Universidad de las Ciencias Informáticas

Facultad 1

Módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*

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

Autora: Eilen Castrillo Hernández

Tutores: M. Sc. Yaniel Lázaro Aragón Barreda

Ing.Osmel Mojena Dubet

La Habana, julio 2020

"Año 61 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo: Eilen Castrillo Hernández, con carné de identidad 97012708279, soy la autora principal del trabajo de diploma titulado: “Módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma RDB-Learning”. El cual ha sido desarrollado como parte del trabajo en el Departamento de Informática, 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 2020.

_____

Autor

Eilen Castrillo Hernández

_____ _____

Tutor **Tutor**

M. Sc. Yaniel Lázaro Aragón Barreda Ing.Osmel Mojena Dubet

DEDICATORIA

Que hoy me convierta en una ingeniera es gracia a muchas personas pero principalmente a mis padres, quienes siempre han estado conmigo y no han dejado ni un solo día de preocuparse por mí, a ellos les dedico esta victoria y sé que están muy orgullosos de la persona que soy.

RESUMEN

La incorporación de las Tecnologías de la Información y las Comunicaciones (TIC) al proceso de enseñanza-aprendizaje en todos los niveles ha servido de apoyo en la formación de los estudiantes. Los objetos de aprendizaje se encuentran entre los recursos que contribuyen a ello, los que pueden ser creados basándose en las especificaciones de las herramientas E-learning. La Universidad de las Ciencias Informáticas cuenta en su Plan de estudios con la asignatura Sistemas de Bases de Datos I, una de las habilidades en la cual los estudiantes presentan dificultades es en la transformación del diagrama entidad relación al diagrama relacional, siendo una de las principales causas, la falta de tecnologías que faciliten la práctica. La herramienta utilizada en la asignatura no permite el desarrollo de esta habilidad; por lo cual, no brindan apoyo a los estudiantes para desarrollar dicha práctica. La presente investigación tiene planteada como objetivo desarrollar un Módulo de transformación del Modelo Entidad-Relación al Modelo Relacional en la plataforma *RDB-Learning* para el proceso de enseñanza-aprendizaje de Sistema de Base de Datos I,que permita solucionar ejercicios de transformación de diagramas y la creación de modelos de datos a partir de esas relaciones resultantes. Este módulo basa su funcionamiento en la tecnología e-learning, al permitir la retroalimentación entre los usuarios(los estudiantes con sus profesores). Para lograrlo se analizaron herramientas informáticas con características similares a la solución y para garantizar una homogeneidad en todos los módulos de esta herramienta se estudiaron los métodos y herramientas utilizados en la creación de la misma.

Palabras claves: e-learning, transformación, modelo, retroalimentación, módulo, relaciones resultantes

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Las tecnologías de Información y Comunicación (TIC) son el conjunto de tecnologías desarrolladas en la actualidad para una información y comunicación más eficiente, las cuales han modificado tanto la forma de acceder al conocimiento como las relaciones humanas. Estas han transformado los parámetros de obtención de información por medio de las tecnologías de la comunicación (diario, radio y televisión), a través del desarrollo de internet y de los nuevos dispositivos tecnológicos (Chen, 2019).

En el mundo actual los diferentes sectores de la sociedad disponen cada vez de más información, en función de gestionar todos los procesos. En este sentido, saber almacenar estos datos, se hace una necesidad. Desde el punto de vista tecnológico el uso de bases de datos brindan una solución al permitir almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. Uno de los tipos de bases de datos usadas con este fin son lasBase de Datos Relacionales (BDR), dominadas por distintos gestores a través del lenguaje SQL, en gran medida (Gutiérrez, 2017).

Una de las formas de representar la estructura de las BDR es el Modelo Entidad-Relación (MER) al permitir representar las entidades relevantes de un sistema de información mediante diagramas, así como sus interrelaciones y propiedades (Leiva, 2015). Sin embargo, este modelo necesita ser transformado al Modelo Relacional (MR), con el fin de organizar y gestionar los datos almacenados en una BDR a partir de tablas compuestas por filas, o tuplas, y columnas o campos. El MR se distingue de otros modelos, por ser comprensible para el usuario inexperto, y basarse en la lógica de predicados para establecer relaciones entre distintos datos (Cruz, 2018).

Parte de los problemas de administración en los sistemas de bases de datos en la actualidad inician en bases de datos mal diseñadas. La calidad requerida en la fase de diseño, transita desde la obtención del MER hasta su correcta transformación a un MR. Para realizar este proceso, existen un conjunto de reglas definidas para cada uno de los elementos del MR, su conocimiento y empleo adecuado evitan futuras anomalías asociadas a la información que se almacena en las base de datos (Coronel, y otros, 2011).

El área del conocimiento relacionada con las Bases de Datos, es de suma importancia en la formación académica de los Ingenieros en Informáticas y carreras afines. De la correcta comprensión de los conceptos involucrados en esta, depende la calidad de los sistemas de información que se desarrollan.

Como resultado, se han diseñado e implementado herramientas computacionales que buscan apoyar el proceso de impartición de las asignaturas que tratan la temática de las BDR.

En este escenario académico se encuentra la Universidad de las Ciencias Informáticas (UCI), centro educacional cuya misión social está encaminada a contribuir en la continua informatización de la sociedad, en un contexto nacional, donde todas las entidades y organismos informatizan sus procesos y por tanto necesitan manejar información. En la actualidad, la universidad lleva a cabo la implementación del Plan de Estudio E que demanda la incorporación de las tecnologías educativas al proceso y el aumento de las actividades prácticas.

La enseñanza de BDR forma parte de este plan de estudios, con el apoyo de una tecnología para el apoyo del proceso de enseñanza aprendizaje que desarrollan los estudiantes guiados por sus profesores. La plataforma para el Aprendizaje de Bases de Datos Relacionales (*RDB-Learning* por sus siglas en inglés), permite a los estudiantes resolver ejercicios (creados por los profesores) relacionados con el diseño de base datos relacionales y la implementación de sentencias para la manipulación de datos desde una interfaz gráfica (Aragón, 2018).

Dentro de esta plataforma, una vez seleccionado el ejercicio según sea el tipo (diseño o implementación), los estudiantes pueden leer la descripción del mismo y seleccionar la opción de resolver. Al crear una respuesta, tienen la posibilidad de enviarla de forma privada, a ser revisada por el profesor, o de forma pública, a un foro de discusión asociado al ejercicio. Si la respuesta es enviada al profesor, este es notificado con el fin de que emita una evaluación (nota) y una retroalimentación, ante errores cometidos. En caso de ser enviada a un foro, la solución estará visible para todos los usuarios que deseen comentar y emitir sus criterios.

La plataforma, brinda la posibilidad a los estudiantes de guardar las respuestas a ejercicios que por algún motivo no hayan podido terminar. En el caso de los profesores contarán en todo momento con un histórico de los resultados evaluados y los ejercicios realizados por estudiantes, posibilidad que permite llevar a cabo una gestión del conocimiento en pos de contribuir al desarrollo de las habilidades y al mejoramiento continuo de la plataforma, adaptando la misma a las necesidades detectadas (Aragón, 2018).

Después de un estudio exploratorio realizado a la plataforma fueron detectadas las siguientes insuficiencias:

- La realización de ejercicios de implementación dependen de un MR no generado por la plataforma, siendo necesario que los profesores dominen la estructura del fichero que debe ser creado para obtener el mismo, lo que agrega un mayor grado de dificultad al uso de este módulo.
- Existe un módulo destinado a resolver ejercicios de diseño del MER y uno a la implementación de sentencias SQL que no se relacionan entre sí, lo que imposibilita el desarrollo de ejercicios integradores que abarquen ambos temas y permitan al estudiante enlazar los conocimientos.
- La plataforma no provee facilidades para el estudio individual del contenido asociado a la transformación del MER al MR, creando una dependencia de los estudiantes hacia el profesor a la hora de realizar ejercicios relacionados con el tema.

Otro elemento a destacar es que la asignatura Sistemas de Bases de Datos I, encargada dentro del plan E de impartir estos contenidos, solo cuenta con una conferencia y dos actividades prácticas (Guillén Pérez, y otros, 2019)la primera, es una clase práctica donde los estudiantes realizan ejercicios orientados por el profesor y aplican las reglas de transformación en sus libretas. La segunda, un laboratorio donde diseñan un MR directamente y observan las transformaciones que realiza de forma automática la herramienta DBDesigner sin necesidad de que tengan dominio de estas. En lo adelante, deben ser capaces de ejercitar este contenido desde su auto preparación y para ello depende de la orientación de ejercicios y su posterior revisión por parte del profesor.

Es por ello que, si se tienen en cuenta las insuficiencias de la plataforma *RDB-Learning* y la necesidad exigida por el plan de estudios E de contar con recursos tecnológicos que faciliten el aprendizaje y la práctica de todos los contenidos que recibe el estudiante desde su auto preparación, se hace necesario desarrollar el siguiente **problema de investigación**: ¿Cómo contribuir al aprendizaje del proceso de transformación de un Modelo Entidad-Relación al Modelo Relacional desde el empleo de las Tecnologías de la Información y las Comunicaciones?

A partir del problema planteado se propone como **objeto de estudio** el proceso de transformación de un Modelo Entidad-Relación al Modelo Relacional, siendo necesario profundizar en el siguiente **campo de acción**, el proceso de transformación de un Modelo Entidad-Relación al Modelo Relacional en la plataforma *RDB-Learning*.

Con el **objetivo general** dedesarrollar un módulo para la transformación del Modelo Entidad-Relación al Modelo Relacional en la plataforma *RDB-Learning* que contribuya al proceso de aprendizaje de este contenido en la asignatura Sistemas de Bases de Datos I se definen las siguientes **preguntas científicas**:

1. ¿Cuáles son los supuestos teóricos que sustentan el proceso de transformación del Modelo Entidad-Relación al Modelo Relacional y el empleo de la tecnología para su aprendizaje*?*
2. ¿Qué características tienen las herramientas informáticas que permiten la transformación del Modelo Entidad-Relación al Modelo Relacional?
3. ¿Qué elementos deben tenerse en cuenta durante el análisis y diseño del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*?
4. ¿Cómo materializar, en términos de componentes y código fuente, los diseños especificados para el módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*?
5. ¿Qué resultados se obtendrán al validar el módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*?

Con el fin de responder estas preguntan se definen las siguientes **tareas de investigación**:

1. Sistematización de los supuestos teóricos que sustentan el proceso de transformación del Modelo Entidad-Relación al Modelo Relacional y el empleo de la tecnología para su aprendizaje.
2. Análisis de las características de las herramientas informáticas que permiten la transformación del Modelo Entidad-Relación al Modelo Relacional.
3. Análisis y diseño de las funcionalidades del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*.
4. Implementación de las funcionalidades del módulo transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*.
5. Validación del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*.

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

Teóricos:

- **Analítico-Sintético:** es utilizado con el fin de realizar un análisis bibliográfico para establecer las bases teóricas en relación con el proceso de transformación del MER al MR y el empleo de las tecnologías como apoyo al proceso de aprendizaje de este contenido**.**
- **Modelación:** para formalizar una representación del proceso estudiado que sirva como guía durante el desarrollo del módulo, al permitir identificar las características y relaciones fundamentales.

Empíricos:

- **Entrevista:** se utiliza para conocer las necesidades actuales que conllevan al desarrollo de la propuesta de solución, en función de conocer las particularidades y restricciones que se deben tener en cuenta. Esta fue realizada a profesores de la asignatura (Anexo1).
- **Encuesta:** se utilizó para conocer el nivel de aceptación de la plataforma por los estudiantes y las facilidades que les brinda (Anexo2).
- **Observación:** con el fin de valorar el funcionamiento de los sistemas existentes en la actualidad que realizan el proceso de transformación del MER al MR.

La presente investigación está estructurada de la siguiente forma:

Capítulo 1: El proceso de transformación del Modelo Entidad-Relación al Modelo Relacional y el empleo de la tecnología para su aprendizaje: En este capítulo se plasman los conceptos asociados al proceso de transformación. Además, se analizan herramientas que permiten la obtención del Modelo Relacional y se plantea el entorno tecnológico a emplear.

Capítulo 2: Análisis y diseño del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*: En este capítulo se documentan los artefactos y elementos del diseño de la solución, que permiten una mejor comprensión de las funcionalidades desarrolladas. Además, se expone las principales características tenidas en cuenta durante la implementación del módulo.

Capítulo 3: Implementación y prueba del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*:En este capítulo se describe la implementación de la solución informática, así como los componentes que la integran. Además, se presentan los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

CAPÍTULO 1. El proceso de transformación del Modelo Entidad-Relación al Modelo Relacional y el empleo de la tecnología para su aprendizaje

Con el objetivo de lograr un buen modelado de la solución en este capítulo se analizan varios conceptos asociados con el dominio del problema. Se realiza un análisis de las herramientas que presentan características similares al módulo que se desarrollará, además se caracteriza la metodología y el entorno de desarrollo a utilizar para dar solución al problema.

1.1. Conceptos asociados al dominio de la investigación

A continuación se exponen un conjunto de conceptos asociados al diseño de Bases de Datos Relacionales que permiten una mejor comprensión del tema de la investigación.

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 e interrelaciones (Costal, 2014).

Lasentidades, representadas por un rectángulo, son la representación de un objeto o concepto del mundo real que se describe en una base de datos. Estas entidades pueden ser fuertes o débiles. Cada entidad está constituida por uno o más atributos (Alegsa, 2018).

En bases de datos, un **atributo** representa una propiedad de interés de una entidad. Los atributos se describen en la estructura de la base de datos empleando un modelo de datos (Alegsa, 2018) .Estos atributos pueden ser de distintos tipos (llave, simple, compuesto, multivaluado, derivado). En un diagrama los atributos se representan con círculos que descienden de la entidad.

Interrelación: es una relación o vínculo entre dos o más entidades que describe alguna interacción entre las mismas. Las relaciones evitan redundancia de datos guardados en las tablas. Estas pueden ser binarias o n-arias (Fernández, 2016).

Estas interrelaciones tienenuna **cardinalidad** la cual es el número de instancias o elementos de una entidad que pueden asociarse a un elemento de la otra entidad relacionada. Esta cardinalidad puede ser 1 a 1,1 a Muchos o Muchos a Muchos (Galeano, 2016).

Cada entidad tiene un atributo denominado **llave**, la cual hace distinguir a esta entidad de los demás registros (Riccio, 2016).

Los conceptos descritos hasta el momento también están presentes en el Modelo Relacional, este proporciona una estructura de los datos que consiste en un conjunto de relaciones con objeto de representar la información que nos interesa del mundo real (Costal, 2014).

Llaves foráneas permiten establecer conexiones entre tuplas de las relaciones. Una clave foránea tiene el conjunto de atributos de una relación que referencian la clave primaria de otra relación(o incluso de la misma relación) (Costal, 2014).

Para obtener un MR a partir del MER correspondiente se hace necesario conocer un conjunto de **reglas**, estas son un modo establecido de ejecutar algo, es decir, un conjunto de operaciones que deben llevarse a cabo para realizar una inferencia o deducción correcta**Fuente especificada no válida.**.

1.2. Reglas de transformación del Modelo Entidad-Relación al Modelo Relacional

Numerosos procesos de ingeniería pueden ser modelados como una transformación de estructuras de datos. Más bien, la producción de un esquema puede ser considerada como una derivación de este esquema desde un esquema fuente a través de lo cual se ejecuta una cadena de operaciones elementales llamadas **reglas de transformación**.

Las reglas de transformación, estudiadas en la asignatura sistemas de bases de datos I que deben tenerse en cuenta como parte del objeto de estudio de la presente investigación (Date, 2003)son las siguientes:

Transformación de entidades

Se transforma cada entidad en un esquema relacional, que va a tomar el nombre de la entidad y sus atributos y llave primaria.

En el caso que estas entidades tengan atributos compuestos se deben representar unívocamente por sus atributos simples.

Si son multivaluados se crea un esquema relacional nuevo que se recomienda tener un nombre en función de la relación del atributo entidad, que tendrá la llave de la entidad a la cual pertenece y el atributo propio, ambos como llave de este nuevo esquema.

Si el atributo es derivado no se agrega en el esquema, es decir quedan eliminados en esta parte del diseño.

Transformación de interrelaciones

Estas se transforman según la cardinalidad de los conjuntos de entidades asociados.

- Interrelaciones Uno-Uno: La llave de una entidad pasa a formar parte del esquema que genera la otra, se le llama llave foránea.
- Interrelaciones Uno-Muchos: se transforma incluyendo la llave primaria de la entidad con cardinalidad uno en el esquema correspondiente de la entidad con cardinalidad mucho, como un atributo o atributos simples. Si la relación tiene atributos se incluirán también en el lado Muchos.
- Interrelaciones Muchos-Muchos: En estos casos se crea un nuevo esquema que lleva como nombre una combinación de las dos entidades relacionadas, tomando como llave primaria una combinación de los atributos que constituyen las llaves primarias de los conjuntos de entidades asociados por la interrelación e incluye, además, sus atributos descriptivos si los tiene.

En el caso de que las entidades sean débiles se crea un esquema, el cual incluye como llave primaria el identificador del conjunto de entidades propietario y el discriminante del conjunto de entidades débiles, e incluye además el resto de los atributos del conjunto de entidades débiles.

- Generalización y Especialización: Crear un esquema para el conjunto de entidades genérico, con sus atributos correspondientes y un esquema para cada conjunto de entidades específico con sus atributos más la llave primaria del esquema del conjunto de entidades genéricas.
- Agregación: La agregación en sí no genera ningún esquema en el Modelo Relacional, para las que se crea un nuevo esquema es para las interrelaciones que la asocian con otros objetos del Modelo Entidad Relación. Estas interrelaciones son transformadas de la misma forma que las interrelaciones n-arias, o sea, se crea un nuevo esquema con los identificadores de los conjuntos de entidades asociados. Para los efectos de la transformación, se puede tomar la agregación como un conjunto de entidades que tiene como atributos e identificador, la combinación de todos los identificadores de los conjuntos de entidades que esta engloba.

- Interrelaciones recursivas: Las interrelaciones recursivas son transformadas de la misma manera que las interrelaciones Uno-Uno, Uno-Muchos o Muchos-Muchos dependiendo de su cardinalidad máxima.

La aplicación de estas reglas de transformación a un MER permite la obtención de un MR. Este último, provee una estructura cercana a la manera física en que los datos de una base de datos están almacenados. Es por ello la importancia de un proceso de transformación correcto que contribuya a la no existencia de anomalías y redundancia en la información que se almacena.

En la actualidad existen un conjunto de aplicación que permiten la obtención de un MR a partir de la transformación del MER es por ello se hace necesario realizar un estudio de los mismos para determinar sus características y posible aplicación o no como solución del problema de investigación que se abordad en el presente documento.

1.3. Aplicaciones informáticas para la obtención del Modelo Relacional a partir del Modelo Entidad-Relación

A continuación se relacionan un conjunto de aplicación informáticas que permiten la obtención del Modelo Relacional a partir del Modelo Entidad-Relación. Este estudio se realiza a partir de una revisión documental y la observación del funcionamiento de algunas de ellas. Es válido destacar que las mismas han sido creadas con fines educativos y por tanto son tenidas en cuentas para la presente investigación.

SUITEDB

Es un sistema de software desarrollado con el objetivo de favorecer la comprensión de los modelos Entidad-Relación y Relacional, es considerada una herramienta learning. Consta actualmente de dos partes principales a saber: editor de Diagramas Entidad Relación y editor de Diagramas Relacionales. En el primero, los estudiantes pueden crear sus Diagramas Entidad Relación. Además, es posible transformar, automáticamente, estos diagramas en su correspondiente Modelo Relacional. En el segundo, los estudiantes pueden crear Diagramas Relacionales. Esta herramienta para realizar la transformación no tiene en cuenta la relación de Mucho-Mucho, pues la única restricción que tiene en cuenta a la hora de relacionar dos entidades es especificar la llave foránea. Es una aplicación web, es multiplataforma y se emplea bajo licencia privativa. Esta herramienta cuenta con la opción de verificación, en caso de que encuentre algún error el sistema informa al usuario en dónde se encuentran los errores. No cuenta con un componente generarJSON, por lo que el modelo resultante no se puede emplear para la realización de consultas SQL. Permite guardar y exportar el diagrama, permite la gestión de usuarios (Palma-Orozco, y otros, 2016).

CasER

Su característica principal es la de asistir en la creación de un esquema conceptual de alto nivel y asistir en la creación de un esquema lógico y físico, es considerada una herramienta learning. Fue diseñada como aplicación de escritorio donde solo se ejecutan bajo el sistema operativo Windows y la licencia es pública. Fue desarrollada con el lenguaje de programación Java. Para la conversión al modelo físico la herramienta brinda la opción de generar el modelo relacional automáticamente o un pasaje paso a paso con la intervención del usuario, este último consiste en que el usuario puede optar entre convertir una relación en dos tablas o en tres, pero no cuenta con la opción de detectar errores o verificar el trabajo. No permite la generación de modelos de datos para ser empleados en la generación de consultas SQL. Permite el almacenamiento del trabajo que se realiza, no permite la gestión entre los usuarios (Rius, 2016).

DBDAp

Es una herramienta gráfica de apoyo a la docencia de base datos y almacenes de datos, esta permite definir esquemas conceptuales de base de datos y transformarlos automáticamente a esquemas relacionales apropiados. No es una aplicación web, fue desarrollada con el lenguaje de programación Java. Incluye un modo de transformación paso a paso de MER a MR, junto con la posibilidad de acceder a explicaciones, consejos y sugerencias. Permite obtener el modelo de datos y su licencia es pública. Permite guardar las respuestas en las herramientas, pero no exportarlas ni editarlas. No permite la gestión de usuarios (Ilarri, 2017).

MySQL Workbench

Es una herramienta con una interfaz visual unificada para los arquitectos de base de datos, desarrolladores y administradores de bases, no es considerada una herramienta learning y tampoco es una aplicación web. Esta herramienta es multiplataforma y fue desarrollada bajo el lenguaje de programación C++.Tiene licencia pública y genera modelos de datos. No cuenta con ninguna ayuda ni evaluación del trabajo .Permite guardar el trabajo realizado y no permite la gestión de usuario(Olano, 2017).

A modo de resumen se presenta la siguiente tabla donde se tienen en cuenta los siguientes parámetros:

-E-learning: especifica si la aplicación es o forma parte de una plataforma educativa.

-Tipo de aplicación: si es una aplicación web o de escritorio.

-Multiplataforma: si se funciona en diferentes sistemas operativos (Linux-Windows).

-Evaluación: El empleo de ayudas o tips durante la obtención del MR o la detección de errores o evaluación del trabajo.

-Modelo de datos: Si permiten obtener el modelo de datos y emplearlo para la realización de consultas SQL.

-Licencia: las aplicaciones pueden ser públicas en caso de que no necesiten de un pago para ser utilizadas, y privativas en aquellas que si lo requieren.

-Gestión de usuarios: Si permite la gestión de usuarios para poder controlar el acceso al sistema y dar seguimiento al trabajo que realizan.

Tabla 1. Resumen de las herramientas estudiadas

Herramientas	E-learning	Tipo de aplicación	Multiplataforma	Evaluación	Modelo de datos	Licencia	Gestión de usuario
SUITEDB	Sí	Web	Sí	Sí	No	Privativa	Sí
CasER	Sí	Escritorio	No	No	No	Pública	No
DBDAp	Sí	Escritorio	No	Sí	Sí	Pública	No
Workbench	No	Escritorio	Sí	No	Sí	Pública	No

Conclusiones del análisis de las soluciones existentes

Después de analizadas las herramientas anteriores se puede concluir que todas fueron creadas con el mismo propósito de apoyar y hacer más fácil el diseño de bases de datos a través de la transformación del Modelo Entidad Relación al Modelo Relacional, presentando estas, características y funcionalidades similares. Las diferencias de estas herramientas radican en:

- En el caso de la herramienta SUITEDB se emplea bajo licencia privativa, no permite generar modelo de datos, no tiene en cuenta la relación Mucho-Mucho.
- DBDAp no es una aplicación web y no es multiplataforma.

- CasER no es una aplicación web, no es multiplataforma, no genera modelo de datos ni evalúa las respuestas.
- MySQL Workbench no es una aplicación web, no es e-learning ni evalúa lo realizado por el estudiante.
- Ninguna de las herramientas estudiadas permite la gestión de usuario, solo SUITEDB, pero al igual que las demás, no se permiten la interacción entre los usuarios, no permiten la creación de ejercicios por parte del profesor como guías para el estudiante y con varios niveles de dificultad, ni la evaluación de los mismos por parte del profesor y la retroalimentación ya sea para aclarar los errores o resaltar sobre los elementos correctos, ni llevar a cabo un histórico de qué ejercicio hace cada estudiante y la evaluación obtenida.

En la UCI existe la herramienta *RDB-Learning*, teniendo en cuenta su aplicación en el ámbito educativo se decide, a partir de que las herramientas anteriores no cubren los elementos necesarios para dar cumplimento al problema de investigación planteado, desarrollar un módulo para la transformación del Modelo Entidad-Relación al Modelo Relacional. Este módulo se creó para la plataforma *RDB-Learning* con el fin de que contribuya en el proceso de aprendizaje de este contenido en la asignatura Sistemas de Bases de Datos I**.** Para ello se describe a continuación el entorno tecnológico bajo el que se desarrollará la propuesta de solución.

1.4. Entorno tecnológico para el desarrollo de la propuesta de solución

Al desarrollar un módulo para la aplicación *RDB-Learning* el mismo debe ser compatible con las tecnologías, herramientas y metodología empleadas durante el desarrollo del mismo. A continuación se exponen un resumen de las mismas.

Metodología

La metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. En un proyecto de desarrollo de software la metodología ayuda a definir: ¿Quién debe hacer, qué, cuándo y cómo debe hacerlo? Es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito (Zúñiga Ángeles, 2017).

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, entre otros), se decide hacer una variación de la metodología Proceso Unificado Ágil (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 de Madurez de Capacidad Integrado para el Desarrollo (CMMI-DEV) v2, para garantizar las buenas prácticas en función de un software de calidad (Sánchez, 2015). Para el desarrollo de la propuesta de solución, se usa la metodología AUP, en su variación para la UCI, ya que por esta es la que se rige la plataforma, basándose en el escenario 2 que propone esta metodología.

En este escenario se deben realizar el Modelo Conceptual y el Modelo de Caso de Uso del Negocio, este aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Herramientas y Lenguajes

Las herramientas y lenguajes que se utilizarán, son los empleados durante el desarrollo de la plataforma *RDB-Learning*, para garantizar así la compatibilidad tecnológica del módulo.

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 utilizará Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) versión 2.1. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un plano o modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguaje de programación, esquemas de base datos y componentes de software reutilizables (Cortes Iglesias, y otros, 2017).

Herramienta de modelado

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE,por sus siglas en inglés, *Computer Aided Software Engineering)*, tienen como objetivo primordial representar objetos de datos de negocios, sus relaciones, y ayudan a comprender mejor la forma en que fluyen estos objetos de datos entre distintas zonas de negocio en el seno de la compañía (Menéndez, y otros, 2018). La herramienta CASE que se utilizará es el Visual Paradigm en la versión 8.0. Esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación (Correa, 2016).

Herramienta para el 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, para esto se usará el cliente Git en su versión 2.8, el cual, según Saldaña (2017) es un sistema de control de versiones distribuido que no depende de acceso a la red.

Marco de trabajo

Entorno de trabajo o marco de trabajo (del inglés *framework*) es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar (Gandarillas, 2017). Para el desarrollo del módulo se utiliza el marco de trabajo Django en su versión 1.9.5. Este es un *framework* web de alto nivel que permite el desarrollo rápido de sitios web seguros. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago (del Pino, 2018).

Lenguaje de programación

Un lenguaje de programación es diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (Olarte, 2018).

Lado del servidor

Para el desarrollo de la propuesta de solución se utilizará como lenguaje del lago del servidor Python en su versión 3.4.1. Este 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. Su elegante sintaxis y tipos de datos dinámicos junto con su naturaleza interpretada, hacen de este un lenguaje ideal para eldesarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas (Fernández, 2016).

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, esJavaScript, en su versión 1.8.5. Este fue ideado para conceder a la web de capacidades interactivas que le ayudarán a dar el salto al siguiente nivel permitiendo crear una interfaz de usuario activa, lo que ofrece retroalimentación a los visitantes según navegan por sus páginas. Con JavaScript se puede crear sobre la marcha páginas HTML personalizadas, dependiendo de las acciones ejecutadas por el usuario (Robledano, 2019).

Librerías

Las librerías son un grupo de archivos que tienen una funcionalidad pre-construida por terceros, y que pueden ser usadas por cualquier ejecutable. Las librerías contienen en su interior variables y funciones, se conoce como librerías (o bibliotecas) a cierto tipo de archivos que se pueda importar o incluir en el programa. Estos archivos contienen las especificaciones de diferentes funcionalidades ya construidas y utilizables, como por ejemplo leer del teclado o mostrar algo por pantalla entre muchas otras más (Chaluisa, 2017). Las librerías utilizadas en el desarrollo del módulo son:

JQuery es una biblioteca de JavaScript que simplifica la forma de desarrollar aplicaciones web. Las aplicaciones que utilizan jQuery suelen necesitar menos tiempo y menos código que las aplicaciones hechas con JavaScript puro. Por este motivo, JQuery es muy popular y se utiliza en muchaspáginas web. Permite manipular elementos del DOM (textos, imágenes, enlaces, entre otros, cambiar el diseño CSS a través de un código muy conciso y sencillo (Domínguez, 2018). Para la implementación del módulo se emplea la librería JQuery en su versión 3.1.1.

También es empleada la librería GoJS en su versión 1.7.5. Es una biblioteca JavaScript muy interesante ya que consta de múltiples características para la implementación de diagramas que además son interactivos, puedes hacer uso de nodos, plantillas, grupos totalmente personalizables, con gojs podrás brindarle al usuario una experiencia muy agradable permitiéndole arrastrar, soltar, copiar, pegar, editar texto, entre otros. 1.5.8 Formato de intercambio de datos (López, 2015).

JSON: este es el acrónimo para *JavaScript Object Notation*, es un estándar basado en texto plano para el intercambio de información. Por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas. La ventaja de JSON al ser un formato que es independiente de cualquier lenguaje de programación, es que los servicios que comparten información por éste método, no necesitan hablar el mismo idioma, es decir, el emisor puede ser Java y el receptor PHP, cada lenguaje tiene su propia librería para codificar y decodificar cadenas de JSON(Cadenas, 2016). Se usará el formato JSON 3 para el desarrollo del módulo.

Entorno de desarrollo integrado

Los Entornos de Desarrollo Integrado (IDE) son programas destinados a manejar proyectos enteros, es decir a manejar muchos archivos. Los IDE analizan todos los ficheros del proyecto, y nos sugieren el autocompletado de código basado en todos los archivos del proyecto, no solo en el lenguaje. Además de esto integran muchas otras herramientas, como compatibilidad con sistemas de control de versiones como lo puede ser Git (Palomares, 2017).

PyCharm es un IDE (entorno de desarrollo integrado) multiplataforma utilizado en el ámbito de la programación, cuenta con dos versiones, una que se divide en comunidady edición educativa que se liberan bajo licencia de Apache y la otra es la edición profesional lanzada bajo licencia propietaria.

PyCharm viene con una consola dePythondonde puede escribir los scripts a medida que se ejecuta (Naranjo, 2018). Para la implementación del módulo se utiliza el IDE PyCharm en su versión 2019.2.3.

Servidor de base de datos

Un servidor de base de datos es un programa que provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio (Alegsa, 2019).

PostgreSQL es un sistema de gestión de base de datos de objeto general y relacional, el sistema de base de datos de código abierto más avanzado. PostgreSQL fue diseñado de modo que pudiera ejecutarse en varias plataformas. Es un software gratuito y de código abierto. Su código fuente está disponible bajo licencia PostgreSQL (postgresqltutoria, 2020). Como sistema gestor de base de datos se emplea PostgreSQL en su versión 9.4.

Servidor web

Un servidor Web es un programa que utiliza el protocolo de transferencia de hiper texto, HTTP para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras (Rouse, 2016).

Para el alojamiento del sistema se emplea el servidor web Apache 2.2. Elservidor web Apache es un software de servidor web gratuito y de código abierto, es multiplataforma (funciona tanto en servidores Unix como en Windows). Es uno de los servidores web más antiguos y confiables. Presenta parches de seguridad regulares y actualizados con frecuencia, es flexible debido a su estructura basada en módulos (Hostinger, 2019).

Módulo de apache para Python.

La Interfaz de puerta de enlace del servidor web (WSGI son las siglas de Web Server Gateway Interface)**.** Es una especificación que describe cómo se comunica un servidor web con una aplicación web**,** y cómo

se pueden llegar a encadenar diferentes aplicaciones web para procesar una solicitud/petición (Alonso, 2019). Para la propuesta de solución se usa esta interfaz en su versión 4.5.10.

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:

- Al definir los conceptos asociados al desarrollo del módulo que se pretende realizar, se alcanza un una mayor comprensión de la propuesta de solución.
- El análisis de los principales elementos asociados al proceso de transformación del Modelo Entidad Relación al Modelo Relacional permite definir cómo deben ser tratadas las reglas que establece dicho proceso desde la propuesta de solución.
- El análisis de los sistemas homólogos, permite identificar las tendencias en cuanto al desarrollo de herramientas informáticas para el diseño del Modelo Relacional, y se identificaron las deficiencias que impiden el uso de estas o la integración con la plataforma *RDB-Learning*.
- El estudio 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 del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*

Un elemento a tener en cuenta durante el desarrollo de un software, es el nivel de 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. En este capítulo se presentan los resultados obtenidos 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.

2.1. Análisis de la propuesta de solución

En esta fase se presentan las características generales del módulo y se realiza un análisis de los conceptos fundamentales asociados al proceso de obtención del MR desde en la asignatura SBD I en la UCI. Estos conceptos se relacionan a través de un modelo conceptual.

2.1.1. Descripción de la solución propuesta

La solución propuesta constituye un módulo para la plataforma de aprendizaje de base de datos relacionales *RDB-Learning*. El mismo permitirá la transformación del Modelo Entidad-Relación al Modelo Relacional y la generación de modelos de datos. En este módulo los profesores serán los responsables de la gestión de los ejercicios (crear, modificar, eliminar, listar y mostrar detalles de los mismos). El estudiante tendrá la posibilidad de visualizar estos ejercicios de transformación y luego resolverlos. Este contará con un botón de ayuda donde se mostrarán todas las reglas de transformación explicadas en el capítulo 1, para que el estudiante pueda consultarlas en caso que tenga alguna duda. Para realizar esta transformación el módulo permitirá transformar cada una de las entidades del ejercicio, donde el estudiante debe especificar el nombre de la entidad, sus atributos y la llave primaria, el sistema validará que la entidad introducida exista en el diagrama entidad relación, ofreciéndole una retroalimentación al usuario (la entidad no existe). Después de transformadas las entidades permitirá transformar las relaciones entre ellas, donde el estudiante debe seleccionar el tipo de relación. Una vez seleccionado, si la relación es de Uno-Uno o Uno-Mucho, se debe especificar las entidades relacionadas, después se mostrará dos listas con los atributos de esas entidades para seleccionar cuál de ellos pasa a ser llave foránea y de que entidad. Si el tipo de relación seleccionado es de Mucho-Mucho o es una relación con

recursividad, el estudiante debe introducir un nombre para la entidad que se crea, además de las llaves primarias y los atributos de la relación si esta los posee. Una vez creada la respuesta a un ejercicio determinado el sistema mostrará una evaluación cualitativa de la misma. El módulo dará la opción a los estudiantes de enviar la respuesta y de generar un modelo de datos. Al seleccionar enviar la solución obtenida puede ser sometida a evaluación por el profesor o enviada a un foro, donde se permitirá que se realicen comentarios sobre la misma, siempre y cuando el ejercicio haya estado asociado a dicho foro. El profesor luego de revisar la respuesta, puede emitir una evaluación y un criterio, para después enviarla al estudiante. Si se selecciona la opción generar modelo de datos, este podrá ser guardado o exportado para poder utilizarse en la creación de ejercicios de implementación.

2.1.2. Modelo conceptual

Un modelo conceptual es una representación de un sistema que utiliza conceptos e ideas para formar dicha representación. El modelado conceptual se utiliza en muchos campos, desde las ciencias hasta la socioeconomía y el desarrollo de software (Powell-Morse, 2017).

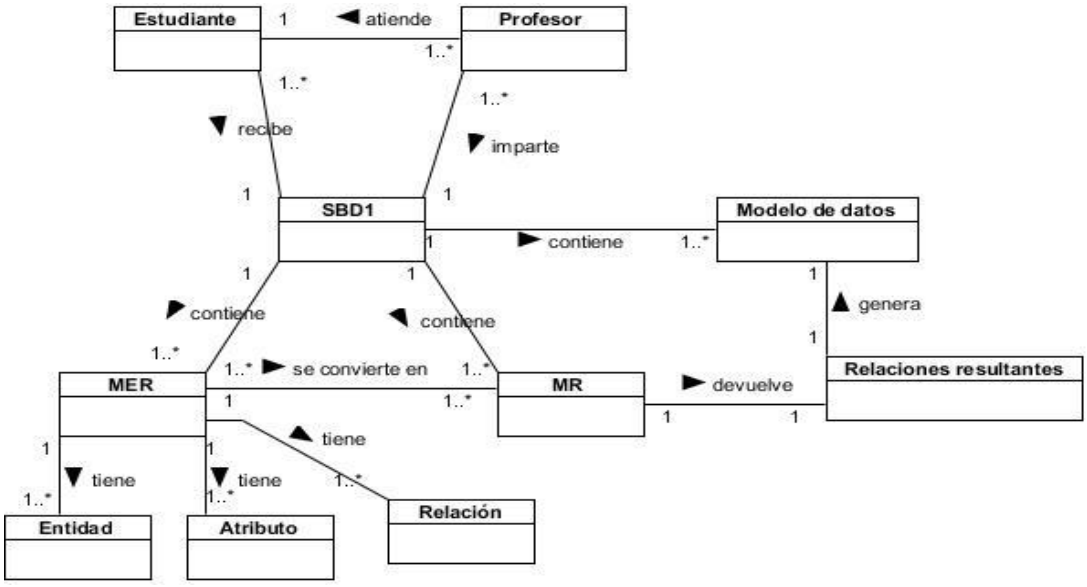


Figura 1 Relación entre conceptos asociados a la propuesta de solución

En el Modelo Conceptual que se muestra en la ilustración 11, los principales actores que intervienen en el proceso son el **estudiante** y el **profesor**, donde un profesor puede atender a muchos estudiantes pero un

estudiante es atendido por un solo profesor. La asignatura **Sistema de Bases de Datos I (SBDI)** puede ser impartida por varios profesores y esta puede ser cursada por varios estudiantes. Uno de los contenidos que se imparten en esta asignatura es el diseño de **Modelo Entidad Relación (MER)**, este está compuesto por varios elementos (**entidades**, **atributos** y **relaciones**). También en esta asignatura se realiza el **Modelo Relacional (MR)** a través de la transformación del MER, este MR está compuesto por las **relaciones resultantes**, a partir de estas relaciones se genera el **Modelo de datos**, que también es utilizado en la asignatura.

2.2. Diseño de la propuesta de solución

Modelar el sistema posibilita reflejar las propiedades o restricciones que este debe cumplir, a fin de satisfacer las necesidades del cliente y los usuarios finales (Labrada, y otros, 2013). Para la identificación de dichas necesidades se emplean técnicas que permiten recolectar la información necesaria para la obtención de los requisitos de una aplicación e investigar aspectos generales para posteriormente ser especificados con un mayor detalle(Troya, 2010)**.**

Como resultado de esta fase se presentan los requisitos funcionales y no funcionales del módulo, además de los respectivos artefactos generados, a partir de la metodología seleccionada.

2.2.1. Requisitos Funcionales

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

En un encuentro con el cliente, donde se aplicó una entrevistapara obtener información cualitativa como opiniones y descripciones subjetivas (Onofre Cortez, 2017) sobre cómo se lleva a cabo el proceso de enseñanza aprendizaje entorno al MR (ver Anexo 1) se obtuvo un total de veintiocho (28) requisitos funcionales. A estos 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 2: Requisitos funcionales

Código	Requisito Funcional	Descripción	Prioridad
RF1	Autenticar usuario	El sistema debe permitir autenticar usuario	Alta
RF2	Asignar roles y permisos a usuarios	El sistema debe permitir asignar roles y permisos a usuarios	Alta
RF3	Crear ejercicio de transformación	El sistema debe permitir crear ejercicio de transformación	Alta
RF4	Modificar ejercicio de transformación	El sistema debe permitir modificar ejercicio de transformación	Baja
RF5	Eliminar ejercicio de transformación	El sistema debe permitir eliminar ejercicios de transformación	Media
RF6	Mostrar detalles de ejercicios de transformación.	El sistema debe permitir mostrar detalles de ejercicios de transformación	Media
RF7	Listar ejercicio de transformación	El sistema debe permitir listar ejercicios de transformación	Alta
RF8	Transformar entidades	El sistema debe permitir transformar entidades	Alta
RF9	Transformar relaciones	El sistema debe permitir transformar relaciones	Alta
RF10	Validar transformación	El sistema debe permitir validar transformación	Alta
RF11	Crear respuesta de ejercicio de transformación	El sistema debe permitir crear respuesta de ejercicio de transformación	Alta
RF12	Mostrar reglas de transformación	El sistema debe permitir mostrar las reglas de transformación	Alta
RF13	Modificar respuesta de ejercicio de transformación	El sistema debe permitir modificar respuesta de ejercicio de transformación	Baja
RF14	Eliminar respuesta de ejercicio de transformación	El sistema debe permitir eliminar respuesta de ejercicio de transformación	Media
RF15	Mostrar detalles de la respuesta de ejercicios de transformación	El sistema debe permitir mostrar detalles de la respuesta de ejercicios de transformación	Media
RF16	Listar respuestas de ejercicios de transformación	El sistema debe permitir listar respuestas de ejercicios de transformación	Alta
RF17	Mostrar evaluación cualitativa de la respuesta del ejercicio de transformación	El sistema debe permitir mostrar evaluación cualitativa de la respuesta del ejercicio de transformación	Alta
RF18	Enviar respuesta de ejercicio de transformación al profesor	El sistema debe permitir enviar respuesta de ejercicio de transformación	Alta
RF19	Publicar respuesta de ejercicios de transformación en el foro.	El sistema debe permitir publicar respuesta de ejercicio de transformación	Media
RF20	Crear comentarios de respuestas de ejercicios de transformación publicadas en	El sistema debe permitir crear comentario de respuestas de ejercicios de transformación	Alta

	un foro	publicadas en un foro	
RF21	Eliminar comentarios de respuestas de ejercicios de transformación publicadas en un foro	El sistema debe permitir eliminar comentario de respuestas de ejercicios de transformación publicadas en un foro	Media
RF22	Modificar comentarios de respuestas de ejercicios de transformación publicadas en un foro	El sistema debe permitir modificar comentario de respuestas de ejercicios de transformación publicadas en un foro	Baja
RF23	Mostrar comentarios de respuestas de ejercicios de transformación publicadas en un foro	El sistema debe permitir mostrar comentario de respuestas de ejercicios de transformación publicadas en un foro	Media
RF24	Evaluar respuesta de ejercicios de transformación	El sistema debe permitir evaluar respuestas de ejercicios de trasformación	Media
RF25	Generar modelo de datos	El sistema debe permitir generar modelo de datos	Alta
RF26	Guardar modelo de datos	El sistema debe permitir guardar modelo de datos	Media
RF27	Exportar modelo de datos	El sistema debe permitir exportar modelo de datos	Media
RF28	Mostrar modelo de datos	El sistema debe permitir mostrar modelo de datos	Media

2.2.2. Requisitos No Funcionales

Estos requisitos se consideran 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).

Al ser un módulo para la plataforma *RDB-Learning* se asumirán los requisitos no funcionales establecidos para la misma. De esta forma se garantiza una homogeneidad en todos los módulos de esta herramienta, logrando que los usuarios sean capaces de usar todas las funcionalidades sin cambios drásticos.

Los dieciséis (16) requisitos no funcionales (RnF) se muestran agrupados en usabilidad, confiabilidad, portabilidad, eficiencia, funcionalidad y mantenibilidad, según lo establece el estándar de calidad internacional ISO/IEC 9126-1 (Rojo, y otros, 2012):

Usabilidad

RnF 1: El módulo de transformación del Diagrama Entidad Relación al Modelo Relacional en la plataforma *RDB-Learning* deberá ser una aplicación web

RnF 2: El módulo deberá contar con una ayuda para los usuarios, donde estarán todas las reglas de transformación

RnF 3: El módulo debe presentar una interfaz agradable e intuitiva para el usuario

RnF 4: Se deben seguir las pautas de la estrategia marcaria de la UCI para aplicaciones en el ámbito formativo

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 a los distintos navegadores y dispositivos

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

2.2.3. Descripción de requisitos de software

Un diagrama de Caso de Uso del Sistema es un artefacto narrativo que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Estas descripciones son independientes de la implementación y se establecen en un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema (ISO 9000, 2015).

A continuación se muestra el Diagrama de Caso de Uso del Sistema, donde se definieron como actores del sistema al Estudiante y el Profesor y fueron agrupados los requisitos en Casos de Usos:

Figura 2. Diagrama de Caso de Uso del Sistema

Para una mejor comprensión del funcionamiento de los requisitos se emplea la descripción de casos de usos donde se usa la técnica de Desarrollo de prototiposcon el fin de tener una visión preliminar del

módulo que se implantará. Los prototipos básicos se diseñaron durante la fase de determinación de los requerimientos (Ruiz, 2016).

A continuación, se describen los casos de uso Transformar Entidad y Transformar Relación, asociados los RF8 y RF9 respectivamente:

Tabla 3: Descripción del Caso de Uso Transformar entidad

Caso de Uso	Transformar entidad	
Objetivo	Permite transformar la entidad deseada por el estudiante.	
Actores	Estudiante: (Inicia) introduce los datos solicitados y transforma la entidad.	
Resumen	El caso de uso se inicia cuando el estudiante decide transformar la entidad para solucionar un ejercicio.	
Precondiciones	El estudiante tiene que estar autenticado.	
Postcondiciones	Que la entidad haya sido transformada.	
Flujo de eventos		
Flujo básico transformar entidad		
Actor		**Sistema**
1.	Selecciona la opción de resolver un ejercicio de transformación de la lista de ejercicios.	
2.		Muestra una pantalla con dos opciones, una que permite transformar entidad y la otra para transformar las relaciones entre ellas.
3.	Selecciona la opción transformar entidad.	
4.		Solicita el nombre de la entidad, la llave primaria y los atributos.
5.	Introduce los datos solicitados (nombre de la entidad, llave primaria, atributos).	
6.		Valida que la entidad exista en el MER
7.	Selecciona la opción Aceptar.	
8.		Termina el caso de uso.
Prototipo elemental de interfaz gráfica		

<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td colspan="3">Flujos alternos</td></tr>
<tr><td colspan="3">No1 entidad no existe</td></tr>
<tr><td colspan="2">Actor</td><td>Sistema</td></tr>
<tr><td>1.</td><td></td><td>Muestra el mensaje “La entidad no existe”.</td></tr>
</table>

Tabla4 Descripción del Caso de Uso Transformar relaciones

<table>
<tr><td>Caso de Uso</td><td colspan="3">Transformar relaciones</td></tr>
<tr><td>Objetivo</td><td colspan="3">Permite transformar la relación deseada por el estudiante.</td></tr>
<tr><td>Actores</td><td colspan="3">Estudiante: (Inicia) introduce los datos solicitados y transforma la relación entre entidades.</td></tr>
<tr><td>Resumen</td><td colspan="3">El caso de uso se inicia cuando el estudiante decide transformar la relación entre entidades para solucionar un ejercicio.</td></tr>
<tr><td>Precondiciones</td><td colspan="3">El estudiante tiene que estar autenticado.</td></tr>
<tr><td>Postcondiciones</td><td colspan="3">Que la relación haya sido transformada.</td></tr>
<tr><td colspan="4">Flujo de eventos</td></tr>
<tr><td colspan="4">Flujo básico transformar relaciones</td></tr>
<tr><td colspan="2">Actor</td><td colspan="2">Sistema</td></tr>
<tr><td>9.</td><td>Selecciona la opción de resolver un ejercicio de transformación de la lista de ejercicios.</td><td colspan="2"></td></tr>
<tr><td>10.</td><td></td><td colspan="2">Muestra una pantalla con dos opciones, una que permite transformar entidad y la otra para transformar las relaciones entre ellas.</td></tr>
<tr><td>11.</td><td>Selecciona la opción transformar relaciones.</td><td colspan="2"></td></tr>
<tr><td>12.</td><td></td><td colspan="2">Permite escoger el tipo de relación y las</td></tr>
</table>

		entidades relacionadas.
13.	Selecciona los datos solicitados (tipo de relación, entidades relacionadas).	
14.		Permite seleccionar que atributo pasa a ser llave foránea
15.	Selecciona la llave foránea	
16.		Muestra la relación resultante.
17.	Selecciona la opción Aceptar.	
18.		Termina el caso de uso.
Prototipo elemental de interfaz gráfica		
		

2.2.4. Diagrama de Clases del Diseño

Un Diagrama de Clases del Diseño es una representación gráfica que sirve para representar la estructura de un sistema que será implementado utilizando un lenguaje orientado a objetos. Los diagramas de clases se realizan en la fase de diseño del software. La idea de estos diagramas es representar las clases que tendrá el sistema así como su contenido y sus relaciones con otras clases (Salas, 2016).

A continuación se muestran los diagramas de Clases del Diseño de los Casos de Uso Transformar entidad y Transformar relaciones, como representación gráfica de las descripciones anteriores.

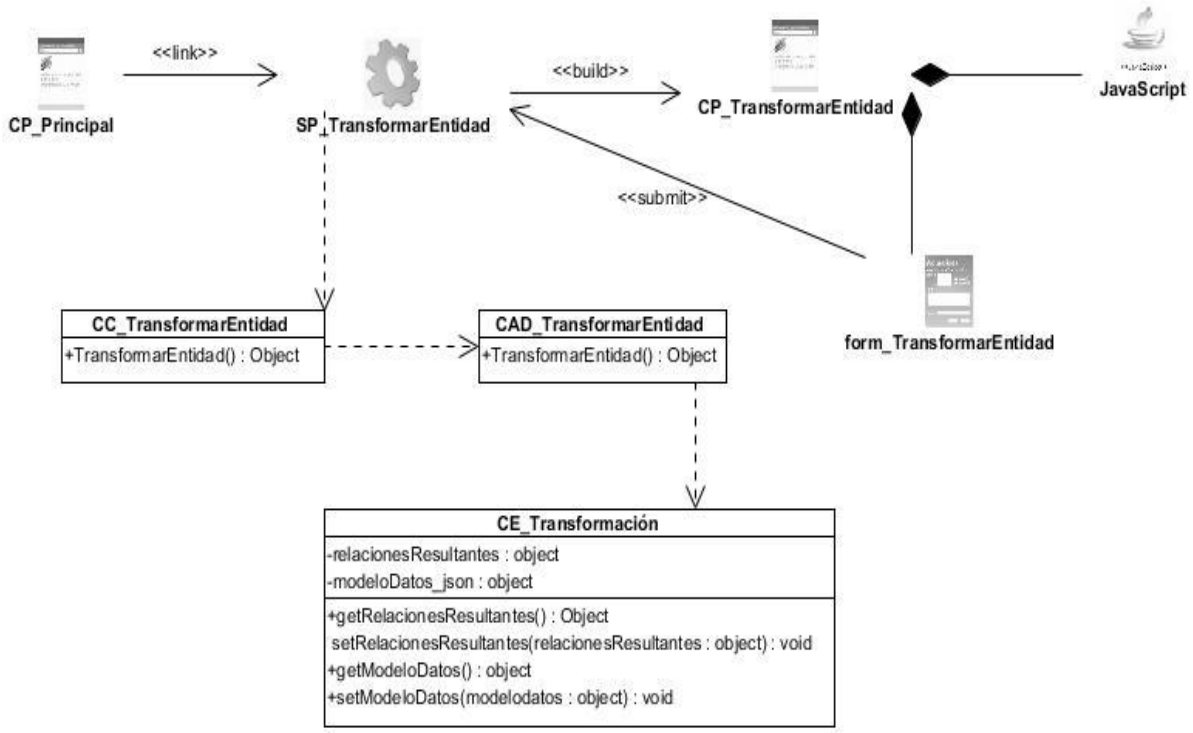


Figura 3. Diagrama de Clase del Caso de uso “Transformar entidad”

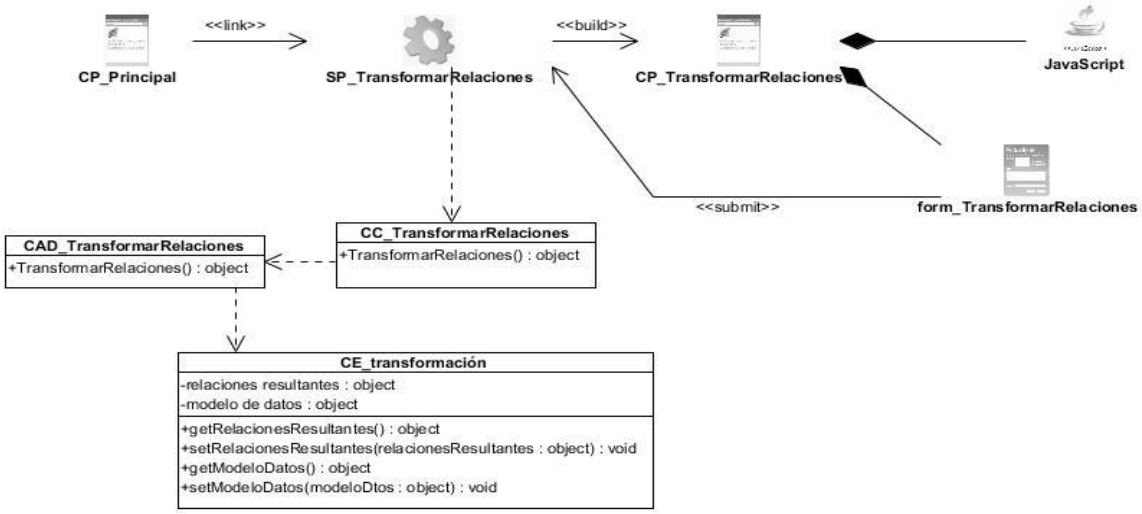


Figura 4. Diagrama de Clase del Caso de Uso “Transformar Relación”

2.2.5. Estilo Arquitectónico

En el desarrollo de la solución se utiliza el estilo arquitectónico MVT (*Model-View-Template*, Modelo-Vista-Plantilla) utilizado por el marco de trabajo Django. En este marco, el Modelo es la capa de acceso a la base de datos y contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene y las relaciones entre los datos. La Vista es la capa de la lógica de negocios incluye la lógica que accede al modelo y la delega a la plantilla apropiada. Es una conexión entre los modelos y las plantillas. La Plantilla es la capa de presentación y comprende las decisiones relacionadas a la presentación: como son mostradas sobre una página web u otro tipo de documento (LibroDjango, 2017).

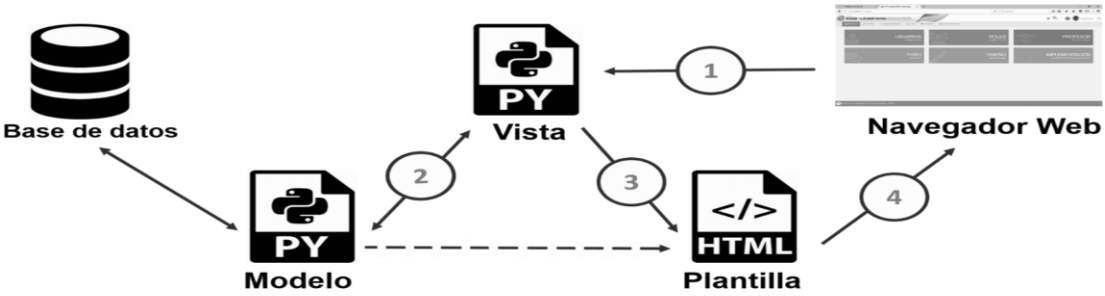


Figura 5. Flujo de trabajo del estilo arquitectónico Modelo-Vista-Plantilla

A continuación se explica el funcionamiento de esta arquitectura, el cual se puede observar en la Figura 5:

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

El Modelo: define los datos almacenados, es representado en forma de clases en 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 con algunas etiquetas extras que son propias de Django.

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 (Estructured Query Language) para hacer las consultas.

2.2.6. Patrones del diseño

Es una solución general reusable que puede ser aplicada a problemas que ocurren comúnmente en el desarrollo de software, es la descripción o plantilla de cómo resolver un problema que puede ser usada en diferentes situaciones. Los patrones de diseño son soluciones probadas, expresivas y fáciles de mantener. Muchos desarrolladores están familiarizados con los patrones de diseño, así que se puede decir que es un tipo de estándar de desarrollo (Amaya, 2016). A continuación, se presentan los patrones utilizados en el desarrollo del módulo en la plataforma *RDB-Learning.*

(*GRASP*): Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos y sirven para asignar correctamente las responsabilidades en el diseño de POO. Los patrones GRASP son un método de enseñanza que ayuda a entender el Diseño de POO y aplica el análisis y diseño de objetos de un modo sistemático, racional y explicable (Uquizo, 2016).

Experto: es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la cosa real que representa, por lo que ofrece una analogía con el mundo real. Brinda soporte a una alta cohesión (Uquizo, 2016).

En el módulo este patrón se utiliza, por ejemplo para determinar los estudiantes que han resuelto un problema determinado. La clase Respuesta y Transformación son las expertas en la información relacionada con los ejercicios de transformación. A continuación, se muestran las clases donde se evidencia dicho patrón.

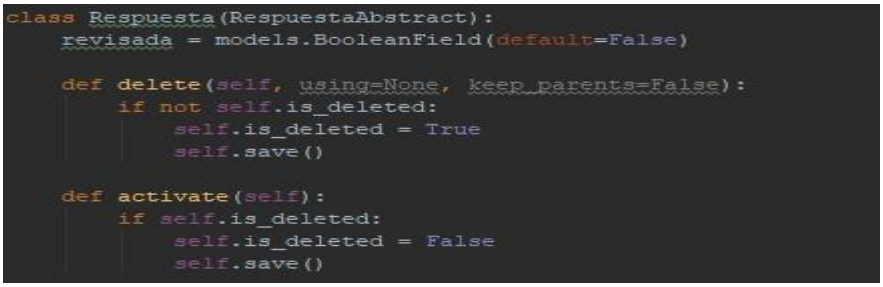


Figura 6. Ejemplo del patrón Experto

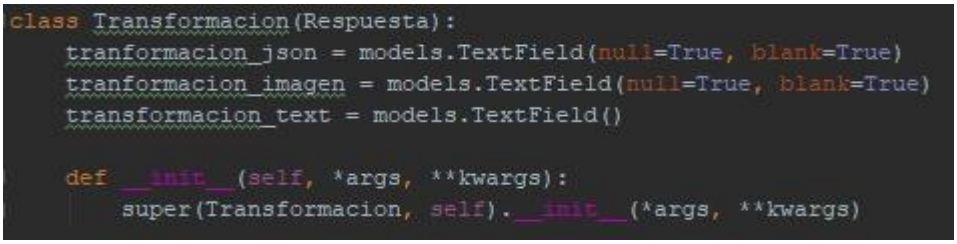


Figura 7. Ejemplo del patrón Experto 2

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Brinda un soporte a Bajo Acoplamiento (Uquizo, 2016).

Este patrón se pone de manifiesto en la implementación de la función GuardarRespuesta que se encarga de guardar las respuestas a un determinado ejercicio cuyo identificador coincide con uno pasado por parámetro, se crea una nueva instancia de la entidad Respuesta, para el ejercicio a resolver. A continuación, se muestra la clase donde se evidencia dicho patrón.

```
def GuardarRespuesta(request, id_ejercicio):
    ejercicio = Ejercicio.objects.get(pk=id_ejercicio)
    if request.is_ajax() and request.POST:
        if ejercicio.tipo_ejercicio == 'disenno':
            diagrama_mer_imagen = request.POST['diagrama_mer_imagen']
            diagrama_mer_json = request.POST['diagrama_mer_json']

            respuesta, created = Transformacion.objects.get_or_create(ejercicio_id=id_ejercicio, resuelto_por=request.user)
            respuesta.diagrama_mer_json = diagrama_mer_json
            respuesta.diagrama_mer_imagen = diagrama_mer_imagen
        else:
            sql_code = request.POST['sql_code']
            respuesta, created = Implementacion.objects.get_or_create(ejercicio_id=id_ejercicio, resuelto_por=request.user)
            respuesta.codigo_sql = sql_code
            if 'diagrama_json' and 'operacion_json' in request.POST:
                diagrama_json = request.POST['diagrama_json']
                operacion_json = request.POST['operacion_json']
                respuesta.diagrama_json = diagrama_json
                respuesta.operacion_json = operacion_json
        respuesta.tipo_respuesta = TIPO_RESPUESTA.GUARDADA
        respuesta.estado = ESTADO_RESPUESTAS.PENDIENTE
        respuesta.save()
```

Figura 8. Ejemplo del patrón Creador

Bajo acoplamiento: es un principio que se debe tener siempre en cuenta durante las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades (Uquizo, 2016).

Este patrón ya viene incluido con Django 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, en la base datos y las plantillas HTML, basta solo con realizarlo una vez. A continuación, se muestran la clase donde se evidencia dicho patrón.

```
url(r'^transformacion/(?P<id_ejercicio>\d+)/nueva$', CrearRespuestaTransformacion, name='transformacion_new'),
url(r'^transformacion/(?P<id_respuesta>\d+)/editar$', EditarRespuestaTransformacion, name='transformacion-edit'),
```

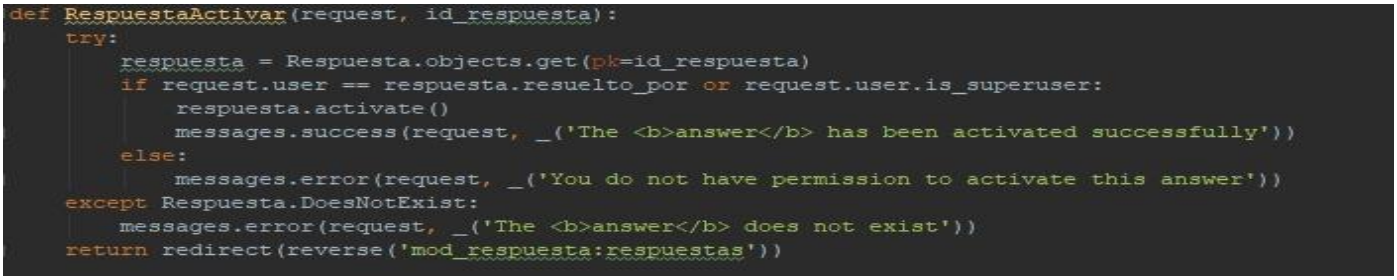
Figura 9. Ejemplo del patrón Bajo Acoplamiento

Alta Cohesión: es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande (Uquizo, 2016).

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 sistema 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: la mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. Otros medios de entrada son los mensajes externos o las señales procedentes de sensores como sucede en los sistemas de control de procesos. En todos los casos, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. La misma clase controlador debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso (Uquizo, 2016).

Este patrón es empleado en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase o función controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión. A continuación, se muestran la clase donde se evidencia dicho patrón.



```
def RespuestaActivar(request, id_respuesta):
    try:
        respuesta = Respuesta.objects.get(pk=id_respuesta)
        if request.user == respuesta.resuelto_por or request.user.is_superuser:
            respuesta.activate()
            messages.success(request, _('The <b>answer</b> has been activated successfully'))
        else:
            messages.error(request, _('You do not have permission to activate this answer'))
    except Respuesta.DoesNotExist:
        messages.error(request, _('The <b>answer</b> does not exist'))
    return redirect(reverse('mod_respuesta:respuestas'))
```

Figura 10. Ejemplo del patrón Controlador

Los GOF (The Gang of Four, Patrulla de los Cuatro) : son patrones que describen soluciones simples a problemas específicos en el diseño de software orientado a objetos, definen los patrones de diseño como combinaciones de componentes, casi siempre clases y objetos que por experiencia se sabe que resuelven ciertos problemas de diseño comunes (Información Tecnológica, 2015).

Decorator (Decorador): Permite agregar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. En la plataforma se emplean los decoradores *user_active_required* y *user_admin_required.*

2.2.7. Modelo de Datos.

Un modelo de datos es un conjunto de herramientas conceptuales para la descripción de los datos y las relaciones entre ellos, su semántica y las restricciones de consistencia (Sánchez, 2016).El modelo de datos que se muestra es el generado en la creación de la plataforma, se le adicionó la entidad transformación. A continuación se muestra y se explica este modelo:

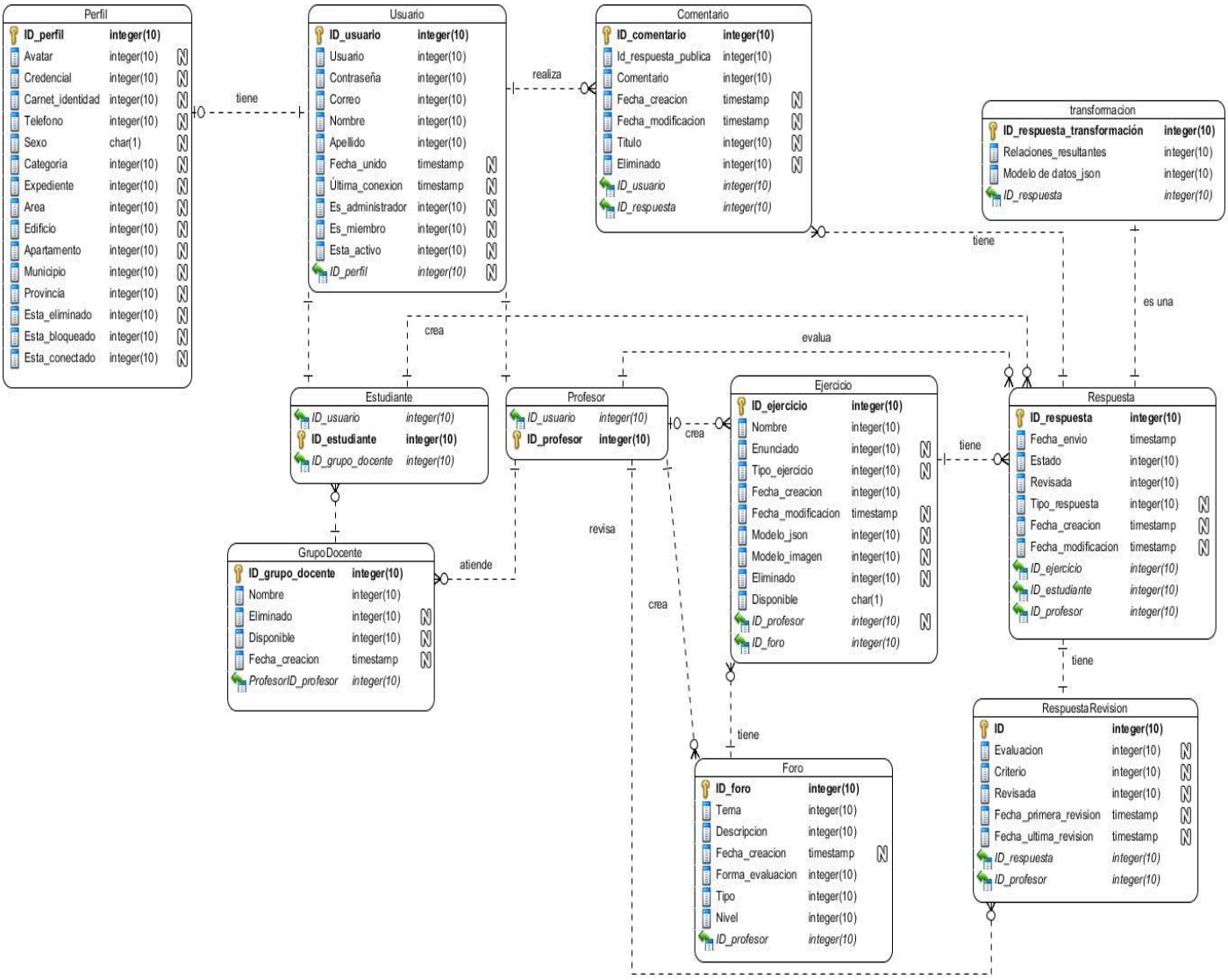


Figura 11. Modelo de datos para la solución propuesta

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.

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_profesor*) 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*) y el profesor que la evalúa.

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.

Transformación: esta entidad, hereda de la entidad respuesta, se conoce su identificador (*id_respuesta_transformación)* y se almacena además las relaciones resultantes y el modelo de datos en formato *JSON*.

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 asocien a él. Se conoce además el profesor que crea el foro (*id_profesor*).

Conclusiones del capítulo

Con el estudio de los temas referentes al análisis y diseño de la propuesta del módulo de transformación se puede arribar a las siguientes conclusiones:

- El análisis de las características del módulo y la modelación del negocio, permiten identificar los principales requisitos funcionales y no funcionales los cuales fueron agrupados y descritos por Casos de uso.
- La definición de los patrones de diseño y el estilo arquitectónico empleado, evidencia que la solución propuesta puede ser adaptable a modificaciones.
- Los diagramas de clases, secuencias y paquete, facilitan una visión en cuanto a la composición lógica y física del módulo.
- La generación de los artefactos requeridos por la metodología selecciona, contribuyen a documentar la solución propuesta, lo cual facilita su posterior mantenimiento.

CAPÍTULO 3: Implementación y prueba del módulo de transformación del Modelo Entidad-Relación al Modelo Relacional para la plataforma *RDB-Learning*.

En el presente capítulo se caracterizan los elementos definidos en el proceso de desarrollo, para obtener un producto de calidad. Es presentado y descrito el diagrama de componentes para una mayor comprensión de los elementos físicos del sistema y sus relaciones.

En esta sección se incluye la especificación del estándar de codificación desarrollado para la implementación de la aplicación. Además, se plasman los resultados de las pruebas realizadas al módulo con el objetivo de comprobar su correcto funcionamiento y validar la propuesta de solución.

3.1. Diagrama de despliegue

Un diagrama de despliegue representa la relación física que se establece entre los distintos componentes o nodos que describen la topología de un sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de *hardware* y la información relacionada con la forma en la que los componentes se comunicarán a lo largo de la infraestructura del sistema. También se utiliza para visualizar la distribución de los componentes de *software* en los nodos físicos. El mismo está compuesto por: nodos, dispositivos y conectores (Troya, 2010).

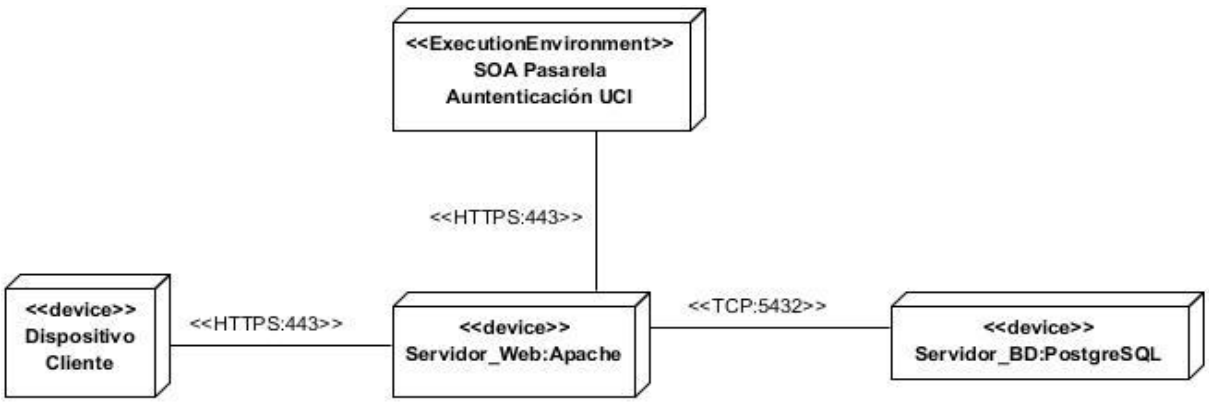


Figura 12. Diagrama de despliegue

PC Cliente: Se refiere a las estaciones de trabajo que realizan las peticiones al servidor de aplicaciones donde está hospedada la plataforma que soporta el módulo de transformación del Modelo Entidad-

Relación al Modelo Relacional para la plataforma *RDB-Learning*, mediante un navegador web utilizando el protocolo de comunicación *HTTPS* por el puerto 443.

Servidor Web: Es el encargado de brindar la interfaz de la plataforma para que los usuarios puedan hacer uso de esta, almacena todo el código fuente del sistema y se comunica por medio de los protocolos *TCP* con el servidor de bases de datos.

Sistema Gestor de Bases de Datos: Almacena toda la información que brinda la plataforma hospedada en el servidor de aplicaciones. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través del protocolo *TCP* empleando el puerto 5432.

Servidor web con el servidor *SOA*, necesario para la obtención de la pasarela de autenticación UCI, que garantiza la identificación y acceso de los usuarios al sistema. A través del protocolo HTTPS con el puerto 443.

3.2. Diagrama de componentes

El diagrama de componentes permite visualizar la estructura de un sistema informático mediante el análisis de las partes que lo conforman. Los componentes son utilizados como una unidad de composición independiente e indispensable dentro de un sistema, donde se identifica las dependencias que existen entre cada uno de los elementos. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables y binarios (Sommerville,2011).

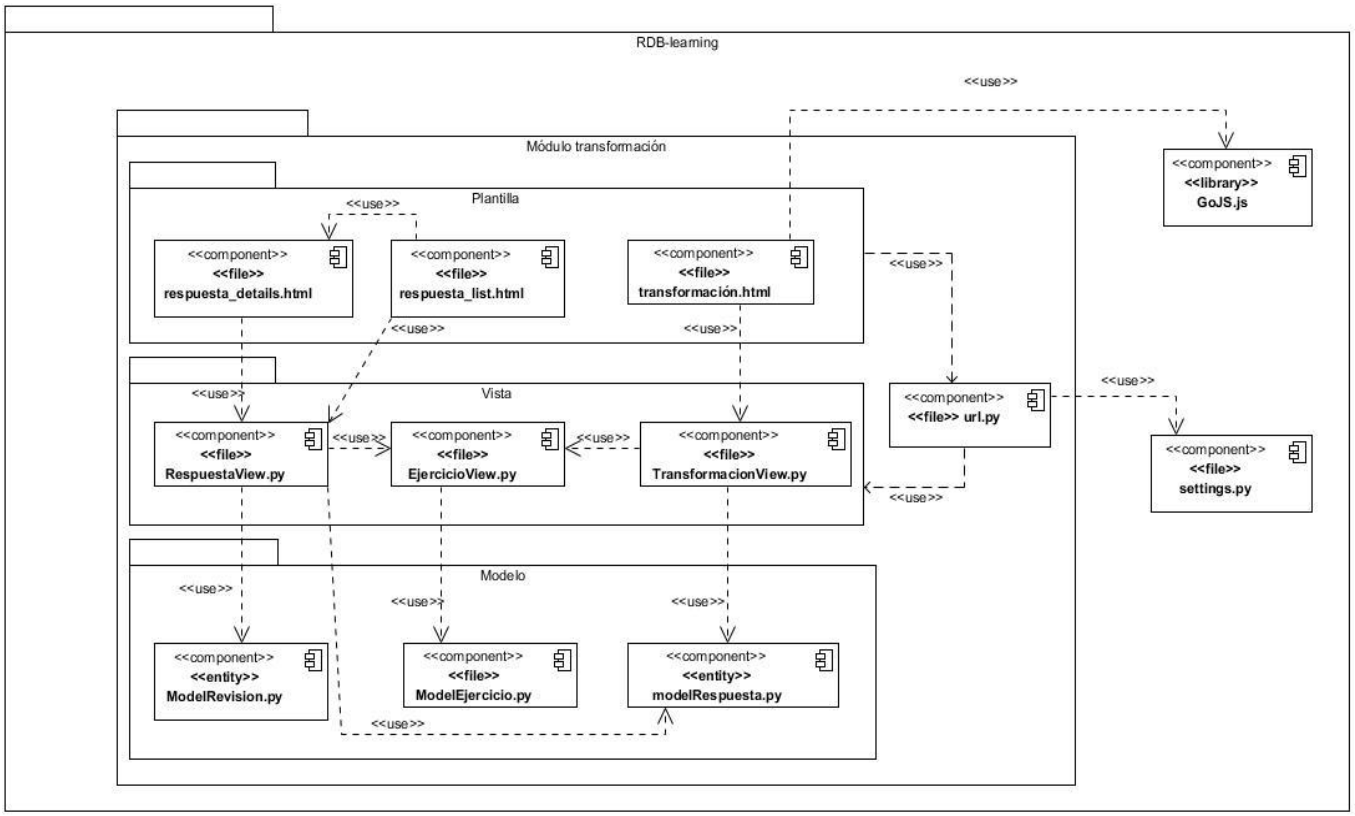


Figura 13. Diagrama de componentes

A continuación, se describen los elementos que componen el diagrama mostrado, según su ubicación dentro del estilo arquitectónico empleado.

Tabla 5. Descripción de los elementos del Diagrama de Componentes

Componentes		**Descripción**
Plantillas	transformación.html	Este componente es el que muestra el área de trabajo para la transformación. Utiliza la librería *GoJS* para generar los modelos de datos.
	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

	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)
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
	TransformaciónViews.py	Este componente se encarga de listar los ejercicios de transformación en el sistema, así como crear y editar las respuestas creadas
Modelo	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

Los estándares de codificación son un conjunto de reglas o patrones de codificación a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del sistema en desarrollo. Estos estándares cumplen con el principio de legibilidad y mantenibilidad, correspondientes al objetivo de que el programador entienda el código y sea capaz de modificarlo (Microsoft, 2016). Para la codificación de la propuesta de solución se utiliza el estándar de codificación para Python:

- Indentación: se utilizaron 4 espacios por cada nivel de indentación.
- Tabuladores o espacios: no se mezclaron tabuladores y espacios en la codificación. Las formas más populares deindentar en Python son utilizando solo espacios o solo tabuladores, el código indentado con una mezcla de tabuladores y espacios se reformateó y se usaron espacios exclusivamente.
- **Tamaño máximo de línea**: se limitaron todas las líneas a un máximo de 79 caracteres.

- **Convenciones de nombres**: no se utilizaron los caracteres 'l' (letra ele minúscula), 'O' (letra o mayúscula) o 'I' (letra i mayúscula) como nombres de variables de un solo caracter debido a que en algunas fuentes, estos caracteres son indistinguibles de los numerales uno y cero.
  Se utilizó CapWords (palabras que comienzan con mayúsculas) para los nombres de las clases. Los nombres de funciones están en letras minúsculas, con palabras separadas mediante guiones bajos según sea necesario para mejorar la legibilidad.
- **Otras consideraciones**: se rodearon siempre los siguientes operadores binarios con un espacio en cada lado: asignación (=), asignación aumentada (+=, -=,*=, /=), comparación (==, <, >!,=, <>, <=, >=, in, not in, is, isnot), booleanos (and, or, not).
  Se utilizaron espacios alrededor de los operadores aritméticos.

A continuación, se muestra la Figura 11 que evidencia la aplicación de los estándares de codificación.

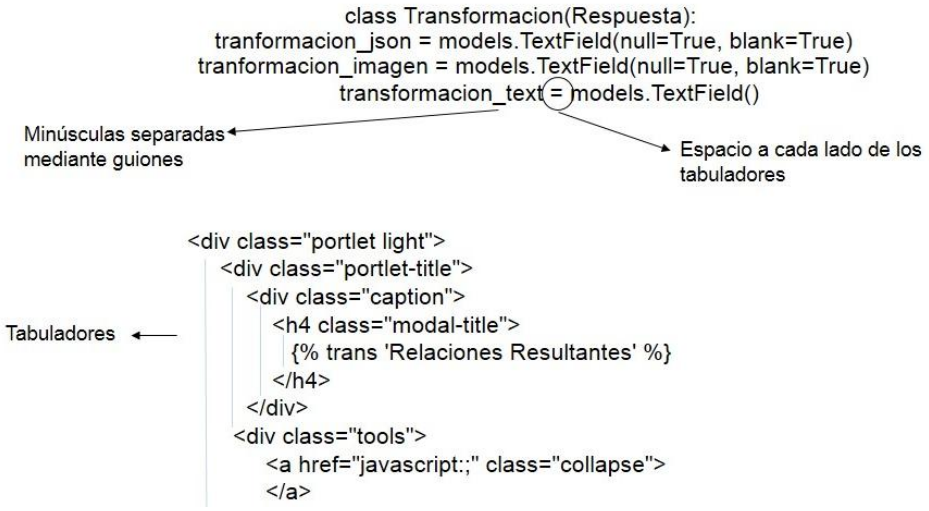


Figura 14. Uso de los estándares dentro de la solución

3.4. Estrategia de pruebas

Las pruebas de software son el proceso que consiste en todas las actividades del ciclo de vida, tanto estáticas como dinámicas relacionadas con la planificación, preparación y evaluación de productos de software y productos relacionados con el trabajo para determinar que cumplen los requisitos especificados, para demostrar que son aptos para el propósito y para detectar defectos (Sánchez, 2015).

3.4.1. Pruebas unitarias

Estas pruebas son una forma de comprobar que un fragmento de código funciona correctamente. Es un procedimiento más de los que se llevan a cabo dentro de una metodología ágil de trabajo. Las pruebas unitarias realizadas utilizan el método de caja blanca y la técnica del camino básico. El método del camino básico permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño procedural y utilizar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (Picurelli, y otros, 2019). A continuación, se realiza la técnica de camino básico al método que posibilita la eliminación de forma adecuada de la respuesta.

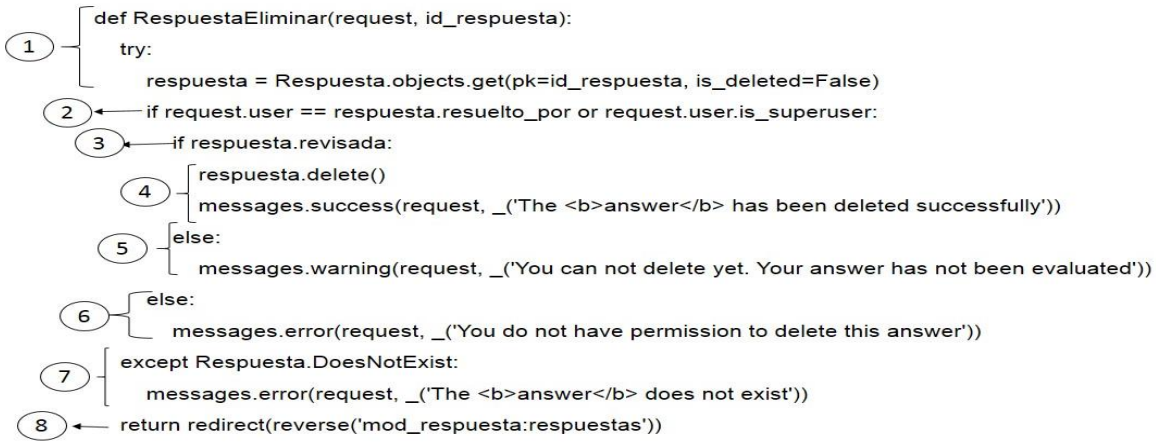


Figura 15. Técnica de Camino mínimo sobre el método EliminarRespuesta

Luego de enumerar el código, se diseña la gráfica del método en la Figura 12, que describe el flujo de control lógico mediante la utilización de nodos y aristas.

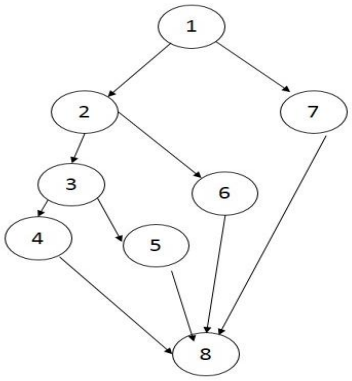


Figura 16. Grafo del Flujo para el método eliminar respuesta

A partir del grafo obtenido con 8 nodos y 10 aristas se calcula la complejidad ciclomática (V (G)). La complejidad ciclomática se basa en el recuento del número de caminos lógicos individuales contenidos en un programa. Para hallar la complejidad ciclomática, el programa se representa como un grafo, y cada instrucción que contiene, un nodo del grafo. Las posibles vías de ejecución a partir de una instrucción (nodo) se representan en el grafo como aristas.

V (G) = cantidad_aristas – cantidad_nodos + 2

V (G)=10-8+2=4

La cantidad de rutas independientes se establecen por la complejidad ciclomática; fueron identificadas 4 rutas, tal y como se muestra en la tabla siguiente:

Tabla 6. Lista de Caminos

No. Ruta	Camino
1	1-2-3-4-8
2	1-2-3-5-8
3	1-2-6-8
4	1-7-8

A continuación, se diseñan casos de pruebas para ser aplicados a algunas de las rutas independientes:

Tabla 7. Caso de prueba unitaria. Ruta 1

Caso de Prueba de Unidad

No. Ruta: 1	**Ruta:** 1-2-3-4-8
Descripción de la prueba: La respuesta se ha eliminado satisfactoriamente.	
Entrada: Una respuesta de un ejercicio	
Resultado esperado: Que la respuestadel ejercicio haya sido revisada.	
Evaluación de la prueba: Satisfactorio. Se elimina la respuesta del ejercicio.	

Tabla 8. Caso de prueba unitaria. Ruta 2

Caso de Prueba de Unidad	
No. Ruta: 2	**Ruta:** 1-2-3-5-8
Descripción de la prueba: La respuesta se ha eliminado satisfactoriamente.	
Entrada: Una respuesta de un ejercicio	
Resultado esperado: Que la respuestadel ejercicio no haya sido revisada.	
Evaluación de la prueba: Satisfactorio. No se puede eliminar la respuesta del ejercicio porque no ha sido revisada.	

Tabla 9Caso de prueba unitaria. Ruta 3.

Caso de Prueba de Unidad	
No. Ruta: 3	**Ruta:** 1-2-6-8
Descripción de la prueba: La respuesta se ha eliminado satisfactoriamente.	
Entrada: Una respuesta de un ejercicio	
Resultado esperado: Que la respuestadel ejercicio este vacía, es decir que no esté resuelto.	
Evaluación de la prueba: Satisfactorio. No se puede eliminar la respuesta del ejercicio porque no ha sido resuelto.	

Como resultado de esta prueba en la primera iteración fueron detectadas dos no conformidades relacionadas a errores de validación y al tratamiento de excepciones. Las deficiencias fueron corregidas mediante una revisión detalla al código y al correcto lanzamiento de excepciones. Fue realizada una segunda iteración donde no se evidenciaron no conformidades.

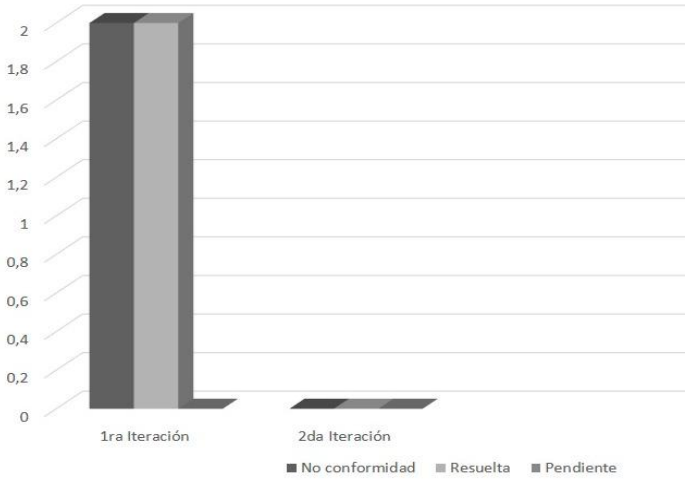


Figura 17. Resultado de las pruebas unitarias

3.4.2. Pruebas funcionales

Las pruebas funcionales se definen teniendo como fuente los requisitos del sistema, estas pruebas validan y verifican que el producto cumple con lo especificado, dando también una idea del grado de calidad del software. Este servicio ayuda a la organización a detectar los posibles defectos derivados de errores en la fase de programación (Talento, 2017).

Tabla 10. Caso de Prueba "Transformar Entidad"

Escenario	**Descripción**	**Entidad**	**llave**	**atributos**	**Respuesta del sistema**	**Flujo Central**
1.1 Transformar entidad insertando datos válidos	La entidad es transformada según los parámetros que introduce el estudiante	V persona	V CI	V estatura	El sistema almacena la entidad nombrada en la base de datos	Se analiza e identifica la entidad nombrada para ver si existe en el MER, luego es almacenada
1.2	La entidad es	I	V	V	El sistema muestra	Se analiza e

Transformar entidad insertando datos inválidos	transformada según los parámetros que introduce el estudiante	Casa	Número	dirección	un mensaje de error porque el nombre de la entidad no existe	identifica la entidad nombrada para ver si existe en el MER, luego es mostrado un mensaje

Tabla 11. Caso de prueba "Transformar Relación"

Escenario	**Descripción**	**Tipo**	**Entidad 1**	**Entidad 2**	**Respuesta del sistema**	**Flujo Central**
Transformar relaciones	Según el tipo de relación, esta es transformada	Uno-Mucho	Escuela	Municipio	El sistema transforma la relación entre las entidades escogidas.	Es procesado el tipo de relación y las entidades escogidas, para según el tipo solicitar al estudiante los parámetros necesarios para luego transformar esa relación

La realización de las pruebas funcionales posibilitó la detección y corrección temprana de errores en la aplicación. En las iteraciones realizadas fue encontrada una no conformidad, en el caso de prueba Transformar entidad, que transformaba entidades que no existían en el MER, este fue solucionado validando que el nombre de la entidad que introduce el estudiante exista en el modelo en caso de que no exista se muestra un mensaje.

3.4.3. Pruebas de carga y estrés.

Las pruebas de cargas tienen como objetivo saber cuánta carga puede soportar un sistema sin empeorar su rendimiento. Estas pruebas sirven para conocer la capacidad máxima de un sistema bajo una carga determinada y poder dar con las causas que puedan condicionar su rendimiento (Santiago Sanchez, 2019).

Las pruebas de estrés son realizadas sobrecargando un sistema más allá de sus especificaciones, para verificar cómo y cuándo fallará. Dentro de informática podemos colocar una gran carga en la base de

datos, entradas (peticiones) continuas al sistema o almacenar información más allá de la capacidad de memoria del sistema (Santiago Sanchez, 2019).

Se utilizó la herramienta Apache *JMeter* para la realización de estas pruebas, a continuación, se describen las variables que miden el resultado de las mismas, realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada *URL*
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
% 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 300, 700 y 1000 usuarios simultáneamente en cada caso, los cuales realizan hasta 5 peticiones por segundo. Se obtuvieron los siguientes resultados:

Tabla 12Resultados de la prueba carga y estrés.

Usuarios	**Muestras**	**Media**	**Mediana**	**Min**	**Max**	**%Error**	**Rend**	**Kb/s**
300	1000	1633	1120	96	4501	0.00%	89.14	189.9
700	2000	1262	1084	121	4992	0.10%	78.25	537.1
1000	3000	1212	1002	73	4755	1.5%	67.25	950.9

Las pruebas que fueron realizadas muestran que el módulo es capaz de responder a las peticiones, evidenciando así que este puede procesar la carga esperada. Para esto se realizaron un total de 300 peticiones de 1000 usuarios conectados simultáneamente en un tiempo promedio de 1633 milisegundos (1.6 segundos aproximadamente) con 0 % de error.

También se realizaron 2000 peticiones iniciadas por 700 usuarios y en este caso el módulo respondió en 1262 milisegundos (1.2 segundos aproximadamente) como tiempo promedio. Aunque el módulo no fue

capaz de responder correctamente el 0.10% de las peticiones realizadas, se demuestra que el módulo puede procesar la carga esperada.

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 3000 peticiones en un tiempo promedio de 1212 milisegundos (1.2 segundos aproximadamente), pero con un porciento de error de 1.5. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado.

3.4.4 Pruebas de aceptación

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido(Sánchez, 2015)

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, sesolicitó la colaboración de 4 estudiantes aventajados en la asignatura. El instrumento utilizado es una encuesta (ver Anexo2) a través del cual se pueda validar el nivel de aceptación de la plataforma y el módulo en los usuarios.

Conclusiones del capítulo

En este capítulo se abordaron aspectos correspondientes a la implementación y validación del componente de categorización semántica de documentos, arribándose a las siguientes conclusiones:

- La confección del diagrama de componentes permite comprender la integración de los componentes qe forman parte del módulo desarollado.
- La utilización de los estándares de codificación de código permitió adoptar una estructura homogénea que facilita la comunicación y asegura la calidad, menos errores y fácil mantenimiento.
- La aplicación de pruebas unitarias, funcionales y de carga y estrés arrojaron como resultados las principales deficiencias en el desarrollo del módulo, que una vez corregidas permite la obtención de una aplicación funcional y completamente operativa.

CONCLUSIONES GENERALES

La presente investigación tuvo como base el desarrollo de un módulo de transformación del diagrama entidad relación al relacional para la plataforma *RDB-Learning*, para ello se dio cumplimiento a una serie de objetivos específicos, los cuales fueron cumplidos satisfactoriamente, por lo que se puede arribar a las siguientes conclusiones:

-El estudio de los referentes teóricos y el análisis de herramientas existentes para la transformación de el diagrama entidad relación al relacional permitió determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.

- 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 la transformación del diagrama entidad relación al relacional en la plataforma *RDB-Learning*.

- El módulo de transformación del diagrama entidad relación al relacional constituye una solución funcional y con calidad, conforme a los resultados obtenidos de las pruebas unitarias, funcionales y de carga y estrés aplicadas.De forma general se concluye que el módulo contribuye a la autoevaluación de los estudiantes y la interacción con sus profesores en el proceso de enseñanza-aprendizaje de Sistema de Bases de Datos I.

RECOMENDACIONES

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

-Adicionar un módulo de corrección automática a la plataforma, que permita identificar errores en la representación del MER mientras se está diseñando.

-Aprovechando que es una herramienta web se podría incorporar un chat interno a la aplicación para discutir con otros usuarios, mientras se realiza el proceso de diseño de una base de datos.

-Ampliar las funcionalidades con contenidos de Sistema de Base de Datos II.

REFERENCIAS BIBLIOGRÁFICAS

Alegsa, Leandro. 2019. Alegsa. [En línea] 2019. [Citado el: 29 de noviembre de 2019.] http://www.alegsa.com.ar/Dic/servidor_de_base_de_datos.php.

Alegsa, Leandro. 2018. ALEGSA.com. [En línea] 2018. [Citado el: 25 de noviembre de 2019.] http://www.alegsa.com.ar/Dic/entidad.php.

Alegsa, Leandro. 2018. ALEGSA.com.ar. [En línea] 2018. [Citado el: 25 de noviembre de 2019.] http://www.alegsa.com.ar/Dic/atributo.php.

Alonso, Nacho. 2019. medium. [En línea] 2019. [Citado el: 29 de noviembre de 2019.] https://medium.com/@nachoad/que-es-wsgi-be7359c6e001.

Amaya, Janneth. 2016. Nearsoft. [En línea] 2016. [Citado el: 15 de enero de 2020.] https://blog.nearsoftjobs.com/qu%C3%A9-es-y-qu%C3%A9-no-es-un-patr%C3%B3n-de-dise%C3%B1o-487643d37a62.

Aragón, Yaniel Lázaro. 2018.*Herramienta para el aprendizaje de Bases de Datos Relacionales.* 2018.

Cadenas, Victor Garibay. 2016. medium. [En línea] 2016. [Citado el: 29 de noviembre de 2019.] https://medium.com/@victor.garibayy/qu%C3%A9-es-y-para-qu%C3%A9-sirve-json-be05fe02e67d.

Chaluisa, Frixon. 2017. [En línea] 2017.

Chen, Caterina. 2019. Significados.com. [En línea] 2019. [Citado el: 15 de mayo de 2020.] https://www.significados.com/tic/.

Coronel, Carlos, Morris, Steven y Rob, Peter. 2011.*Base de Datos:Diseno,Implementación y Administración.* 2011.

Correa, Javier Mauricio. 2016. Prezzi. [En línea] 2016. [Citado el: 29 de noviembre de 2019.] https://prezi.com/j84ywfydzvit/visual-paradigm/.

Cortes Iglesias, Manuel, Rodriguez Hernñandez, Cinthya y Cabrera, Marianelis. 2017.*SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE RIESGOS EN PROYECTOS.* 2017.

Costal, Dolors. 2014.*Introducción al diseño de Bases de Datos.* 2014. P06/M2109/0250.

Cruz, Ángel. 2018. [En línea] 2018.

Date, C.J. 2003.*Introducción a los sistemas de bases de datos.* La Habana : Félix Varela, 2003.

del Pino, Javier. 2018. developer. [En línea] 2018. [Citado el: 29 de noviembre de 2019.] https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n.

Domínguez, Pablo. 2018. OpenClassrooms. [En línea] 2018. https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery.

Fernández, María de los Ángeles. 2016.*Introducción práctica a la programacíon con Python.* 2016.

Fernández, Yahaira. 2016. SlideShare. [En línea] 2016. [Citado el: 25 de noviembre de 2019.] https://es.slideshare.net/yahairafernandezsegura/relaciones-en-bases-de-datos-70535917.

Galeano, Eduardo. 2016. aulapc.es. [En línea] 2016. [Citado el: 25 de noviembre de 2019.] http://www.aulapc.es/lupa_busquedas_posit.html1accesA~A60.00.

Gandarillas, Aurelio. 2017. metodologia. [En línea] 2017. [Citado el: 29 de noviembre de 2019.] https://metodologia.es/marco-de-trabajo/.

Guillén Pérez, Lianne, Gómez León, Yamilka y Fernández Pérez, Yamilis. 2019.*MODELO DE PLANIFICACION Y CONTROL DEL PROCESO DOCENTE .* 2019.

Gutiérrez, Pedro. 2017. GENBETA. [En línea] 2017. [Citado el: 1 de diciembre de 2019.]

Hostinger, Gustavo. 2019. hostinger. [En línea] 2019. [Citado el: 29 de noviembre de 2019.] https://www.hostinger.es/tutoriales/que-es-apache/.

Ilarri, Sergio. 2017.*Herramienta de Apoyo a la Docencia de Bases de Datos y Almacenes de Datos.* 2017.

ISO 9000. 2015. iso. [En línea] 2015. https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:es.

Labrada, Evelyn y Aragón, Yaniel. 2013.*Desarrollo del Módulo de Gestión de Reportes Estadísticos para el sistema AiresProxyAudit.* La Habana : s.n., 2013.

Leiva, Ivan. 2015. monografias. [En línea] 2015. [Citado el: 1 de diciembre de 2019.] https://www.monografias.com/trabajos72/base-datos/base-datos2.shtml.

LibroDjango. 2017. uniwebsidad. [En línea] 2017. [Citado el: 29 de enero de 2020.] https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv.

López, Ivan David. 2015. byspel. [En línea] 2015. [Citado el: 29 de noviembre de 2019.] https://byspel.com/librerias-javascript-interesantes/.

Menéndez, Rafael y Barzanallana, Arsenio. 2018. [En línea] 2018.

Microsoft. 2016. Revisiones de código y estándares de codoficación. [En línea] 2016. [Citado el: 20 de enero de 2020.] http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx..

Naranjo, David. 2018. DESDELINUX. [En línea] 2018. [Citado el: 29 de noviembre de 2019.] https://blog.desdelinux.net/pycharm-un-entorno-de-desarrollo-para-python/.

Olano, Jimmi. 2017.*Workbench MySQL.* 2017.

Olarte, Luis. 2018. conogasi. [En línea] 2018. [Citado el: 29 de noviembre de 2019.] http://conogasi.org/articulos/lenguaje-de-programacion/.

Onofre Cortez, Michel. 2017. SCRIBD. [En línea] 2017. https://es.scribd.com.

Palma-Orozco, Rosaura y Chavarría-Báez, Lorena. 2016.*Sistema de software para apoyar la impartición de base datos.* 2016.

Palomares, Kiko. 2017. kikopalomares. [En línea] 2017. [Citado el: 29 de noviembre de 2019.] https://kikopalomares.com/que-es-un-ide-de-programacion-y-un-editor-de-codigo-ligero-diccionario-del-programador/.

Pérez, Porto ,Julián. 2017. Definición.DE. [En línea] 2017. https://definicion.de/diagrama/.

Picurelli, Luis y Badal, Hector. 2019. Yeeply. [En línea] 2019. [Citado el: 20 de enero de 2020.] https://www.yeeply.com/blog/que-son-pruebas-unitarias/.

postgresqltutoria. 2020. postgresqltutorial. [En línea] 2020. [Citado el: 29 de noviembre de 2019.] https://www.postgresqltutorial.com/what-is-postgresql/.

Powell-Morse, Andrew. 2017. Airbrake. [En línea] 2017. [Citado el: 10 de enero de 2020.] https://airbrake.io/blog/sdlc/conceptual-model.

Riccio, Francisco. 2016. Manual Técnico de bd oracle. [En línea] 2016. [Citado el: 25 de noviembre de 2019.] https://manual-tecnico-bd-oracle.readthedocs.io/es/latest/Modelo%20de%20datos.html.

Rius, Durán Alejandra. 2016.*Herramienta para el diseño de Bases de Datos.* 2016.

Robledano, Ángel. 2019. OpenWebinars. [En línea] 2019. [Citado el: 29 de noviembre de 2019.] https://openwebinars.net/blog/que-es-javascript/.

Rojo, Silvana del Valle y Oliveros, Alejandro. 2012.*Requerimientos No funcionales para aplicaciones Web.* Argentina : s.n., 2012. ISSN:1850-2792.

Rouse, Margare. 2016. techtarget. [En línea] 2016. [Citado el: 29 de noviembre de 2019.] https://searchdatacenter.techtarget.com/es/definicion/Servidor-Web.

Ruiz, Alva. 2016. SlideShare. [En línea] 2016. [Citado el: 10 de enero de 2020.] https://es.slideshare.net/Alva_Ruiz/requerimientos-de-un-sistema-y-desarrollo-del-prototipo.

Salas, Josue. 2016. SlideShare. [En línea] 2016. [Citado el: 10 de enero de 2020.] https://es.slideshare.net/josuesalas5/diagramas-uml-diseo-de-sistemas-66609390.

Saldaña, Gabriel. 2017. Nethazard. [En línea] 2017. [Citado el: 10 de enero de 2020.] https://blog.nethazard.net.

Sánchez, Francisco. 2016. Base de Datos I. [En línea] 2016. [Citado el: 15 de enero de 2020.] https://manual-tecnico-bd-oracle.readthedocs.io/es/latest/Modelo%20de%20datos.html.

Sánchez, José Manuel. 2015.*Pruebas de software.Fundamentos y técnicas.* 2015.

Sánchez, Tamara Rodríguez. 2015.*Metodología de Desarrollo para la Actividad Productiva de la UCI.* 2015.

Santiago Sanchez, Avalos. 2019. SOMOSPNT. [En línea] 2019. [Citado el: 20 de enero de 2020.] https://somospnt.com/blog/103-load-testing-vs-stress-testing.

Sommerville. 2011.*Ingeniería de Software.* México : s.n., 2011. ISBN:978-607-32-0603-7.

Talento, Guillermo. 2017. Software Testing Bureau. [En línea] 2017. [Citado el: 20 de enero de 2020.] https://www.softwaretestingbureau.com/pruebas-funcionales/.

Techopedia. 2017. Techopedia. [En línea] 2017. [Citado el: 27 de noviembre de 2019.] https://www.techopedia.com/definition/20810/modeling-language.

Troya, Jose Maria. 2010.*Pressman.* 2010.

Uquizo, Elisa. 2016. Slideshare. [En línea] 2016. [Citado el: 15 de enero de 2020.] https://es.slideshare.net/Indiana_1969/patrones-grasp-76283581.

Zúñiga Ángeles, Albert. 2017.*Metodología de desarrollo de software.* 2017.

ANEXOS

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

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

-¿Considera que todos los estudiantes logran desarrollar adecuadamente la habilidad de transformación del MER al MR?

-¿Cuáles son las causas que podrían estar influyendo en el desarrollo de esta habilidad?

-¿La aplicación *RDB-Learning* permite la transformación del MER al MR?

-Consideras que resulta de interés seguir mejorando y ampliando la aplicación *RDB-Learning*.

-¿Qué nivel de apoyo le brinda el uso de la herramienta para la orientación y evaluación de ejercicios?

-¿Qué otras características y funcionalidades considera que deba presentar el módulo?

-¿Cómo debería realizarse la revisión y evaluación de los ejercicios resueltos en la plataforma?

-¿Conoces alguna aplicación que tenga estas funcionalidades?

Anexo 2. Encuesta para conocer el nivel de aceptación de la plataforma por los estudiantes y las facilidades que les brinda.

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

Respecto a la plataforma RDB-Learning responda:

1. ¿Conocías ya la herramienta RDB-Learning?
   _Sí _No
2. ¿La utilizas habitualmente?

_Sí _No

Indique su grado de conformidad del 1 al 10 (donde el 1 significa inconformidad total y el 10 conformidad total) con las afirmaciones siguientes:

1. RDB-Learning es una herramienta que ayuda en el aprendizaje de diversos aspectos relacionados con la asignatura de base datos.
2. RDB-Learning es una buena herramienta como soporte para el diseño de base datos y la realización de ejercicios.
3. RDB-Learning es fácil de manejar y posee una interfaz agradable.
4. RDB-Learning brinda apoyo para el estudio individual, contribuyendo notablemente en el desarrollo de las habilidades de la asignatura.
5. Es necesario seguir manteniendo y ampliando la herramienta.