	V	V	V	El sistema comprueba los campos y no permite crear el ejercicio con un tipo distinto al del foro a asociar, mostrando un mensaje de error
		Ejercicio_3	Esto es otra prueba 3	Implementación	Desactivado	Foro#1	

Como resultado final de las pruebas funcionales, se obtuvo, en una primera iteración, un total de diez (10) no conformidades, divididas en tres (5) de ortografía, una (1) de redacción, dos (2) de funcionalidad y cuatro (2) de validación. De estas, se resolvieron seis (6), y cuatro (4) quedaron pendientes. En una segunda iteración, no se identifican nuevas no conformidades y de las cuatro (4) pendientes, solo una (1) se mantuvo para la próxima iteración, donde fue resuelta y no se detectan nuevas no conformidades. En una cuarta iteración no se identifican nuevas inconformidades, obteniendo, de esta manera, resultados satisfactorios. La siguiente gráfica, muestra los resultados antes descritos:

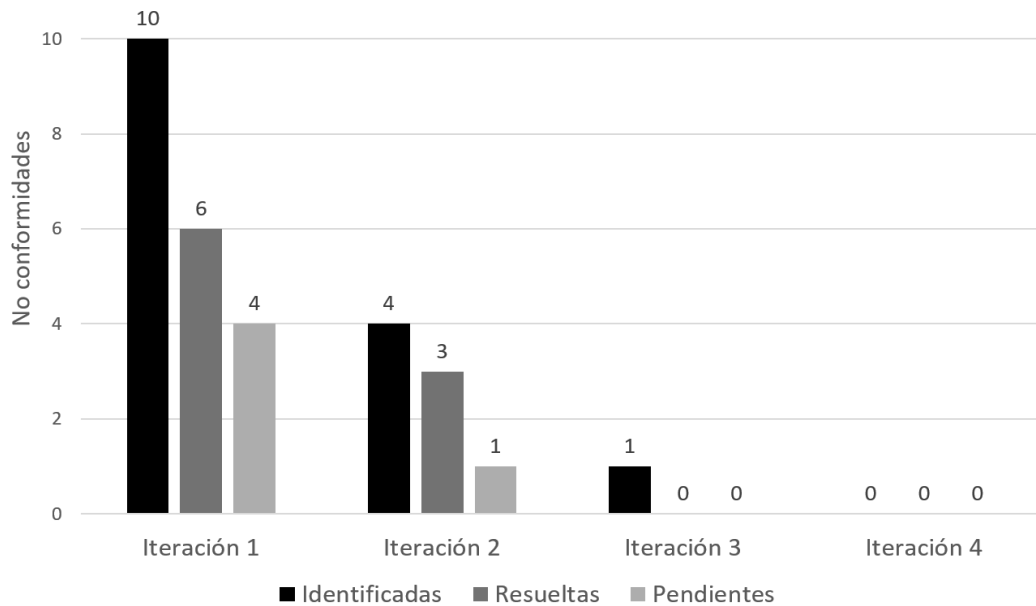


Figura 12. Resultados de las pruebas funcionales

Las no conformidades de funcionalidad, estuvieron relacionadas con crear un ejercicio en el sistema y calificar una respuesta. En el primer caso, no se permitía crear el ejercicio sin asociarlo a un foro, sin embargo, es un evento necesario en el sistema. Esta no conformidad fue resuelta al permitir que el atributo *id\_foro* de la entidad *Ejercicio*, el cual estaba especificado como campo obligatorio, pudiese tomar valor nulo. En el segundo caso, al calificar una respuesta solo se modificaba la evaluación de la primera que se encontraba en la lista de las respuestas para evaluar. El problema estaba dado porque el identificador de la ventana emergente de evaluación se especificó como atributo único, por lo que se le adicionó el identificador de la respuesta que se pretendía evaluar en cada caso para dar solución a este problema.

Las no conformidades de validación fueron encontradas en el formulario para crear un ejercicio, donde se permitía introducir caracteres extraños en el nombre de un ejercicio y crear el ejercicio con un tipo distinto al del foro al que se pretendía asociar. La solución para los caracteres extraños fue validar el nombre del ejercicio mediante una expresión regular que permite solo el uso de caracteres alfa-numéricos, las tildes, las diéresis, espacios, puntos y el guion bajo. Mientras que, para crear un ejercicio asociado correctamente a un foro, se validó que el tipo de ejercicio fuese el mismo que el del foro, lo cual no permite una asociación de tipos diferentes. El sistema muestra un mensaje de error si se intenta realizar esta acción de manera incorrecta.

### 3.4.2. Pruebas de seguridad

Las pruebas de seguridad se realizan para comprobar que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas (Pressman, 2010). Además, se encargan de certificar que los datos y las funciones del sistema solo son accesibles por los actores debidamente autorizados (Toll, y otros, 2007).

Para la realización de este tipo de prueba, se empleó la herramienta *Acunetix Web Vulnerability Scanner*, caracterizada en el epígrafe 1.5. En una primera iteración, se obtuvo un total de quince (15) no conformidades, divididas en seis (6) de nivel medio, seis (6) de nivel bajo y tres (3) de carácter informativo. De las de nivel medio, destacó el uso del protocolo no seguro para el envío de datos, así como los mensajes de error que se muestran en el modo DEBUG de Django para el desarrollo. Las de nivel bajo estuvieron relacionadas con problemas para la protección contra ataques de fuerza bruta a la página de autenticación, así como directorios que pueden ser accesibles directamente sin pasar la autenticación y la protección de las cookies y las sesiones en el navegador. De carácter informativo fueron detectadas una dirección de correo y una posible cuenta de usuario en un fichero. Todas estas deficiencias fueron corregidas en la primera iteración, y para una segunda, no se identificó ninguna nueva, por lo cual se obtuvo finalmente una herramienta que cumple con los requisitos de seguridad definidos para la misma. Los resultados antes descritos, se muestran a continuación en la siguiente gráfica:

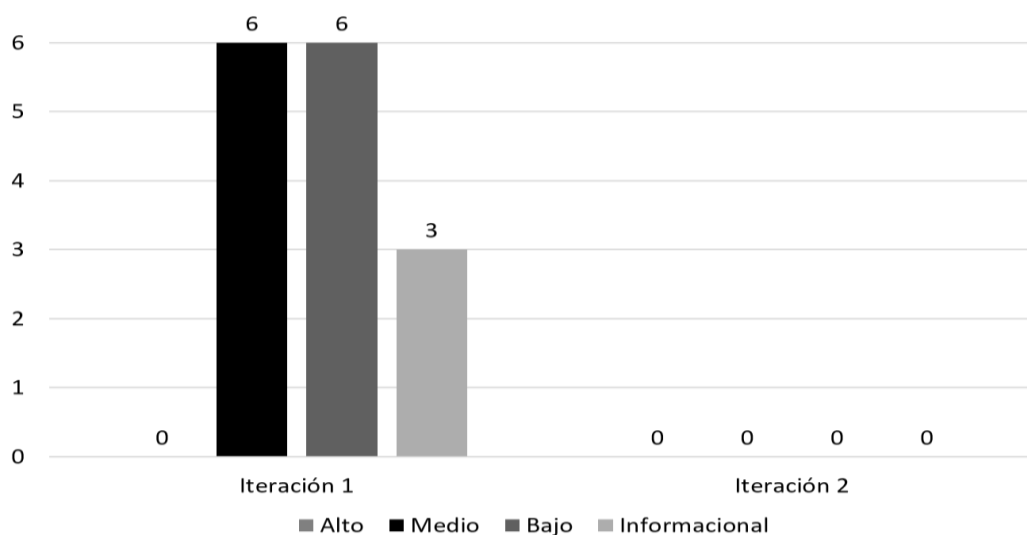


Figura 13. Resultados de las pruebas de seguridad

### 3.4.3. Pruebas de usabilidad

Según diversos estándares de la Ingeniería de Software, se puede definir la usabilidad como el grado en el que un producto puede ser utilizado por usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción, en un determinado contexto de uso. Como se puede apreciar, la usabilidad de un sistema está ligada a usuarios, necesidades y condiciones específicas (Carcasés, 2016). De manera general, el término usabilidad es empleado para referirse a la capacidad que posee un producto de ser utilizado por los usuarios de forma fácil, eficiente y con satisfacción, en un determinado contexto de uso.

Se puede decir que el proceso de prueba de usabilidad se enfoca en satisfacer las necesidades de los usuarios finales basados en métricas definidas durante la planeación. Para la realización de las pruebas de usabilidad, se hace uso de la “Lista de Chequeo de Usabilidad para sitios web”, desarrollada por los especialistas del grupo de Seguridad del Departamento de Evaluación de Productos de Software (DEPSW), perteneciente al Centro Nacional de Calidad de Software (CALISOFT). A continuación, se muestran los resultados de dichas pruebas.

La tabla 11 refleja las categorías en las que se dividen los indicadores, cuántos indicadores existen por cada categoría, de ellos, cuántos se aplican a la evaluación del MDMER (Proceden), y de los evaluados, cuántos están implementados correcta e incorrectamente en el módulo.

Tabla 11. Resultado de las pruebas de usabilidad

Categoría de los indicadores	Indicadores	Proceden	Correctos	Incorrectos
Visibilidad del sistema	17	14	13	1
Lenguaje común entre sistema y usuario	11	6	6	0
Libertad y control por parte del usuario	29	14	11	3

### Capítulo 3: Implementación y validación de la propuesta de solución

Categoría de los indicadores	Indicadores	Proceden	Correctos	Incorrectos
Consistencia y estándares	33	22	19	3
Estética y diseño minimalista	18	8	8	0
Prevención de errores	8	5	4	1
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	11	7	1	6
Ayuda y documentación	11	3	1	2
Flexibilidad y eficiencia de uso	6	4	3	1
<b>Total</b>	<b>144</b>	<b>83</b>	<b>66</b>	<b>17</b>

Como se observa en la tabla, de los 144 parámetros originales de la lista de chequeo, solo proceden 83. En una primera iteración se evaluaron como correctos 66 parámetros, identificando 17 no conformidades, para un 80 % de usabilidad. Los principales problemas estuvieron relacionados con la ayuda a los usuarios mediante documentación y para enfrentar errores en el módulo. Las no conformidades se revisaron y de las 17 encontradas, fueron solucionadas 15, obteniendo un sistema con un 97,6 % de usabilidad.

#### 3.4.4. Pruebas de aceptación

Según Huaraca (2013), el uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada prueba de aceptación. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.



### Capítulo 3: Implementación y validación de la propuesta de solución

Por su parte, la Junta Internacional de Cualificaciones de Pruebas de Software (*ISTQB* por sus siglas en inglés) define la “Aceptación” como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (PMOinformatica, 2016).

Para realizar la prueba de aceptación, se entregó la aplicación al cliente, el cual emitió su criterio a través de una carta de aceptación, a partir de sus consideraciones respecto a las ventajas que ofrece el módulo y las necesidades que resuelve. Para respaldar dicho criterio, se solicitó la colaboración de un grupo de expertos de la asignatura SBD I. El instrumento utilizado es una encuesta (ver Anexo 2) a través de la cual se pueda validar el nivel de aceptación del MDMER en los usuarios finales. La encuesta fue realizada a 4 profesores de SBD I de la facultad 1, los cuales cuentan con un promedio de 4 años de experiencia en la impartición de la asignatura.

La tabla 12 resume el resultado de los juicios emitidos por los encuestados, de acuerdo a los siguientes parámetros:

1. Tratamiento de los aspectos teóricos de la asignatura en el MDMER.
2. Nivel de apoyo que brinda al profesor el uso del MDMER para la orientación y evaluación de ejercicios.
3. Contribución del MDMER al tema de diseño de MER en la asignatura SBD I.
4. Presentación de una interfaz agradable e intuitiva para el usuario.
5. Usabilidad del módulo.

Para el procesamiento y análisis de la información obtenida se analizaron las respuestas de cada uno de los parámetros que aparecen en la encuesta. De esta manera se presentan los resultados teniendo en cuenta que los niveles empleados para la valoración fueron: **MA**: Muy adecuado, **A**: Adecuado, **PA**: Poco adecuado y **NA**: No adecuado. En la tabla, N se refiere a la cantidad de encuestados que emitieron una valoración determinada y % al por ciento que representa con respecto al total de encuestados.

### Capítulo 3: Implementación y validación de la propuesta de solución

Tabla 12. Resultado de las encuestas realizadas para pruebas de aceptación

Parámetros evaluados	Niveles de valoración									
	MA		A		PA		NA		Total	
	N	%	N	%	N	%	N	%	N	%
1	4	100	0	0	0	0	0	0	4	100
2	2	50	2	50	0	0	0	0	4	100
3	2	50	2	50	0	0	0	0	4	100
4	3	75	1	25	0	0	0	0	4	100
5	4	100	0	0	0	0	0	0	4	100

Los resultados antes mostrados, son representados de forma gráfica a continuación:

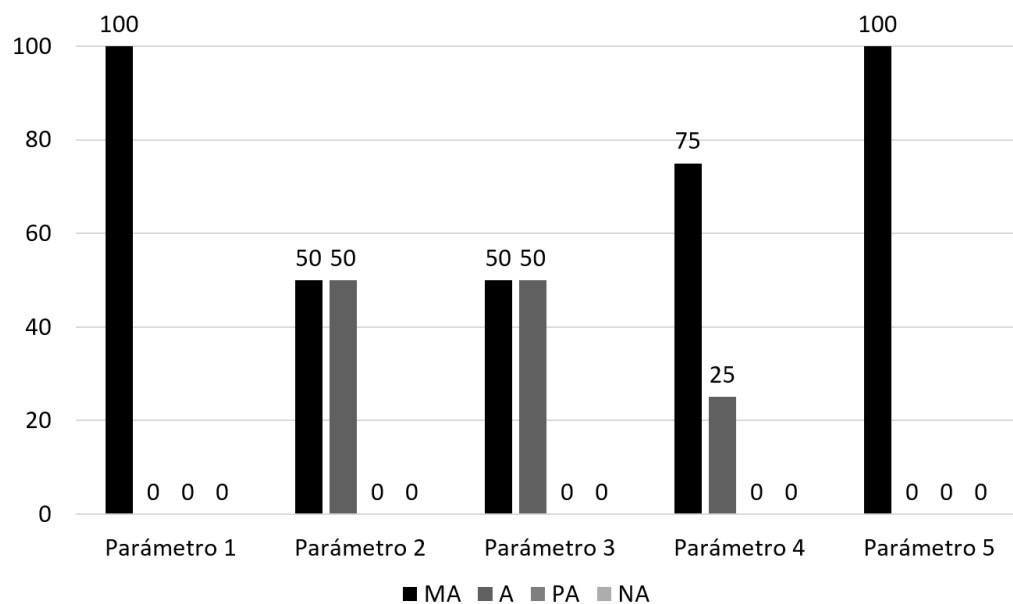


Figura 14. Resultados de la encuesta de aceptación del MDMER

Como es posible observar, al analizar los resultados de la consulta, se pudo constatar que el 100% de los encuestados consideran de muy adecuado el tratamiento de los aspectos teóricos de la asignatura SBD I en el MDMER. El nivel de apoyo que brinda al profesor el uso del MDMER para la orientación y evaluación de ejercicios, así como su contribución al tema de diseño de MER en la asignatura, fueron valorados de muy adecuado por el 50% de los encuestados. El otro 50% considera que estos aspectos son afrontados de manera adecuada. Respecto a la presentación de una interfaz agradable e intuitiva para el usuario, solo el 75% de los encuestados considera que es muy adecuada, sin embargo, el 100% de ellos considera muy adecuada su usabilidad.

Teniendo en cuenta los criterios, así como otras consideraciones expresadas por los profesores de la asignatura en la encuesta, se puede comprobar que la solución implementada tiene un nivel satisfactorio de aceptación para ellos. Los aspectos evaluados, los cuales están en concordancia con el objetivo general de la investigación, fueron valorados todos entre los niveles de adecuado y muy adecuado. Esto demuestra el correcto cumplimiento de este objetivo, desde el punto de vista de los expertos. Además, se obtuvo un conjunto de recomendaciones y valoraciones que aportan mejoras a la propuesta de solución, las cuales se tienen en cuenta para el resultado final de la investigación, así como futuras profundizaciones sobre la misma.

### 3.4.5. Pruebas de carga y estrés

La prueba de carga y estrés se refiere, generalmente, a la práctica de comprobar el comportamiento de una aplicación mediante cargas o entradas pesadas. Las mismas se realizan con el fin de verificar si el sistema satisface los requisitos de rendimiento para situaciones críticas como pueden ser: la cantidad límite de usuarios accediendo de forma concurrente a los servicios brindados, documentos extremadamente grandes, cantidad de transacciones que se pueden procesar de forma concurrente cada minuto, tiempo de respuesta, entre otros (ITI, 2017).

Para la realización de esta prueba se utilizó la herramienta *Apache JMeter*, descrita en el epígrafe 1.5. Las pruebas se realizaron desde un ordenador con 2GB de RAM, microprocesador Intel Core i3 con 2.00 GHz y sistema operativo *GNU/Linux*. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

**Muestra:** Cantidad de peticiones realizadas para cada *URL*.

### Capítulo 3: Implementación y validación de la propuesta de solución

**Media:** Tiempo promedio en milisegundos en el que se obtienen los resultados.

**Mediana:** Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

**Min:** Tiempo mínimo que demora un hilo en acceder a una página.

**Max:** Tiempo máximo que demora un hilo en acceder a una página.

**Línea 90 %:** Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

**% Error:** Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

**Rendimiento (Rend):** El rendimiento se mide en cantidad de solicitudes por segundo.

**Kb/s:** Velocidad de carga de las páginas.

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas al módulo por un total de 100, 500 y 1000 usuarios simultáneamente en cada caso, los cuales realizan hasta 5 peticiones por segundo. Se obtuvieron los siguientes resultados:

Tabla 13. Resultado de las pruebas de carga y estrés

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90%	%Error	Rend	Kb/s
100	500	1633	1120	96	4501	2802	0.00%	45.6	189.9
500	2500	1262	1084	121	4992	2091	0.50%	226.5	537.1
1000	5000	1212	1002	73	4755	1693	2.93%	419,5	950.9

Las pruebas realizadas muestran que el módulo es capaz de responder a 500 peticiones de 100 usuarios conectados simultáneamente en un tiempo promedio de 1633 milisegundos (1.6 segundos aproximadamente) con 0 % de error, esto evidencia que el módulo puede procesar la carga esperada.

Por otra parte, se realizaron 2500 peticiones iniciadas por 500 usuarios y en este caso el módulo respondió en 1262 milisegundos (1.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el módulo puede procesar la carga esperada, aunque no fue capaz de responder correctamente el 0.50% de las peticiones realizadas.

Por último, y con el objetivo de analizar el comportamiento del módulo en condiciones extremas, se realizó una prueba de estrés para un conjunto de 1000 usuarios conectados simultáneamente. En este caso, el módulo responde a las 5000 peticiones en un tiempo promedio de 1212 milisegundos (1.2 segundos aproximadamente), pero con un porcentaje de error de 2.93. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado.

### 3.5. Interfaces principales del MDMER

Una vez desarrollado el MDMER, es posible visualizar las pantallas principales del mismo, donde se observa el resultado obtenido durante la implementación de las historias de usuarios descritas en el sub-epígrafe 2.3.3.

The screenshot displays the RDB-LEARNING platform interface. At the top, the logo 'XAUCE RDB-LEARNING' is visible, along with the tagline 'Plataforma para el aprendizaje de Bases de Datos Relacionales'. The navigation menu includes 'Principal', 'Herramientas', 'Foro', and 'Ayuda'. The user profile 'aaballi' is shown in the top right corner. The main content area is divided into three sections: 'EJERCICIO Detalles', 'ENUNCIADO DEL EJERCICIO', and 'RESPUESTAS'. The 'EJERCICIO Detalles' section shows the exercise name 'Ejercicio 1', creator 'Carlos Jordán', type 'DISEÑO', and duration '47 minutos'. The 'ENUNCIADO DEL EJERCICIO' section contains a detailed text description of a construction management problem. The 'RESPUESTAS' section is currently empty.

EJERCICIO Detalles	
Nombre:	Ejercicio 1
Creado por:	Carlos Jordán
Tipo de Ejercicio:	DISEÑO
Fecha:	47 minutos
Foro asociado:	MER

**ENUNCIADO DEL EJERCICIO**

La compañía cubana ETECSA desea gestionar, a través de su Departamento de Economía Legal, la información referente a las nuevas dependencias que la empresa debe construir a lo largo del país o en otros países. De cada nueva dependencia se conoce su código, la dirección, el área (en metros cuadrados) que la misma ocupa, y el número de empleados que la misma tendrá. Se conocen los datos de los empleados que estarán en cada nueva dependencia: nombre, identificador, especialidad a la que pertenece y escala salarial que posee, conociendo que cada uno labora en un solo lugar. De los empleados que son jefes además interesa conocer su cargo y el salario adicional. Se tienen registrados además los datos de las compañías constructoras que afrontarán la realización de una o de varias dependencias de ETECSA: tipo, código que la identifica y nombre. En la obra constructiva de cada dependencia interviene una única constructora. De cada país donde se localizarán dichas dependencias se conoce su identificador, el código del área geográfica a la que pertenece (1 -America, 2-Europa, 3-Asia, etc.) y el nombre del mismo. El monto total de construcción de cada dependencia es convenido con cada compañía constructora.

**RESPUESTAS**

Vacio

Figura 15. Interfaz de la página para mostrar un ejercicio

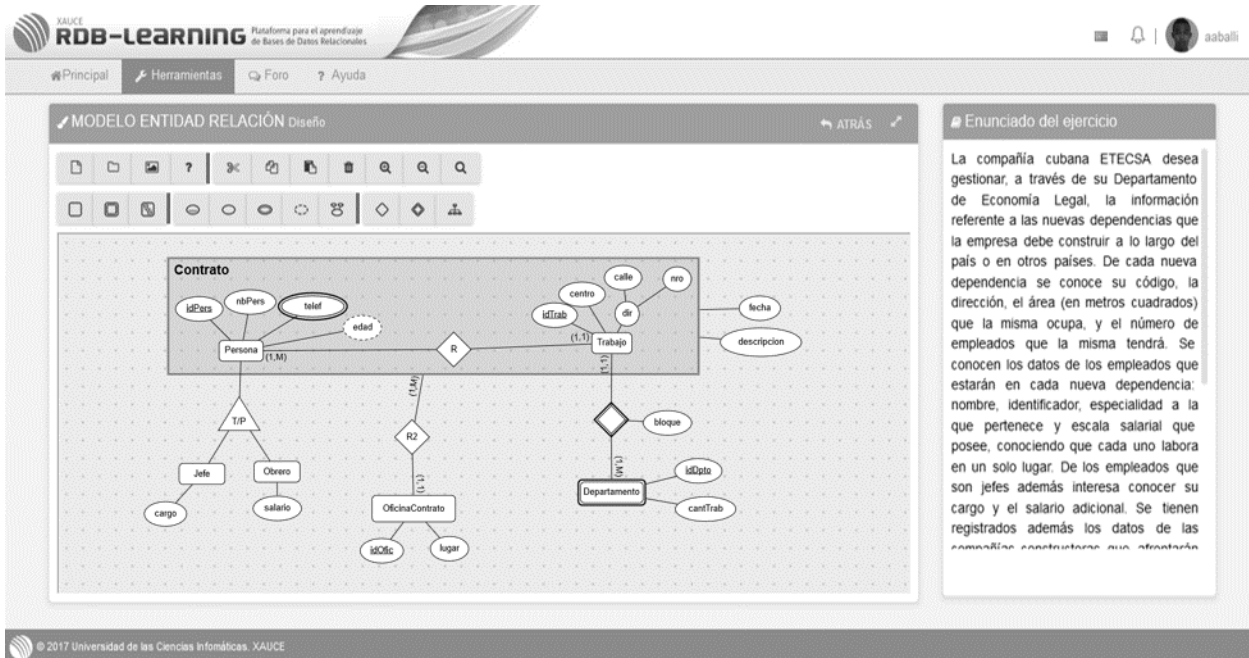


Figura 16. Interfaz de la página para crear una respuesta

### 3.6. Conclusiones del capítulo

En este capítulo se han abordado los elementos de la implementación del MDMER, así como las pruebas realizadas al mismo y los resultados obtenidos; lo cual permite arribar a las siguientes conclusiones:

1. La elaboración del diagrama de componentes, permite una mejor comprensión de la estructura de los componentes del módulo implementado.
2. El correcto uso de los estándares de codificación, permite que el código del módulo desarrollado fuera legible para lograr una fácil y mejor comprensión del mismo, la cual es de utilidad para el mantenimiento del sistema.
3. La implementación del módulo permite la obtención de una aplicación funcional y completamente operativa.
4. El proceso de validación de la propuesta de solución, a través de la estrategia de pruebas especificada, arroja como resultado que el módulo implementado responde a los requerimientos definidos por el cliente.

### Conclusiones Generales

De manera general, la presente investigación concluyó con el desarrollo del Módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*, el cual sirve de apoyo al proceso de enseñanza – aprendizaje de la asignatura SBD I y contribuye al diseño de modelos entidad relación en dicha asignatura. Esto se debe a que el módulo fue desarrollado como una tecnología del aprendizaje y el conocimiento que facilita al estudiante sus actividades de auto-preparación, y al profesor, llevar el control de la realización de estas actividades.

Otros aspectos significativos que se pueden destacar son:

1. El análisis y la fundamentación teórica de los principales conceptos asociados a la investigación, permitió lograr una mayor comprensión del alcance de la investigación y esclarecer su objeto de estudio.
2. La sistematización del marco teórico de la investigación científica, del estado actual de las herramientas informáticas para el diseño de modelos entidad relación, y de la documentación de la plataforma *RDB-Learning*, posibilitó la definición del ambiente de desarrollo para la implementación de la solución propuesta.
3. La integración de diversas áreas del conocimiento como son la ingeniería y gestión de software, base de datos, programación, entre otras, permitió el análisis, diseño e implementación del Módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*.
4. La solución fue validada a partir de la definición correcta de una estrategia de pruebas, que permitió comprobar el correcto funcionamiento del Módulo para el diseño de modelos entidad relación en la plataforma *RDB-Learning*, a partir de los requerimientos definidos por el cliente.

### Recomendaciones

Para el desarrollo de futuras investigaciones relacionadas con la presente, se propone:

1. Adicionar un módulo de corrección automática a la plataforma, que permita identificar errores en la representación del MER mientras se está diseñando.
2. Adicionar la funcionalidad de transformar el MER al Esquema Lógico o MR.
3. Adicionar la funcionalidad de generar el código *SQL* a partir del MER diseñado.



## Referencias Bibliográficas

- Abraira, José Angel Alvarez. 2016.** *Sistema interactivo - experimental para el proceso de enseñanza – aprendizaje de la Matemática Discreta en la carrera Ingeniería en Ciencias Informáticas.* La Habana : s.n., 2016.
- Alegsa, L. 2015.** Definición de Aplicación Web. Diccionario de informática y tecnología. 2015.
- Araujo, Pedro Bernabé y Rodríguez, Sebastián Alberto. 2013.** *Janeiro Studio.* 2013.
- Báez, Lorena Chavarría, Orozco, Rosaura Palma y Ledesma, Elena Ruiz. 2016.** *Sistema de software para apoyar la impartición de bases de datos.* México D.F : s.n., 2016.
- Carcasés, Arianne Ferrer. 2016.** Guía para los procesos de Usabilidad que se desarrollan en el Laboratorio de Pruebas de Software (LPS) de la Dirección de Calidad-UCI. 2016.
- Carignano, María Celeste. 2016.** *Representación y razonamiento sobre las decisiones de diseño de arquitectura de software.* Entre Ríos, Argentina : s.n., 2016.
- CódigoMaestro. 2008.** Código Maestro. [En línea] Agosto de 2008. [Citado el: 10 de Abril de 2017.] <http://www.codigomaestro.com/dns/registros-soa-en-servidores-dns/>.
- Costal, Dolors. 2014.** Introducción al diseño de Bases de Datos. 2014. P06/M2109/02150.
- Django. 2014.** Django Software Foundation. *The Django Book.* [En línea] 2014. <http://www.djangobook.com>.
- e-ABC. 2017.** e-ABC. *Definición de e-Learning.* [En línea] 2017. <http://www.e-abclearning.com/definicion-e-learning>.
- Fonseca, Heberto. 2009.** Retroalimentación durante el proceso de enseñanza–aprendizaje: un arma de doble filo. 2009.
- FORTES. 2015.** ZERA. *Plataforma Educativa.* [En línea] 2015. <https://eva.uci.cu>.
- Gamma, Erich, y otros. 1997.** *Design Patterns: Elements of Reusable Object-Oriented Software.* 1997.
- García, Carlos, Rodríguez, Abel y González, Luisa. 2008.** *Consideraciones acerca de la abstracción de agregación en la herramienta ERECASE.* Santa Clara : Universidad Central de Las Villas, 2008. ISSN 1657-7663.

- GoJS. 2017.** GoJS. *Interactive JavaScript Diagrams in HTML*. [En línea] 2017. [Citado el: 30 de Abril de 2017.] <https://gojs.net>.
- Gómez, Eva, Barco, Patricio Martínez y Pozo, Paloma Moreda. 2013.** *Bases de Datos 1*. Alicante : Dpto. de Lenguajes y Sistemas Informáticos. Universidad de Alicante, 2013.
- Gómez, Yamilka. 2016.** Programa Analítico de la Asignatura Bases de Datos, 2016. [En línea] 2016. [http://eva.uci.cu/pluginfile.php/5003/mod\\_resource/content/2/Documentos%20generales/Progama%20anal%C3%ADtico.pdf](http://eva.uci.cu/pluginfile.php/5003/mod_resource/content/2/Documentos%20generales/Progama%20anal%C3%ADtico.pdf).
- Granados, John, y otros. 2014.** Las tecnologías de la información y las comunicaciones, las del aprendizaje y del conocimiento y las tecnologías para el empoderamiento y la participación como instrumentos de apoyo al docente de la universidad del siglo XXI. Ecuador : Universidad de Guayaquil, 2014. ISSN 1727-897X.
- Grosso, A. 2011.** Prácticas de software, Experiencias sobre la Ingeniería y Management del Software. [En línea] 2011. <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
- Guevara, Edelmira Belkis Soca. 2015.** El trabajo independiente en el proceso de enseñanza-aprendizaje. La Habana : RCIM, 2015. Vol. 7, 2. ISSN 1684-1859.
- Gutiérrez, Emmanuel. 2009.** *JavaScript: Conceptos básicos y avanzados*. Barcelona : Informática Técnica, 2009. ISBN: 978-2-7460-4341-1.
- Huaraca, Abner Valdez. 2013.** *Pruebas de Sistemas y Pruebas de Aceptación*. s.l. : FIS-UNICA, 2013.
- Infante, S. 2012.** Curso Django para perfeccionistas con deadlines. 2012.
- ITI. 2017.** Instituto Tecnológico de Informática. *Testeo de estrés y carga*. [En línea] 2017. [Citado el: 20 de Abril de 2017.] <http://www.iti.es/servicios/servicio/resource/7240/index.html>.
- Koopman, Bevan. 2016.** Information retrieval as semantic inference: a Graph Inference model applied to medical search. s.l. : Inf Retrieval J, 2016. Vol. 19, 6. ISSN: 10791-015-9268-9.
- Labrada, Evelyn y Aragón, Yaniel. 2013.** *Desarrollo del Módulo de Gestión de Reportes Estadísticos para el sistema AiresProxyAudit*. La Habana : s.n., 2013.
- Larman, Craig. 2004.** *UML y patrones. Introducción al análisis y diseño orientado a objetos*. La Habana : Félix Varela, 2004.

- Lozano, R. 2011.** www.thinkepi.net. [En línea] 12 de enero de 2011. <http://www.thinkepi.net/las-tictac-de-las-tecnologias-de-la-informacion-y-comunicacion-a-las-tecnologias-delaprendizaje-y-del-conocimiento>.
- Martínez, Alejandro y Martínez, Raúl. 2014.** Guía a Rational Unified Process. Albacete : Escuela Politécnica Superior de Albacete. Universidad de Castilla la Mancha, 2014.
- Mateos, Rodrigo Cepeda, Francisco, Cristina Marco De y Gordillo, Tello Serrano. 2009.** *DBCASE. “Una herramienta para el diseño de Bases de Datos.”*. Madrid : Facultad de Informática. Universidad Complutense de Madrid, 2009.
- Mato, Rosa María. 2007.** *Sistemas de Bases de Datos*. La Habana : Félix Varela, 2007. ISBN: 959-258-826-0.
- Microsoft. 2017.** Revisiones de código y estándares de codificación. [En línea] 2017. [Citado el: 10 de Abril de 2017.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
- Mifsuf, E. 2013.** Apache. España : s.n., 2013.
- Ministerio de Educación Superior. 2014.** Plan de estudios "D" Ingeniería en Ciencias Informáticas. La Habana : s.n., 2014.
- Oré, Alexander. 2009.** FUNCTIONAL TESTING - PRUEBAS FUNCIONALES. *CalidadSoftware.com*. [En línea] 2009. [Citado el: 2017 de Abril de 10.] [http://www.calidadsoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadsoftware.com/testing/pruebas_funcionales.php).
- Peralta, Claudia y 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 : Universidad de las Ciencias informáticas, 2014.
- PMOinformatica. 2016.** Pruebas de aceptación de software según el ISTQB. *PMOinformatica.com*. [En línea] 2016. [Citado el: 20 de Abril de 2017.] <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>.
- PostgreSQL. 2014.** PostgreSQL.org. [En línea] Diciembre de 2014. [Citado el: 20 de Noviembre de 2016.] <https://www.postgresql.org/about/press/presskit94/es/>.

- Pressman, Roger S. 2010.** *Ingeniería de Software, un enfoque práctico. Séptima Edición.* Madrid : s.n., 2010.
- PyCharm. 2015.** jetbrains. [En línea] Diciembre de 2015. <http://www.jetbrains.com/pycharm/>.
- Python. 2017.** Guía de estilo para el código Python – PEP 8 en Español. [www.recursospython.com](http://www.recursospython.com). [En línea] 2017. [Citado el: 12 de Abril de 2017.] [www.recursospython.com/pep8es.pdf](http://www.recursospython.com/pep8es.pdf).
- Ramez, Elmasri y Shamkant, Navathe. 2015.** *Fundamentals of Database Systems 7th Edition.* Boston, USA : Addison Wesley, 2015. 978-0133970777.
- Rastrepo, Jairo. 2007.** Análisis de los procesos básicos de un sistema de comunicaciones. [aut. libro] Jairo Restrepo Angulo. *Compendio de Telecomunicaciones.* Medellín : Sello Editorial Universidad de Medellín, 2007.
- Rodríguez, S. 2012.** *Módulo Informático para la Gestión de la Información Académica de los Estudiantes.* Matanzas : s.n., 2012.
- Rojo, Silvana del Valle y Oliveros, Alejandro. 2012.** *Requerimientos No funcionales para aplicaciones Web.* Argentina : INTEC – UADE, 2012. ISSN: 1850-2792.
- Russo, Patricia. 2011.** *Gestión documental en las Organizaciones.* Barcelona : UOC, 2011. ISBN: 978-84-9788-863-9.
- Saldaña, Gabriel. 2017.** Nethazard. [En línea] 2017. [Citado el: 30 de Abril de 2017.] <http://blog.nethazard.net>.
- Sánchez, Tamara Rodríguez. 2015.** Metodología de desarrollo para la Actividad productiva de la UCI v1.2. La Habana, Cuba : s.n., 6 de Marzo de 2015.
- Sarmiento, J. 2016.** UML: Diagrama de despliegue. Obtenido de Visión general de los diagrama de despliegue. [En línea] Febrero de 2016. <http://umldiagramadespliegue.blogspot.com/>.
- SEI. 2010.** Mejora de los procesos para el desarrollo de mejores productos y servicios. s.l. : Software Engineering Institute. Technical Report, 2010.
- Significado. 2014.** Significado. [En línea] Diciembre de 2014. [Citado el: 10 de Abril de 2017.] [www.significado.net/http/](http://www.significado.net/http/).

- Soler, Josep. 2010.** Entorno virtual para el aprendizaje y la evaluación automática en Bases de Datos. Girona : Universidad de Girona, 2010. Vol. Tesis Doctoral. ISBN 978-84-694-0260-3.
- Sommerville, Ian. 2011.** *Ingeniería de Software*. México : Pearson Educación de México, 2011. ISBN: 978-607-32-0603-7.
- TeachTerms. 2013.** TeachTerms. *Framework Definition*. [En línea] Marzo de 2013. <https://techterms.com/definition/framework>.
- Techopedia. 2017.** Modeling Language. [En línea] 2017. <https://www.techopedia.com/definition/20810/modeling-language>.
- Toll, Yuniet y Mendoza, Yilennis. 2007.** Propuesta de manual de procedimiento de Pruebas de Sistema y su aplicación en el Proyecto CICPC. [En línea] 2007. [Citado el: 20 de Abril de 2017.] [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_0717\\_07/1/TD\\_0717\\_07.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0717_07/1/TD_0717_07.pdf).
- UNESCO. 2017.** UNESCO. *Building peace in the minds of men and women*. [En línea] 2017. <http://en.unesco.org/themes/ict-education>.
- Urquiza, Jorge Reinier. 2013.** *Aplicación Web para estudiantes aventajados en la asignatura Idioma Extranjero III de la UCI*. La Habana : s.n., 2013.
- Visconti, Marcello y Astudillo, Hernán. 2011.** *Fundamentos de Ingeniería de Software*. s.l. : Universidad Técnica Federico Santa María, 2011.

## Anexos

### Anexo 1. Entrevista al cliente para conocer la necesidad del desarrollo de la propuesta de solución y definir los requisitos funcionales y no funcionales

Estimado profesor o estudiante: Se necesita de su cooperación en una investigación para una tesis de pregrado. Por ello, sería de gran ayuda que respondiera lo siguiente:

1. ¿Considera que los estudiantes logran desarrollar adecuadamente la habilidad diseñar MER, como se refleja en su plan de estudio?
2. Respecto a los tres últimos años académicos, ¿qué porcentaje de estudiantes responden incorrectamente la pregunta de diseño en los exámenes de la asignatura?
3. ¿Cuáles son las causas que podrían estar influyendo en el desarrollo de esta habilidad?
4. ¿Existe alguna herramienta en la Universidad para el diseño de MER?
5. ¿Cuáles son sus características?
6. A partir de estas características, ¿considera que la herramienta posee lo necesario para diseñar MER partiendo de la teoría de los SBDR?
7. ¿Considera que el módulo para el diseño de MER deba estar integrado a una plataforma educativa?
8. ¿La plataforma educativa *RDB-Learning* permite el diseño de MER?
9. ¿Cómo debería realizarse la revisión y evaluación de los ejercicios resueltos por los estudiantes en la plataforma *RDB-Learning*?
10. ¿Qué otras características, considera que deba presentar el módulo, en cuanto a la usabilidad, seguridad, interfaz u otro aspecto que garantice su calidad?

**Muchas gracias por su colaboración.**

## Anexo 2. Encuesta al profesor para evaluar el nivel de aceptación del MDMER

Estimado profesor: Se necesita de su cooperación en una investigación para una tesis de pregrado para evaluar el nivel de aceptación de una herramienta web, por ello, sería de gran ayuda que respondiera lo siguiente, con la sinceridad y seriedad que el proceso necesita.

Respecto al Módulo para el diseño de modelos entidad relación (MDMER) en la plataforma *RDB-Learning*, emita su valoración respecto a los siguientes parámetros: **MA**, si valora el parámetro como Muy Adecuado; **A**, si lo valora como Adecuado; **PA**, si considera que es Poco Adecuado; o **NA** si considera que es No Adecuado. Marcar con una X

No	Parámetros	Valoración			
		MA	A	PA	NA
1	Tratamiento de los aspectos teóricos de la asignatura en el MDMER				
2	Nivel de apoyo que brinda al profesor el uso del MDMER para la orientación y evaluación de ejercicios				
3	Contribución del MDMER al tema de diseño de MER en la asignatura SBD I				
4	Presentación de una interfaz agradable e intuitiva para el usuario				
5	Usabilidad del módulo				