

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



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

Título: Módulo que aplique técnicas descriptivas de minería de datos y analítica visual integrada a la Plataforma Educativa ZERA.

AUTORES: Jorge Delgado Rúa
Henry Díaz Torres

TUTORES: Ing. Irina Ivis Santiesteban Pérez
Ing. Miguel Medina Ramírez
Ing. Ing. Noralys Almeida Milanés

La Habana, junio de 2016



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)”

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

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

Henry Díaz Torres

Firma del Autor

Jorge Delgado Rúa

Firma del Autor

Ing. Irina Ivis Santiesteban Pérez

Firma del Tutor

Ing. Miguel Medina Ramírez

Firma del Tutor

Ing. Noralys Almeida Milanés

Firma del Tutor

Dedicatoria

De Henry:

La presente tesis la dedico a mi familia que gracias a su apoyo pude concluir mi carrera. A mi abuela, tías, mi padrastro y hermanito por su apoyo y confianza en todo lo necesario para cumplir mis objetivos como estudiante y persona. A mi madre por ser el espejo en el cual me reflejo, por sus virtudes infinitas y su gran corazón, lo que hacen que la admire cada día. A mis amigos, a mi institución y maestros por sus esfuerzos para que finalmente pudiera graduarme como un feliz profesional.

De Jorge:

Dedico esta tesis a mis padres, principalmente a mi madre, que siempre me apoyaron incondicionalmente en la parte moral y económica para poder llegar a ser un profesional de la Patria. A mi hermana y demás familia en general por el apoyo que siempre me brindaron día a día en el transcurso de cada año de mi carrera universitaria. A mis amigos, incluso los que hoy no están aquí y que cientos de kilómetros nos separan, porque fueron mi cimiento para la construcción de mi vida profesional, sentaron en mí las bases de responsabilidad y deseos de superación.

Agradecimientos

De Henry:

Quisiera agradecer antes que, a todos a mi madre, que ha jugado en mi vida el rol de padre, hermana, amiga, compañera y confidente.

A mi padrastro José Ángel, que nunca ha sido un padrastro, es un padre. A mi hermanito Brayan, que desde que nació trajo luz a nuestras vidas.

A mi abuela, a mis tías Maribel y Mireya, por cuidar todo lo que yo amo y que no pude cuidar por estar lejos.

A Odalis y a Yanet que siempre estuvieron presentes para brindarme su apoyo y que saliera adelante.

A mis amigos Guillermo Adrián, Wendys, Arián Danilo, Jorge, Bilmarys, Maylin, José Andrés, Yaiselín, Malidia, Rogelio, Tania, el Guille, Frank y Lester por los momentos compartidos. A los tutores por siempre confiar en nosotros.

De Jorge:

De Jorge: A mi mamá por ser mi guía a seguir en todo momento de mi vida en la universidad.

A mi hermanita, mi papá y Kennia, Guille, Edil y Darwis por su amor incondicional y apoyo.

A mis abuelos, a los que están y lo que no están por creer en mí y saber que si lo lograría. A mis tíos por tener confianza en mí.

Al profesor Piña a pesar de no ser nuestro tutor se portó como uno, gracias por tu apoyo.

A los tutores por estar ahí cada vez que nos hacía falta.

A mis amigos, los de mi casa, lo que están a km de aquí y los de la universidad. A mi compañero de tesis por soportarme y aguantarme. Y a todas las personas que de una forma u otra estuvieron presente en mi formación como ingeniero que voy hacer.

Resumen

Las Tecnologías de la Información y las Comunicaciones constituyen medios que facilitan el proceso de enseñanza, evidenciándose en diversas herramientas. La Plataforma Educativa ZERA es una de estas herramientas creada para la gestión del aprendizaje. ZERA almacena información referente a las actividades que realiza el estudiante a través de reportes. Sin embargo, la manera en la que estos reportes muestran la información le hace difícil al profesor realizar un análisis rápido del comportamiento de los estudiantes, además, hay información que no se almacena, por lo que la plataforma necesita resolver estas dificultades. Por ello se decide desarrollar un módulo haciendo uso de técnicas de agrupamiento de minería de datos y analítica visual para facilitar el seguimiento y control de la interacción del estudiante con la Plataforma Educativa ZERA. Se seleccionaron la metodología AUP-UCI para el desarrollo de software y CRISP-DM para la minería de datos. Como lenguajes del lado del cliente: HTML, CSS y JavaScript a través de la librería jQuery y del lado del servidor PHP. Se utilizó WEKA como herramienta de análisis de datos, Symfony como framework de desarrollo, NetBeans IDE como Entorno de Desarrollo Integrado, NGINX como servidor web y PostgreSQL como gestor de bases de datos.

Palabras claves: agrupamiento, analítica visual, minería de datos

Índice

<i>Introducción</i>	10
<i>Capítulo 1 Fundamentación Teórica</i>	14
1.1 <i>Introducción</i>	14
1.2 <i>Minería de datos y minería de datos educacional</i>	14
1.2.1 <i>Principales tareas y técnicas de minería de datos</i>	15
1.2.2 <i>Algoritmos de agrupamiento</i>	16
1.2.3 <i>Herramienta de minería de datos</i>	18
Integración con otras aplicaciones.....	19
1.2.4 <i>Aplicación de técnicas de minería de datos en plataformas educativas</i>	20
1.3 <i>Analítica visual</i>	21
1.3.1 <i>Técnicas de visualización de la información</i>	21
1.3.2 <i>Aplicación de técnicas de analítica visual en plataformas educativas</i> .	24
1.4 <i>Metodologías de desarrollo de software</i>	25
1.5 <i>Herramientas para el desarrollo de software</i>	29
1.5.1 <i>Lenguaje Unificado de Modelado (UML)</i>	29
1.5.2 <i>Herramienta CASE</i>	29
1.5.2.1 <i>Visual Paradigm versión 8.0 como herramienta de modelado</i>	30
1.5.3 <i>Lenguaje de programación</i>	30
1.5.3.1 <i>Lenguajes y tecnologías del lado del servidor</i>	31
1.5.3.2 <i>Lenguajes y tecnologías del lado del cliente</i>	34
1.5.4 <i>Entorno de Desarrollo Integrado</i>	35
1.7 <i>Librerías para graficar resultados</i>	38
<i>Conclusiones parciales</i>	39
<i>Capítulo 2 Propuesta de solución</i>	40

2.1	<i>Introducción</i>	40
2.2	<i>Modelo de dominio</i>	40
2.3	<i>Análisis de los conceptos del dominio</i>	40
2.4	<i>Diagrama del modelo del dominio</i>	41
2.5	<i>Propuesta del módulo</i>	42
2.5.1	<i>Historias de Usuario (HU)</i>	42
2.5.2	<i>Requisitos no funcionales</i>	44
2.6	<i>Descripción de actores del sistema</i>	45
2.7	<i>Descripción del proceso de agrupamiento</i>	45
2.7.2	<i>Análisis de los datos</i>	47
2.7.3	<i>Preparación de los datos</i>	50
2.7.4	<i>Modelado</i>	53
2.7.5	<i>Desarrollo o implementación</i>	56
	<i>Conclusiones parciales</i>	57
Capítulo 3 Desarrollo de la propuesta de solución		58
3.1	<i>Introducción</i>	58
3.2	<i>Patrón de arquitectura</i>	58
3.3	<i>Patrón de diseño</i>	58
3.4	<i>Diagramas de clases del análisis</i>	59
3.5	<i>Diagramas de colaboración del análisis</i>	60
3.5.1	<i>Diagramas clases del diseño</i>	61
3.5.2	<i>Diagramas de secuencia</i>	62
3.5.3	<i>Diagrama de componentes</i>	63
3.6	<i>Modelado de los datos</i>	64
3.7	<i>Descripción de las tablas de la base de datos</i>	65
3.8	<i>Pruebas de software</i>	67
3.8.1	<i>Niveles de prueba</i>	68

<i>3.8.2 Métodos de prueba</i>	69
<i>3.8.3 Diseño de casos de prueba (CP)</i>	69
<i>3.8.3.1 Resultados de las pruebas de caja negra</i>	70
<i>3.8.4 Pruebas unitarias</i>	70
<i>Conclusiones parciales</i>	71
<i>Conclusiones generales</i>	72
<i>Recomendaciones</i>	73
<i>Bibliografía</i>	74

Introducción

El empleo de las Tecnologías de la Información y las Comunicaciones (TIC)¹ ha originado cambios en numerosas esferas de la sociedad. Su incorporación en la educación ha abierto grandes posibilidades para mejorar los procesos de enseñanza y de aprendizaje. El uso de tecnologías en la educación tiene la finalidad de desarrollar las capacidades cognitivas del individuo. Se enfoca en el trabajo colaborativo y en red para el desarrollo de la participación y atiende la demanda de conocimiento de los recursos informáticos latente entre el público escolar.

Las aplicaciones de las TIC pueden situarse en el campo de la enseñanza superior y en el de la formación profesional, con destacado papel para el *e-learning* o educación virtual a distancia. *E-learning* se define como un "sistema de impartición de formación a distancia, apoyado en las TIC que combina distintos elementos pedagógicos: Instrucción clásica (presencial o autoestudio), las prácticas, los contactos en tiempo real (presenciales, videoconferencias o chats) y los contactos diferidos (tutores, foros de debate, correo electrónico" (1).

En la práctica, para llevar a cabo un programa de formación basado en *e-learning*, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Entre las herramientas más utilizadas para los ambientes o sistemas *e-learning* están los Sistemas de Administración de Aprendizaje o LMS², también ampliamente conocidos como plataformas de aprendizaje.

Un LMS es un sistema de gestión de aprendizaje *online* que permite administrar, distribuir, monitorear, evaluar y apoyar las diferentes actividades de un proceso de aprendizaje. Estos sistemas pueden utilizarse como núcleo del aprendizaje a distancia o como un complemento del aprendizaje presencial. Los LMS facilitan el seguimiento del proceso de aprendizaje de cada alumno, realizan evaluaciones, generan informes y ofrecen herramientas de comunicación tales como foros, chats o incluso videoconferencias (2).

En Cuba se crea el concepto Hiperentorno Educativo o de Aprendizaje. La Universidad de las Ciencias Informáticas (UCI), específicamente el Centro de Tecnologías para la Formación (FORTES) hace uso de hiperentornos educativos, la misma se trazó la tarea de implementar la Plataforma Educativa ZERA para la gestión del aprendizaje.

¹ TIC: Tecnologías de la Información y las Comunicaciones, se usa a menudo para referirse a cualquier forma de hacer cómputo.

² LMS: Learning Management Systems.

Esta plataforma está compuesta por varios subsistemas entre los que se encuentran: Administración, Bachiller, Gestión del Aprendizaje, Materia, Recursos y Reportes.

En la actualidad se generan reportes respecto a las acciones que realizan los estudiantes; tiempo que dedica a cada actividad, la calificación en cada tarea realizada, la interacción en los foros, las horas de uso en los ejercicios, entre otras. Sin embargo, estos reportes no muestran la información relacionada con la fecha y hora de conexión, el tiempo en el que se encuentra autenticado un usuario en la plataforma, el tiempo en cada uno de los módulos, la cantidad de recursos que visita, el tipo de recurso accedido, la cantidad de visitas que realiza al glosario, el total de palabras que revisa y cuáles fueron las mismas, además de la cantidad de visitas realizadas a un tema y el tiempo dedicado al mismo.

A medida que el estudiante interactúa con la plataforma se almacena en la misma toda la información respecto a los ejercicios realizados, tareas que constituyen evidencias enmarcadas en tres tipos específicos de actividades: evidencias, ejercicios y recorridos dirigidos. De estas actividades se muestra en el reporte los siguientes datos: nombre, complejidad, tipo de actividad y calificación alcanzada. Sin embargo, de estas evidencias no se muestra la fecha en el que se resolvió la misma, el momento en el que realizó un ejercicio con fecha de caducidad, ya sea en el primer o último día, el tipo de ejercicio que más resuelve en una jornada determinada, el tema que más se debate en el foro. De estas informaciones obviadas en el reporte, el profesor no cuenta con mecanismos que le permitan visualizar toda la información almacenada de uno o varios estudiantes en un período determinado, tampoco tiene una vía para agrupar a los mismos basados en el desempeño dentro de la Plataforma Educativa ZERA.

A partir de la situación problemática descrita surge como **problema a resolver**: La inexistencia de reportes para el análisis de la interacción de los estudiantes con los módulos Recurso, Tarea y Foro afecta el seguimiento y control de la actividad de los estudiantes en la Plataforma Educativa ZERA.

Se plantea como **objeto de estudio**, aplicación de técnicas de minería de datos y analítica visual en LMS.

El **campo de acción** está centrado en: la aplicación de técnicas descriptivas de minería de datos y analítica visual en la Plataforma Educativa ZERA.

Objetivo general: Desarrollar un módulo que aplique técnicas descriptivas de minería de datos y analítica visual para facilitar el seguimiento y control de la interacción del estudiante con la Plataforma Educativa ZERA.

Con el fin de dar cumplimiento al objetivo planteado se concretan las siguientes tareas de la investigación:

- Elaborar el marco teórico de la investigación.
- Realizar el análisis y diseño del módulo.
- Implementar un módulo que permita darle seguimiento y control a la interacción del estudiante con la Plataforma Educativa ZERA.
- Realizar pruebas a las funcionalidades del módulo.

Posibles resultados:

- Un módulo que muestre a través de las diferentes técnicas de visualización de la información, la interacción tanto individual como grupal de los estudiantes con la Plataforma Educativa ZERA, para contribuir a su seguimiento y control.

Se tiene como **hipótesis** que: El desarrollo de un módulo que aplique técnicas descriptivas de minería de datos y analítica visual facilitará el seguimiento y control de la interacción del estudiante con la Plataforma Educativa ZERA.

En la investigación se utilizan los siguientes **métodos teóricos**:

- **Análisis-síntesis**, el cuál posibilitó el estudio de fuentes bibliográficas referente al tema objeto de investigación, identificando elementos importantes y necesarios para dar solución al problema planteado
- **Inductivo-deductivo**, utilizado para determinar las características generales de las técnicas de visualización de la información y minería de datos en plataformas educativas, con el objetivo de determinar las soluciones factibles a incorporar en la propuesta.
- **Hipotético-deductivo**, el cual permitió proponer una hipótesis como consecuencia de inferencias del conjunto de datos empíricos o de principios y leyes más generales

Métodos empíricos:

- **Entrevista:** Se utilizó este método en la realización de entrevistas, con el fin de precisar el problema a resolver, así como validar la propuesta que se presenta.
- **Análisis documental:** El cual sirvió de guía en la revisión de los documentos utilizados para la investigación del presente trabajo de investigación.

El trabajo está estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica: En el mismo se aborda los principales conceptos para la comprensión de los temas que han sido considerados válidos para el desarrollo de la investigación. Se realiza un estudio de las plataformas educativas que hacen uso de la analítica visual en el ámbito educativo y se abordan las metodologías y herramientas analizadas para el desarrollo de la solución.

Capítulo 2: Propuesta de solución: Se realiza una descripción de la solución y sus principales características. Se presenta el modelo conceptual, así como los requisitos funcionales y no funcionales identificados. Además de las Historias de Usuario y la descripción de la metodología de minería de datos.

Capítulo 3: Desarrollo de la propuesta de solución: En este capítulo se expone aspectos a tener en cuenta en la implementación del módulo y los métodos utilizados para probar su funcionamiento.

Capítulo 1 *Fundamentación Teórica*

1.1 *Introducción*

Este capítulo contiene la base teórica del presente trabajo. En él se resumen los principales conceptos relacionados con el objeto de estudio y campo de acción de la investigación. Se incluyen además las últimas tendencias en la aplicación de la Analítica Visual en la educación, así como un análisis de las herramientas y técnicas que se han de utilizar para el desarrollo de la aplicación.

Los conceptos y tendencias aquí presentadas han sido obtenidos tras la investigación y revisión cuidadosa de la literatura disponible a nivel internacional, nacional y en el ámbito de la Universidad de las Ciencias Informáticas. Los mismos ofrecen una panorámica global sobre el estado (conocimientos, desafíos, vías de solución entre otros) pasado y presente en el dominio del problema.

1.2 *Minería de datos y minería de datos educacional*

La **Minería de Datos** (MD) consiste en la extracción de información que reside de manera implícita, desconocida o previamente ignorada, que puede ser potencialmente útil, de un conjunto de datos. Se puede considerar a la MD como una colección de diferentes técnicas que sirven para inducir el conocimiento e información de manera estructurada de un gran conjunto de datos (3).

A continuación, se muestran dos conceptos de MD:

- La Minería de Datos es el proceso que se enfoca en la búsqueda de información nueva, valiosa y no trivial dentro de grandes volúmenes de datos (4).
- La Minería de Datos es el proceso de descubrir conocimientos interesantes, como patrones, asociaciones, cambios, anomalías y estructuras significativas a partir de grandes cantidades de datos almacenadas en bases de datos, Data Warehouses, o cualquier otro medio de almacenamiento de información (5).

La MD es un proceso apoyado en técnicas y herramientas que descubre a partir de datos almacenados nuevos conocimientos válidos y potencialmente útiles para ayudar la toma de decisiones más seguras, que propicien algún beneficio a las organizaciones.

La **Minería de Datos Educativo (MDE)** desarrolla métodos y aplica técnicas tomadas de las estadísticas, el aprendizaje automatizado y la minería de datos; con el fin de analizar los datos recogidos durante los procesos de enseñanza y aprendizaje. La MDE pone a prueba las teorías de aprendizaje e informa la práctica educativa (6).

Los investigadores de la MDE consideran como las metas de su investigación, éstas que a continuación se relacionan:

- Predecir el comportamiento futuro de los estudiantes mediante la creación de modelos que incorporen información detallada, por ejemplo: el conocimiento, la motivación y las actitudes de los estudiantes.
- Descubrir o mejorar los modelos de dominio que caractericen el contenido que ha de aprenderse y las secuencias de instrucción óptimas.
- Estudiar los efectos de los distintos tipos de apoyo pedagógico que pueden proveer el software de aprendizaje.

1.2.1 Principales tareas y técnicas de minería de datos

En la práctica, las dos metas principales de la MD tienden a ser la predicción y la descripción, aunque comprende también otras tareas dependiendo de las necesidades de la persona que analiza los datos. El modelado descriptivo consiste en describir todos los datos (o los procesos que los generan), ejemplos de tales descripciones incluyen modelos de la distribución general de probabilidad en los datos (estimación de densidad), el particionamiento del espacio p-dimensional en grupos (agrupamiento o segmentación) y los modelos que describen la relación entre las variables (modelado de dependencias) (7). El modelado predictivo tiene como objetivo construir un modelo capaz de predecir el valor de una variable a partir de los valores conocidos de las demás variables. La diferencia clave entre la predicción y la descripción, es que la predicción tiene como objeto una sola variable; mientras que en los problemas descriptivos ninguna de las variables es primordial para el modelo (5).

Técnicas de minería de datos

Las técnicas de MD provienen de la inteligencia artificial y de la propia estadística. Dichas técnicas, no son más que algoritmos sofisticados que se aplican sobre un conjunto de datos para obtener resultados, patrones o modelos a partir de los datos recopilados. Las técnicas de Minería de Datos se clasifican en dos categorías: supervisadas o predictivas y no supervisadas o descriptivas.

- Agrupamiento (8):

El proceso de agrupamiento es una técnica para el modelado descriptivo que consiste en la división de los datos en grupos de objetos similares. El representar los datos en una serie de *clusters* implica la pérdida de detalles, pero consigue la simplificación de los mismos. El agrupamiento es una técnica más del aprendizaje automatizado en la que el aprendizaje es no supervisado (no se conocen a priori las etiquetas de las clases). Desde un punto de vista práctico, el *clustering* juega un papel muy importante en aplicaciones de MD, tales como exploración de datos científicos, recuperación de la información, minería de texto, aplicaciones sobre bases de datos espaciales (datos procedentes de astronomía), aplicaciones web, *marketing*, diagnóstico médico, análisis de ADN³ en biología computacional y muchas otras.

- Clasificación (8):

La clasificación tiene como idea colocar un objeto dentro de una clase o categoría, basándose para ello en sus otras características. La etiqueta de clase es un valor (simbólico) discreto y es conocido para cada objeto. Esto se conoce como aprendizaje supervisado. Los modelos de clasificación son utilizados sobre todo para el modelado predictivo.

1.2.2 Algoritmos de agrupamiento.

Debido al gran número de algoritmos de agrupamiento y a las diversas estrategias seguidas a la hora de agrupar los datos, no es sencillo realizar una clasificación clara. La clasificación tradicional de los algoritmos los divide en tres categorías principales: jerárquicos, particionales y basados en densidad.

Los algoritmos jerárquicos construyen una serie de particiones anidadas de forma aglomerativa y divisiva. Una partición anidada de forma aglomerativa se refiere a aquellos clústeres que se van uniendo, formando particiones con cada vez menos clústeres. En el segundo caso o sea una partición divisiva es el caso en que los clústeres se van dividiendo de forma que las particiones tienen cada vez más clúster.

La ventaja principal de este tipo de algoritmo es que es capaz de capturar la posible estructura jerárquica de los datos. Un algoritmo jerárquico puede crear una jerarquía de clúster donde se representen los distintos niveles de detalle en los que podemos estructurar los datos.

³ ADN: Ácido desoxirribonucleico.

El particional genera una única partición de los datos. Generalmente este tipo de algoritmo es menos costoso y más utilizado.

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad.

Los algoritmos que serán descritos a continuación fueron seleccionados teniendo en cuenta el tipo de datos a procesar, la calidad del agrupamiento y el tiempo de ejecución.

K-means⁴

Es uno de los más simples y conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (fijados a priori).

La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente (9).

El problema del empleo de estos esquemas es que fallan cuando los puntos de un grupo están muy cerca del centroide de otro grupo, también cuando los grupos tienen diferentes tamaños y formas.

DBSCAN⁵

Es el primer algoritmo basado en densidad, en el que se definen los conceptos de punto central (puntos que tienen en su vecindad una cantidad de puntos mayor o igual que un umbral especificado), borde y ruido.

El algoritmo comienza seleccionando un punto p arbitrario, si p es un punto central, se comienza a construir un grupo y se ubican en su grupo todos los objetos alcanzables desde p. Si p no es un punto central se visita otro objeto del conjunto de datos. El proceso continúa hasta que todos los objetos han sido procesados. Los puntos que quedan fuera de los grupos formados se llaman puntos ruido, los puntos que no son ni ruido ni centrales se llaman puntos bordes. De esta forma DBSCAN construye grupos

⁴ K-means: Método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo más cercano a la media.

⁵ DBSCAN: Densidad basada en agrupación espacial de aplicaciones con ruido. (Density Based Spatial Clustering of Applications with Noise)

en los que sus puntos son o puntos centrales o puntos borde, un grupo puede tener más de un punto central (10).

OPTICS⁶

La motivación para la realización de este algoritmo se basa en la necesidad de introducir parámetros de entrada, en casi todos los algoritmos de agrupamiento existentes son difíciles de determinar, además en conjuntos de datos reales no existe una manera de determinar estos parámetros globales, el algoritmo OPTICS trata de resolver este problema basándose en el esquema del algoritmo DBSCAN creando un ordenamiento de la base de datos para representar la estructura del agrupamiento basada en densidad, además puede hacer una representación gráfica del agrupamiento incluso para conjuntos de datos grandes (10).

1.2.3 Herramienta de minería de datos

Existen algunas herramientas diseñadas para extraer conocimiento desde bases de datos que contienen grandes cantidades de información. A continuación, se analizarán varias de estas herramientas con el fin de seleccionar la indicada para el desarrollo de la presente investigación.

Weka⁷ es una extensa colección de algoritmos de máquinas de conocimientos implementados en java. Estos algoritmos son útiles para ser aplicados sobre datos mediante los interfaces que ofrece o para embeberlos dentro de cualquier aplicación. Esta herramienta permite realizar un grupo de transformaciones necesarias sobre los datos, trae implementada un grupo de tareas de la MD: clasificación, regresión, agrupamiento, asociación y visualización. Weka está diseñada como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla (11).

Entre sus características se encuentra que, contiene un grupo de herramientas que permiten el análisis de los datos. Su licencia es GPL⁸, lo que significa que este programa es de libre distribución y difusión. Además, está programado en Java⁹, es independiente de la arquitectura y funciona en cualquier plataforma sobre la que haya una máquina virtual de Java disponible (9).

⁶ OPTICS: Puntos pedidos para identificar la estructura de clústeres (Ordering Points to Identify the Clustering Structure)

⁷ Weka: Waikato Environment for Knowledge Analysis, en español «entorno para análisis del conocimiento de la Universidad de Waikato.

⁸ GPL: Licencia Pública General de GNU.

⁹ Java: Lenguaje de programación orientado a objetos.

RapidMiner es el líder mundial de código abierto para la minería de datos debido a su combinación de tecnología de primera calidad y gran número de funcionalidades. Esta aplicación cubre un amplio rango de minería de datos. Además de ser una herramienta flexible para aprender y explorar la minería de datos, la interfaz gráfica de usuario tiene como objetivo simplificar el uso para las tareas complejas de esta área (12).

Características de RapidMiner (12):

- Es un sistema de prototipo para el descubrimiento del conocimiento y MD.
- Es un software de tipo Open-Source con licencia GNU¹⁰ GPL, basado en java.
- Trabaja bajo las plataformas Windows y Linux.
- La característica más importante es la capacidad de jerarquizar cadenas del operador y de construir complejos árboles de operadores.

SAS Enterprise Miner es una solución de minería de datos que proporciona gran cantidad de modelos y de alternativas. Permite determinar pautas y tendencias, explica resultados conocidos e identifica factores que permiten asegurar efectos deseados. Además, compara los resultados de las distintas técnicas de modelado, tanto en términos estadísticos como de negocio, dentro de un marco sencillo y fácil de interpretar (13).

Fundamentación de la selección

La herramienta seleccionada para realizar el proceso de Minería de Datos es Weka en su versión 3.7.10. A continuación se describen algunas de las razones que justifican el uso de Weka.

- Es de distribución libre y multiplataforma.
- Cuenta con una librería que puede ser embebida en la propia aplicación.
- Es una de las herramientas de MD más utilizadas. Muchos de los experimentos consultados hacían uso de esta herramienta.
- Cuenta con amplia documentación y ayuda en línea.
- Además, el conocimiento previo de la herramienta por parte de los autores, permitió un ahorro considerable de tiempo en el desarrollo de la solución.

Integración con otras aplicaciones

MOA¹¹ es un framework de código abierto para la minería de flujo de datos, con una creciente comunidad muy activa (blog). Incluye una colección de algoritmos de

¹⁰ GNU: Acrónimo recursivo de "GNU's Not Unix" (en español: GNU no es Unix). Es un sistema operativo de tipo Unix desarrollado por y para el Proyecto GNU y auspiciado por la Free Software Foundation.

¹¹ MOA: Massive Online Analysis. (Análisis masivo en línea)

aprendizaje automático (clasificación, regresión, clustering, detección de valores atípicos, detección concepto de deriva y los sistemas de recomendación) y herramientas para la evaluación. MOA apoya la interacción bidireccional con Weka (aprendizaje automático). Fue publicado bajo la licencia GNU GPL (14).

1.2.4 Aplicación de técnicas de minería de datos en plataformas educativas

Recientemente, se ha incrementado el interés en utilizar la minería de datos en el estudio educacional, centrándose en el desarrollo de métodos de descubrimiento que utilicen los datos de plataformas educativas y en el uso de esos métodos para comprender mejor a los estudiantes y el entorno en el que aprenden. A continuación, se menciona un conjunto de investigaciones relacionadas a la aplicación de técnicas de minería de datos en plataformas educativas.

- Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.(Proceso de identificación de comportamiento de comunidades educativas centradas en EVEAs¹²) (15): Este proyecto de investigación se apoya en el uso de Entornos Virtuales de Enseñanza y Aprendizaje (EVEAs) para determinar la comprensión de comportamientos de comunidades mediadas por tecnología con énfasis en comunidades educativas y la formulación de una propuesta de identificación de patrones de comportamientos poblacionales en comunidades educativas mediadas por EVEAs utilizando procesos de explotación de información.
- Universidad Nacional de Chimborazo, Escuela Superior Politécnica de Chimborazo (Patrones de uso docente de Evos en la Escuela de Ingeniería en Sistemas Informáticos de la ESPOCH) (16): Este trabajo encuentra y valora patrones de uso docente de los Entornos Virtuales de Aprendizaje (Evos) de los profesores de la Escuela de Ingeniería en Sistemas (EIS) de la Facultad de Informática y Electrónica (FIE) de la Escuela Superior Politécnica de Chimborazo (ESPOCH). Se utiliza un procedimiento en base a los log del apartado “Informes” de los Evos, que permite determinar y analizar variables tales como el lugar, horario y fecha frecuente de trabajo, los diferentes niveles de interacción y los recursos y actividades generados a lo largo de nueve semestres académicos. Para recoger y analizar los datos se aplicó un procedimiento basado en conceptos de técnicas estadísticas y de minería de datos.

¹² EVEAs: Entornos Virtuales de Enseñanza y Aprendizaje.

- Universidad Autónoma del Estado de Hidalgo - Escuela Superior Huejutla (Identificación de Estilos de Aprendizaje en Alumnos Universitarios de Computación de la Huasteca Hidalguense mediante Técnicas de Minería de Datos) (17): Esta investigación se centra en la aplicación de técnicas de minería de datos para descubrir las combinaciones de estilos de aprendizaje mostradas por estudiantes de educación superior de computación o áreas afines. El modelo seguido para determinar los estilos de aprendizaje es el desarrollado por Neil Fleming y Collen Mills llamado VARK con el cual se puede determinar los estilos de aprendizaje de un alumno dentro de las cuatro modalidades que cubre este modelo Visual, Auditivo, Lectura/Escritura y Quinésica.

1.3 Analítica visual

La analítica visual es un área emergente de la investigación y su práctica tiene como objetivo apoyar el razonamiento analítico a través de interfaces visuales interactivas (18). La visualización se integra como núcleo y, en conjunto con otras disciplinas científicas, mejora la división del trabajo entre el hombre y la máquina, combina el razonamiento analítico con visualización interactiva, es decir, representaciones visuales y técnicas de interacción, de apoyo a la producción, presentación y difusión de resultados analíticos que comuniquen información en el contexto adecuado para una variedad de audiencias.

1.3.1 Técnicas de visualización de la información

Actualmente, el uso cada vez mayor de las nuevas tecnologías para apoyar el aprendizaje ha fomentado la creación de nuevos mecanismos que ayuden a extraer información que no está disponible a primera vista. La analítica visual se enfoca en proponer y desarrollar innovadores soluciones visuales.

Existen numerosas técnicas de visualización de la información, dentro de las cuales se pueden citar:

- **Línea de tiempo en espiral**

Es una representación visual interactiva que se utiliza para las etiquetas que, por lo general se muestran en orden, utilizadas para analizar el uso de un LMS a través del tiempo. Este punto de vista se puede adaptar a las necesidades del usuario, por lo que se pueden explorar todos los datos temporales disponibles o pasar de la vista general al detalle de una persona o actividad en un período de tiempo determinado.

La visualización consta de tres paneles visuales que proporcionan diferentes vistas de datos. La representación principal es la línea de tiempo en espiral, en la parte central, que, en su forma más simple, no es más que una secuencia de eventos codificados por colores. Estos están ordenados siguiendo las agujas del reloj, colocando los datos más antiguos en el centro de la espiral y los eventos más recientes más alejados (18).

En el presente trabajo se pretende usar esta técnica para descubrir y analizar los intervalos de tiempo en que la participación del usuario es más representativa.

- **Nubes de etiquetas o nubes de palabras**

La nube de etiquetas proporcionan un medio para que los usuarios se formen una impresión general del conjunto de los contenidos y la esencia de lo que se trata. Son una lista de las etiquetas que, por lo general se muestran en órdenes alfabéticos y dependiendo de su popularidad. El tamaño de las etiquetas se corresponde con su frecuencia de uso.

La popularidad de una etiqueta se expresa por el tamaño de la fuente (en relación con las otras etiquetas) y, por tanto, se reconoce fácilmente. Las nubes de palabras pueden evolucionar a medida que los datos asociados cambian a través del tiempo.

Esta visualización se adapta a las necesidades del usuario, explorando todas las discusiones, foros, cursos y datos de los usuarios disponibles, y pasando de la vista general al detalle de una determinada persona, curso o discusión dentro de un período de tiempo (18).

Esta técnica se utilizará en el glosario de términos para mostrar aquellas palabras leídas por los estudiantes y aquel término con mayor índice de visita.

- **Coordenadas paralelas**

Las coordenadas paralelas son una forma común de visualizar los datos de alta dimensión en el espacio limitado y permitir la exploración de las tendencias. En los ejes verticales representan diferentes métricas. Las métricas difieren sobre la base de los datos disponibles. A cada estudiante se muestra como una línea horizontal polígona; de igual forma para el caso de los cursos, las métricas serían el promedio de uso de los estudiantes, el promedio de notas de éstos, el número de foros, discusiones y postas.

Esta herramienta de analítica visual aborda el análisis de los datos por medio del uso de servicios web en un entorno virtual de aprendizaje, da la posibilidad de utilizar en cualquier momento dado, es independiente de la versión del ambiente virtual de

aprendizaje, además de poder ser utilizada offline, online, en web o aplicación de escritorio (18).

Esta técnica de representación de la información se pretende utilizar para determinar cuáles fueron las actividades más realizadas por los estudiantes.

- **Gráfico de radar**

El gráfico radar es una visualización compartida sobre cómo se está en relación a cierto tema y cómo se desea estar. Sirve para reconocer las diferentes percepciones de los integrantes de un equipo, establecer metas de crecimiento, buscar sinergia en la acción y hacer seguimiento de los avances (19).

Los pasos para armarlo y usarlo pueden ser:

1. Factores, ¿en qué queremos mejorar?
2. Situación inicial, ¿cómo estamos
3. Situación deseada, ¿cómo queremos estar?
4. Chequeo avance, ¿cómo vamos?

- **Grafos**

Los grafos, han tenido una amplia aceptación dentro de las representaciones de la estructura de la web y más aún en las redes sociales online. Los registros de datos a menudo tienen alguna relación con otras piezas de información. Los grafos son muy utilizados para representar esas interdependencias. Un grafo se compone de un conjunto de objetos denominados nodos y conexiones entre estos objetos llamadas arcos. Es común imaginar que la estructura de una página web se pueda representar como un grafo, donde los recursos son los nodos y los enlaces los arcos a estos. Por ello parece acertado pensar en grafos a la hora de representar relaciones entre entidades (20).

El dibujado de grafos direcciona el problema de visualizar información estructural o relacional construyendo representaciones visuales geométricas de grafos o redes que son los modelos subyacentes en una gran cantidad de datos abstractos. Los grafos permiten expresar de una forma visualmente muy sencilla y efectiva las relaciones que se dan entre elementos de muy diversa índole, y resultan especialmente útiles en la representación visual de estructuras de datos en red y jerárquicas. En el caso de estructuras de datos multidimensionales, muchas veces es necesario emplear previamente técnicas de poda de enlaces entre nodos, con el objetivo de que el grafo resulte comprensible y “visualizable” (21).

- **TreeMaps**

Los TreeMaps son un tipo de visualización ideada a principios de los 90 por Ben Shneiderman, que permite representar jerarquías de forma que se optimiza el llenado del espacio y, además, ver los atributos de la misma e identificar patrones anómalos o propiedades de la jerarquía usando colores.

Los TreeMaps son representaciones planas bi-dimensionales de estructuras de datos jerárquicas, en los que se optimiza el espacio visual rellenándolo u ocupándolo completamente (21).

El espacio de la visualización está dividido en rectángulos a los que se les asigna un tamaño y un orden en función de una variable cuantitativa. Los niveles de la jerarquía del TreeMaps se visualizan como rectángulos que contienen otros rectángulos. Cada conjunto de rectángulos del mismo nivel de la jerarquía representa una columna o una expresión de una tabla de datos. Cada rectángulo individual de un nivel de la jerarquía representa una categoría de una columna.

1.3.2 Aplicación de técnicas de analítica visual en plataformas educativas

La creciente participación de los estudiantes, la omnipresencia de las TIC y el beneficio potencial de la observación del proceso de aprendizaje han llevado a la aparición en los últimos años de un campo de investigación en la que la tecnología se utiliza para supervisar, analizar, deducir y actuar en el desarrollo de una experiencia de aprendizaje, el “*Learning Analytics*” (LA). Muchos han sido los sistemas e instituciones que han investigado y creado proyectos sobre la base del mismo criterio, el control y seguimiento de la actividad estudiantil a través de la participación de estos en las plataformas educativas. Algunos de estos trabajos se presentan en la siguiente investigación:

- Facultad de Informática de la Universidad Complutense de Madrid, España: (Sistema de detección de estudiantes en riesgo de abandonar un curso o dejar los estudios): En este sistema se hace uso de las coordenadas paralelas como técnica de visualización de la información. Haciendo uso de indicadores numéricos, como el número de accesos a la plataforma, el número de mensajes escritos, análisis de contenido de la calidad de los mensajes, la relación entre la lectura y la escritura, y la percepción que los estudiantes tienen de la actividad a través de entrevistas, cuestionarios e informes de evaluación del curso realizados por el alumno, se puede controlar y dar seguimiento a futuros estudiantes que puedan abandonar el curso o los estudios.

- Universidad de Granada, España (Sistema de Información Geográfica): En este trabajo se presentan las ideas esenciales de la integración de los Sistemas de Información Geográfica con la visualización científica de la información, se mencionan las principales técnicas de visualización de datos multiparamétricos (coordenadas paralelas), así como las más adecuadas para el análisis visual de datos meteorológicos. Se muestra el uso de estas técnicas con datos meteorológicos de la provincia de Villa Clara, se analizan algunas variantes de implementación y se implementa una integración mezclada mediante la incorporación de un módulo de VC¹³ a un sistema de información geográfica. Actualmente se utilizan este sistema para resolver problemas tan diversos como la planificación de la extinción de incendios, el análisis de riesgos ambientales o la propagación de contaminantes.
- Universidad de Salamanca (Herramienta de analítica visual para el seguimiento de la actividad de los estudiantes de asignaturas de Ingeniería del Software en el Campus Virtual Studium): A través de la técnica de nubes de palabras se creó una aplicación que facilitó la obtención de información de los períodos de mayor uso del campus virtual y en qué actividades, además permitió conocer qué elementos del campus virtual (foros, discusiones, etc.) son los más sobresalientes y qué usuarios han participado en ellos.

Problemas encontrados en los trabajos actuales:

- Flexibilidad e independencia de conexión con las fuentes de datos y respeto a la privacidad e integridad de los datos personales.
- Poca expresividad visual.
- Pérdida del contexto de navegación.
- Saturación de elementos visuales.
- Falta de mecanismos de interacción.
- Manejo inadecuado de los planos visuales.
- Escasa personalización de la exploración.

1.4 Metodologías de desarrollo de software

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. La misma consiste en una filosofía de desarrollo con una base de procesos de desarrollo de software como múltiples herramientas, modelos y métodos, para asistir en el proceso de desarrollo de software.

¹³ VC: Visualización científica.

Es promovida por algún tipo de organización ya sea esta pública o privada que es la que se encarga de promover estas metodologías. Cada metodología tiene más o menos su propio enfoque de en lo que debería de consistir un proyecto de desarrollo de software. Pero todas ellas se basan en una serie de enfoques generalistas (22).

Entre las metodologías de desarrollo se encuentran:

Tradicionales:

- RUP (Rational Unified Process)
- MSF (Microsoft Solution Framework)
- Win-WinSpiralModel
- Iconix

Ágiles:

- XP (eXtremeProgramming)
- Scrum
- Crystal
- DSDM (Dynamic Systems Development Method)
- FDD (Feature Driven Development)
- Extreme Modeling
- AUP (Agile Unified Process)

Diferencias entre las metodologías ágiles y las tradicionales:

Tabla 1. Diferencias entre metodologías ágiles y tradicionales

<i>Metodologías Ágiles</i>	<i>Metodologías Tradicionales</i>
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas

No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Para la selección de la metodología de desarrollo a utilizar, se tuvieron en cuenta las metodologías ágiles, éstas entregan productos en un corto período de tiempo, realizan un mejoramiento constante de la calidad del software, detectan errores, permiten conocer el estado actual del proyecto y mantienen un control del mismo por parte del cliente y el equipo de desarrollo.

En esta sección se describen las metodologías SCRUM, AUP y XP:

- **Scrum** es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (23).
- **XP** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en re-alimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos

con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (24).

- **AUP** es una versión simplificada de RUP que describe de forma simple y fácil de comprender el uso de técnicas ágiles para el desarrollo de aplicaciones que permanezcan dentro de los conceptos de RUP. AUP promueve principios ágiles, de los cuales se acentúan particularmente los siguientes: Integración continua, Refactorización, Propiedad colectiva del código, Bienestar del programador. AUP establece que debe determinarse una persona que se encargue de velar por el cumplimiento de los requisitos y las prioridades del sistema, esta persona puede ser o bien el cliente en cuestión o un equipo de analistas. En el caso particular, descrito en el documento, este importante rol fue desempeñado por el equipo de analistas, quienes fueron los encargados de hacer cumplir, por parte del equipo de desarrollo, las exigencias del cliente de manera eficiente. RUP Ágil es flexible y propone los mismos roles y artefactos que RUP, solo que no hay necesidad de generar toda la documentación que se requiere en cada flujo de trabajo (25).

Fundamentos de la selección

AUP es una metodología flexible que no requiere de una gran cantidad de desarrolladores. Es concisa en el aspecto de la documentación, permitiendo generar solo la necesaria y no la especificada para cada flujo de trabajo como lo hace RUP. Está diseñada para trabajar en proyectos pequeños donde la atención se centra en las actividades que realmente son importantes. Permite el uso de herramientas de cualquier tipo, incluyendo aquí las de código abierto. Es fácil de manejar a través de herramientas de edición HTML sin necesidad de ser adaptada y es una metodología que se ajusta y aprovecha las ventajas que brindan las metodologías ágiles.

Aunque XP es una metodología popular y que brinda muchas ventajas presenta también desventajas. Ejemplo de estas es que no se genera documentación. En la actualidad en cualquier sistema por más simple que sea el cliente solicita que le sea entregado además del software manuales para los usuarios y otras documentaciones.

XP por otra parte, no es viable en el desarrollo de sistemas que no requieren de la presencia del usuario, pues este requiere que el usuario esté siempre presente como un miembro más del equipo de desarrollo. Además, al ser uno de los principios la programación en pareja, en esta metodología surge la interrogante de que si permite que los desarrolladores se encuentren en diferentes situaciones geográficas. Siendo esto un inconveniente para los desarrolladores.

De acuerdo con las características que presentan los servicios a desarrollar y tras el estudio realizado a las diferentes metodologías de desarrollo se determina como metodología de desarrollo a utilizar AUP.

1.5 Herramientas para el desarrollo de software

Los lenguajes para modelado, los de programación y los entornos de desarrollo integrado, constituyen herramientas que permiten la confección y el desarrollo de las aplicaciones, a continuación, se mencionan los estudiados:

1.5.1 Lenguaje Unificado de Modelado (UML)

Para modelar la integración se utilizará *Unified Modeling Language* (UML por sus siglas en inglés), este es un lenguaje útil para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML ayuda a visualizar el diseño y a hacerlo más accesible para otros (26).

Propiedades que hacen de UML un lenguaje de modelado estándar (27):

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por el Grupo de Gestión de Objetos (OMG, por sus siglas en inglés).
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clases, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de las máquinas.

1.5.2 Herramienta CASE

Las herramientas de ingeniería de software asistidas por ordenador o herramientas CASE (Computer Aided Software Engineering) son elementos indispensables para concebir una aplicación informática. El uso de las mismas reduce los gastos y el tiempo en el ciclo de desarrollo del software ya que permiten modelar fácilmente procesos y sistemas, con previo dominio de al menos un lenguaje de modelado soportado por la herramienta. Estas incluyen un conjunto de programas que facilitan la

optimización de un producto ofreciendo apoyo permanente a los analistas, ingenieros de software y desarrolladores.

CASE es la aplicación de métodos y técnicas quedan utilidades a los programas, por medio de otros, procedimientos y su respectiva documentación. En esta investigación se hace referencia a las herramientas que ayudan a la gestión de requisitos; es decir al proceso de identificación, asignación y seguimiento de los mismos, incluyendo interfaz, verificación, modificación y control de cada requisito, durante el ciclo de vida del proyecto. Los cambios/actualizaciones de requisitos deben ser gestionados para asegurar que se mantenga la calidad del producto (28).

Hasta hace poco tiempo las herramientas para la gestión de requisitos de software se limitaban a editores de texto, los cuales hacían de esta tarea una labor tediosa y confusa. Actualmente, se cuenta con múltiples opciones, como las que se mencionan a continuación:

Se hará uso de las mismas para facilitar el modelado de procesos y del sistema.

1.5.2.1 Visual Paradigm versión 8.0 como herramienta de modelado

Visual Paradigm para UML 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. El software de modelado UML contribuye a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra con IDE's¹⁴ como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland). Su diseño está centrado en casos de uso y se usa lo mismo para UML¹⁵ que para BPMN (Business Process Modeling Notation) (29).

1.5.3 Lenguaje de programación

El lenguaje de programación se define como un conjunto de instrucciones consecutivas y ordenadas que llevan a ejecutar una tarea específica. Dichas instrucciones se denominan "código fuente", el cual es único para cada lenguaje y está diseñado para cumplir una función o propósito específico.

El lenguaje de programación es el idioma artificial utilizado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Permite

¹⁴ IDE's: Entorno de programación integrado.

¹⁵ UML: Son las siglas de "Unified Modeling Language" o "Lenguaje Unificado de Modelado". Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos

especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo otras circunstancias. Está compuesto por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Son utilizados para controlar el comportamiento físico y lógico de una máquina a través de la creación de programas, logrando expresar algoritmos con precisión, o como modo de comunicación humana (30).

1.5.3.1 Lenguajes y tecnologías del lado del servidor

Los lenguajes de programación se pueden clasificar en diferentes tipos como los de máquina, ensamblador, compilado, interpretado, declarativo, imperativo y orientado a objeto. Estos se fueron desarrollando a medida que la complejidad de las tareas que realizaban las computadoras aumentaban, donde pasaron de instrucciones que ejercían el control directo sobre el hardware a algoritmos en los que se podía expresar de una manera adecuada a la capacidad cognitiva humana. Entre esta última clasificación se encuentra el lenguaje de programación orientado a objeto, siendo el más utilizado por los desarrolladores de software en su diseño, debido a la representación y trabajo con los objetos.

PHP versión 5.3.0

El acrónimo recursivo que significa PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML¹⁶. Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML, enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno (31).

Es una alternativa de fácil acceso debido a que es libre y permite aplicar técnicas de programación orientada a objetos. Está completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Su programación es segura y confiable porque al escribir el código fuente en

¹⁶ HTML: HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.

PHP, se hace invisible al navegador y al cliente, debido a que el servidor es el encargado de ejecutar el código y enviar su resultado HTML al navegador.

Este lenguaje, es empleado durante el desarrollo de la extensión para la confección de las clases necesarias en el trabajo con la capa de acceso a datos del ORM¹⁷ ya que brinda la facilidad de generar páginas dinámicas e interactivas, además de que el framework utilizado en la solución de la investigación (Symfony), está desarrollado en PHP.

Framework: Symfony, versión 1.4.20

Symfony es un framework que sigue el patrón arquitectónico MVC (Modelo-Vista-Controllador) diseñado para optimizar el desarrollo de las aplicaciones web. Automatiza las tareas más comunes de los proyectos web, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony es maduro, estable, profesional y está muy bien documentado. Está desarrollado completamente con PHP 5 y está enfocado al desarrollo en el mismo lenguaje de programación. Algunas de sus características más relevantes son (32):

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos. Esto se garantiza a través de un ORM (Propel o Doctrine).
- Implementa Front Controller como patrón derivado del MVC.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Soporta la instalación de extensiones o “plugins” para añadir nuevas funcionalidades.

Este framework tiene las siguientes facilidades para el desarrollo de la propuesta de solución:

¹⁷ ORM: Object-Relational mapping, o lo que es lo mismo, mapeo de objeto-relacional.

- Simplifica el desarrollo de la aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.
- Está desarrollado en PHP.
- Adaptable a las políticas y arquitecturas propias de la Plataforma Educativa ZERA.
- Proporciona herramientas para el desarrollo de pruebas unitarias y funcionales.

Doctrine

Doctrine es una librería para PHP que permite trabajar con un esquema de base de datos como si fuese un conjunto de objetos, y no de tablas y registros. Inspirado en Hibernate, que es uno de los ORM más populares y grandes que existen y que brinda una capa de abstracción de la base de datos muy completa. La característica más importante es que da la posibilidad de escribir consultas de base de datos en un lenguaje propio llamado Doctrine Query Language (33).

Se selecciona ya que entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Lime

En el ámbito de PHP existen muchos frameworks para crear pruebas unitarias, Symfony incluye su propio framework llamado Lime. Se basa en la librería Test: Morede Perl y es compatible con TAP, lo que significa que los resultados de las pruebas se muestran con el formato definido en el “Test Anything Protocol”, creado para facilitar la lectura de los resultados de las pruebas (34).

Lime proporciona soporte para las pruebas unitarias y tiene las siguientes ventajas (34):

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas.
- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. Los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- Symfony utiliza Lime para sus propias pruebas y su “regression testing”, por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.
- El núcleo de Lime se valida mediante pruebas unitarias.

- Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.

1.5.3.2 Lenguajes y tecnologías del lado del cliente

Son aquellos que pueden ser directamente "asimilados" por el navegador y no necesitan un pretratamiento. Un lenguaje de lado cliente es totalmente independiente del servidor (35).

HTML 5 ¹⁸

Lenguaje de marcado de hipertexto. Cuenta con etiquetas para incluir diversos elementos, tales como, imágenes y scripts de lenguajes JavaScript y CSS¹⁹. Es el lenguaje que se emplea para el desarrollo de páginas de internet y está constituido de elementos que el navegador interpreta y los despliega en la pantalla de acuerdo a su objetivo. Existen elementos para disponer imágenes sobre una página, hipervínculos que permiten dirigirse a otra página, listas, tablas para calcular datos (36).

Esta tecnología brinda la facilidad de que la aplicación desarrollada cargue más rápido en el navegador, al ser HTML 5 más sencillo y de código simple.

CSS 3

CSS son las siglas de Cascading Style Sheets; en español, Hojas de Estilo en Cascada. Es una tecnología que permite crear páginas web de una manera más exacta. Gracias a los CSS se tiene un mayor control de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores. Incluso se pueden definir diferentes estilos en un archivo externo a la página (35).

Java

Java es un lenguaje de programación orientado a objetos que está inspirado en la sintaxis de C++, pero su funcionamiento tiene mayor semejanza al de Small talk que a este. Es un lenguaje compilado e interpretado. Este lenguaje facilita la creación de aplicaciones distribuidas proporcionando una colección de clases para ser usadas en aplicaciones de red. El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos por lo que la fase de enlazado hace de Java además un lenguaje de alto rendimiento (37).

Java brinda las siguientes facilidades en el desarrollo de la presente investigación:

¹⁸ HTML: Hyper Text Markup Language.

¹⁹ CCS: Cascading Style Sheets.

- Rapidez de desarrollo y de mejora del módulo ya que en Java es fácil reutilizar el código de programación, por tanto, el desarrollo del módulo será rápido, pues es más rápido reutilizar objetos y sus componentes que reescribir el código desde el principio.
- Personalización del módulo a gran nivel.
- Compatibilidad con varios sistemas operativos.

JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web. JavaScript se usa generalmente en su forma del lado del cliente, implementado como parte de un navegador web y permite mejoras en la interfaz de usuario y páginas web dinámicas, aunque también presenta una forma del lado del servidor. El código JavaScript puede detectar acciones de los usuarios que HTML por sí sola no puede, ejemplo de ello son las pulsaciones de teclado (38).

Framework jQuery en su versión 1.9.0

jQuery es un framework JavaScript que ha tenido éxito debido a que está orientado para programadores con poca experiencia. Además, posee licenciamiento bajo la Licencia MIT (Licencia que permite reutilizar el software así licenciado tanto para hacer software libre como privativo), la cual posibilita utilizarlo y liberarlo bajo BSD (Licencia que mantiene la protección de los derechos de autor, pero permite la libre redistribución y modificación, permitiéndose el uso en software comercial) (39).

Este framework JavaScript, ofrece una infraestructura con la que se tendrá mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Con jQuery se obtendrá ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax (Asynchronous JavaScript And XML, técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications)). Por último, posee ventajas gráficas, es multiplataforma y brinda una gran cantidad de efectos.

jQuery brinda la facilidad en el desarrollo de la presente investigación de producir páginas dinámicas, totalmente compatibles con la mayoría de los exploradores, pues realiza los asuntos de compatibilidad de manera automática.

1.5.4 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE, por sus siglas en inglés), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir,

consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

Un IDE debe tener las siguientes características (40):

- Multiplataforma.
- Soporte para diversos lenguajes de programación.
- Integración con sistemas de control de versiones.
- Reconocimiento de sintaxis.
- Extensiones y componentes para el IDE.
- Integración con frameworks populares.
- Importar y exportar proyectos.
- Múltiples idiomas.
- Manual de usuarios y ayuda.

NetBeans en su versión 8.0:

Netbeans es una Herramienta que se utiliza para desarrollar aplicaciones web, móvil y de escritorio para diferentes lenguajes de programación como son Java, C++, Ruby y PHP, entre otros. Es de código abierto, es multiplataforma, multilenguaje, contiene servidores web y es fácil de instalar y de utilizar (41).

Características de NetBeans:

- Formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web.
- Ofrece documentación y recursos de formación exhaustivos.
- Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas.
- Es un IDE multiplataforma.
- Está desarrollado para usarse generalmente con lenguaje Java, aunque permite su uso con otros lenguajes de programación como PHP.

Fundamentos de la selección

Se decide utilizar NetBeans7.3 debido a que ofrece opciones de desarrollo con plugins y herramientas incorporadas, así como completamiento de código, facilitando el trabajo a los programadores para la creación de aplicaciones web con rapidez. Es una herramienta fácil de instalar y consume pocos recursos. El editor de código fuente que presenta es muy ágil y a la vez robusto, características que hacen que sea una excelente herramienta para el desarrollo de soluciones informáticas.

Sistema Gestor de Base de Datos

PostgreSQL en su versión 9.4 es el sistema de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de Multihilo para garantizar la estabilidad del sistema. Es un sistema de gestión de bases de datos objeto-relacional, con código fuente disponible libremente. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (42).

Posee numerosas ventajas entre ellas se pueden mencionar:

- Estabilidad y confiabilidad legendarias.
- Extensible pues el código fuente está disponible para todos sin costo.
- Multiplataforma.
- Diseñado para ambientes de alto volumen.
- Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

PgAdmin III

Es una herramienta para la administración de bases de datos para PostgreSQL. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL²⁰ simples hasta desarrollar bases de datos complejas. La aplicación incluye un editor SQL con resaltado de sintaxis, un agente para lanzar scripts programados y mucho más (43).

1.6 Servidor web

Un servidor web sirve de contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP (Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la World Wide Web)

NGINX en su versión 1.4.6

Se trata de un servidor web y proxy inverso de código abierto ligero de alto rendimiento, que también incluye servicios de correo electrónico con acceso al Internet Message Protocol (IMAP) y al servidor Post Office Protocol (POP). Además, NGINX

²⁰ SQL: Structured Query Language

está listo para ser utilizado como un proxy inverso. En este modo, NGINX se utiliza para equilibrar la carga entre los servidores back-end, o para proporcionar almacenamiento en caché para un servidor back-end lento (44).

Son muchas las características que nos ofrece este servidor web, pero una de las más importantes es que se trata de un software que es asíncrono, a diferencia de Apache que está basada en procesos. La ventaja principal de ser asíncrono, es su escalabilidad. En un sistema basado en procesos, cada conexión simultánea requiere de un hilo, lo que puede llevar a sobrecargar el servidor, mientras que en un servidor asíncrono se gestionan las peticiones en muy pocos hilos, reduciendo las posibilidades de sobrecarga en el servidor.

Otras características que ofrece el servidor NGINX son:

- Capaz de manejar más de 10.000 conexiones simultáneas con un uso bajo de memoria.
- Balanceo de carga, distribuye la carga entre los servidores que formen parte de la estructura, redirigiendo cada vez la petición hacia aquella máquina que tenga una menor carga.
- Alta tolerancia a fallos.
- Soporte para TSL, SSL, FastCGI, SCGI o uWSGI, entre otros.
- Compatible con el nuevo estándar de direcciones IPv6.
- Compresión y descompresión con Gzip, que permite comprimir al vuelo los archivos y datos que se mueven por la red, desde el servidor web hasta el navegador del usuario.
- Reescritura de urls, para crear urls amigables que nos ayuden en el proceso del posicionamiento web, aunque a diferencia de Apache, NGINX no hace uso del fichero .htaccess, sino que las reglas de reescritura las carga directamente en su configuración.
- Permite limitar el número de conexiones concurrentes.

1.7 Librerías para graficar resultados

La selección de la librería a utilizar a la hora de graficar los resultados se realiza teniendo en cuenta la información que se desea representar. Las librerías analizadas son Morris JS y jqCloud.

jqCloud en su versión 2.0.1

jqCloud es un plugin de jQuery que construye nubes de palabras o nubes de etiquetas.

Morris en su versión 2.1.0

Morris.js es una API sencilla para dibujar gráficos de barras, líneas, áreas y también diagramas circulares. Posee un estilo bastante simple, que podría ir a la perfección con sitios de estilo minimalista y diseño flat. Morris.js es un plugin de gráficos basada en jQuery creado por Olly Smith. Incluye la función de cambio de tamaño, lo que hace que los gráficos sean totalmente sensibles.

Conclusiones parciales

- Luego de realizar un estudio profundo sobre analítica visual y las técnicas de visualización de la información para obtener gráficas robustas y de fácil entendimiento se seleccionaron las técnicas nube de palabras, gráfico de barra, línea de tiempo y gráficos de tablas.
- La MD es un poderoso campo para la obtención de conocimiento a partir de la aplicación de técnicas y algoritmos. Permite determinar patrones de comportamiento en grandes volúmenes de datos.
- Las herramientas disponibles permiten automatizar gran parte de la tarea de encontrar los patrones de comportamiento ocultos en los datos y aplicar las transformaciones necesarias para esto. Dentro de estas herramientas se selecciona Weka para apoyar el proceso de MD.
- La selección de la metodología de minería de datos CRISP-DM le permite amplitud y flexibilidad al proyecto, gracias a las tareas con las que cuenta para la guía, planificación y ejecución del proyecto.
- Las metodologías, herramientas y tecnologías seleccionadas permiten la obtención de un módulo robusto, favoreciendo su construcción.

Capítulo 2 Propuesta de solución

2.1 Introducción

Durante el desarrollo de software, la correcta definición de los procesos que intervienen en este se considera de gran importancia pues de ello dependerá la comprensión del sistema que se desea implementar.

La metodología AUP define en cada una de sus fases un grupo de artefactos que brindan una comprensión clara del sistema a desarrollar, gracias a su enfoque ágil permite que el equipo elabore la documentación necesaria que facilite la comunicación entre los desarrolladores y las partes interesadas. En este capítulo se presentarán los siguientes artefactos: Modelo de dominio, la especificación de requisitos funcionales y no funcionales que tendrá la propuesta de solución, se modelarán las Historias de Usuarios (HU) y la interacción de los actores con el sistema, así como una descripción detallada de los mismos. Además, se describen las fases y tareas propuestas por CRISP-DM.

2.2 Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades (47).

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Cuando se realiza la programación orientada a objetos, el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema.

2.3 Análisis de los conceptos del dominio

Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Se identificaron los siguientes conceptos:

- **Plataforma Educativa ZERA:** destinada a apoyar el proceso de enseñanza y aprendizaje
- **Rol:** permiten definir los distintos niveles de interacción (y control) y acceso.
- **Profesor:** es la persona que estructura y organiza el proceso de aprendizaje, dirige el proceso de aprendizaje, facilita los recursos.
- **Reportes:** es el módulo dentro de la plataforma destinado a los reportes.

- **Comportamiento del estudiante:** son los reportes relacionados con la interacción del estudiante en la plataforma, uso del glosario, realización de ejercicios, acceso a recursos, realización de tareas.
- **Recurso:** módulo dentro de la Plataforma Educativa ZARA destinado a gestionar los recursos multimedia, estructurales, interactivos, materiales del docente, así como el glosario de términos para la plataforma.
- **Multimedia:** recursos multimedia a los que accede el estudiante.
- **Estructurales:** recursos estructurales a los que accede el estudiante
- **Interactivos:** recursos interactivos a los que accede el estudiante.
- **Glosario:** recurso glosario al que accede el estudiante.
- **Ejercicios:** modalidad de tarea que realiza el estudiante.
- **Recorridos dirigidos:** modalidad de tarea que realiza el estudiante.
- **Evidencias:** modalidad de tarea que realiza el estudiante.

2.4 Diagrama del modelo del dominio

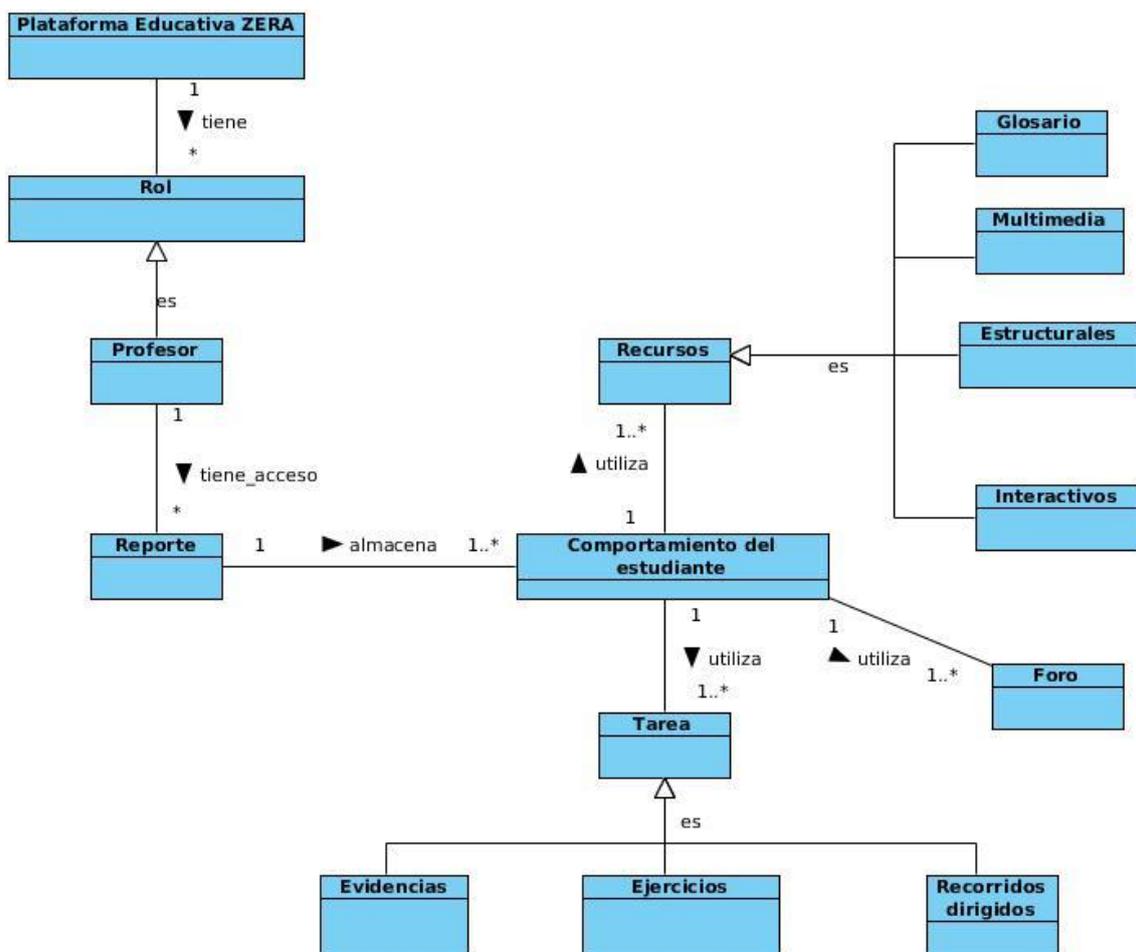


Figura 1. Modelo de dominio

2.5 Propuesta del módulo

Diversos estudios ilustran sobre la utilidad de las técnicas de visualización de información. Estos muestran que la representación visual le facilita al usuario la utilización de complejas inferencias perceptuales ya implementadas en su corteza visual. El diseño adecuado de herramientas de visualización permite simplificar la búsqueda de información, mejorar las posibilidades de detección de patrones, aumentar los recursos accesibles por los usuarios, entre otros (48).

El módulo será capaz de graficar datos almacenados en los diferentes reportes con los que cuenta la Plataforma Educativa ZERA sobre los estudiantes matriculados en un determinado curso, utilizando técnicas de analítica visual. Los usuarios típicos del sistema son: Administrador y Profesor. Para utilizar el módulo el usuario deberá autenticarse en la aplicación, esta verificará sus privilegios, permitiéndole acceder o no a las gráficas.

Para la confección del módulo fue necesario integrar el Weka con la Plataforma Educativa ZERA con el objetivo de procesar la base de datos de la plataforma y poder realizar un correcto agrupamiento de los datos a través del algoritmo K-means. El agrupamiento constituye una de la técnica descriptiva de minería de datos utilizada en la solución de la presente investigación con el objetivo de poder agrupar a estudiantes con elementos en común que describen su interacción con la Plataforma, ya sea la hora de conexión, la cantidad de visitas que realiza al glosario, la cantidad de temas que crea en el foro, el tiempo de conexión, entre otros.

La analítica visual constituye uno de los elementos fundamentales en la propuesta de solución, para el desarrollo de la misma se contó con varias técnicas para mostrar la información, ejemplo de esto lo constituye la nube de palabras o nube de etiquetas, utilizada para representar aquellas palabras visitadas en el glosario y resaltar el término con mayor índice de visitas. Se utilizó el gráfico de pastel para mostrar los recursos y la cantidad de visitas en el día. Se hizo uso de gráficos del tipo de tablas para representar el resultado del agrupamiento de estudiantes. Para representar la actividad general de los estudiantes con la Plataforma Educativa ZERA se hizo uso de gráficos de líneas y gráficos de barra.

2.5.1 Historias de Usuario (HU)

Una historia de usuario (HU) describe una funcionalidad que, por sí misma, aporta valor al usuario (49). Las HU son creadas a partir de las funcionalidades definidas. En las siguientes tablas se mostrarán algunas de las HU de los diferentes Requisitos Funcionales (RF), que presentan prioridad alta por constituir principales

funcionalidades del módulo. Para documentar las HU se utiliza una plantilla con los aspectos definidos a continuación:

- Número: posee el número asignado a la HU.
- Nombre del requisito: atributo que contiene el nombre del requisito funcional al que corresponde la HU.
- Programador: usuario encargado de desarrollar la HU.
- Prioridad: evidencia el nivel de prioridad de la HU en el negocio.
- Tiempo estimado: estimación hecha por el equipo de desarrollo del tiempo de duración de la HU.
- Tiempo real: tiempo real de desarrollo de la HU.
- Descripción: Descripción de la HU.
- Observaciones: Dependencias de la HU.

Requisitos funcionales:

- **RF1:** Visualizar palabras visitadas en el glosario.
- **RF2:** Filtrar palabras visitadas en el glosario.
- **RF3:** Visualizar actividad en la plataforma.
- **RF4:** Filtrar actividad en la plataforma dada una fecha
- **RF5:** Filtrar actividad en la plataforma dada una institución.
- **RF6:** Mostrar actividad en el módulo Recurso.
- **RF7:** Filtrar actividad en el módulo Recurso.
- **RF8:** Visualizar actividad del estudiante en las tareas.
- **RF9:** Graficar grupos de estudiante según el horario de conexión.
- **RF10:** Graficar grupos de estudiantes por la complejidad de los ejercicios resueltos.
- **RF11:** Mostrar cantidad de comentarios en el foro.
- **RF12:** Mostrar cantidad de visitas al foro.
- **RF13:** Mostrar actividad de un estudiante en el foro.
- **RF14:** Graficar grupos de estudiantes según cantidad de visitas al glosario.

A continuación, se muestra la descripción textual de la HU: Visualizar palabras visitadas en el glosario. Las restantes HU están presentes en el Anexo 1.

Tabla 2. HU: Visualizar palabras visitadas en el glosario.

Historias de Usuario	
Número: 1	Nombre del requisito: Visualizar palabras visitadas en el glosario.
Programador: Jorge Delgado Rúa	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 semanas

N/A	Tiempo Real: 2 semanas
<p>Descripción:</p> <p>1- Objetivo: Permitir que el usuario consulte las palabras visitadas en el glosario.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para visualizar las palabras visitadas en el glosario hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol profesor. - Tener los permisos para poder acceder a esta funcionalidad. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir visualizar palabras visitadas en el glosario hasta ese momento, esta acción puede realizarse seleccionando la opción Nube de palabras. - Cuando el usuario selecciona la opción Nube de palabras se muestra un gráfico con aquellas palabras o términos visitados en el glosario. 	

2.5.2 Requisitos no funcionales

Los requisitos no funcionales son los requisitos que imponen restricciones al diseño o funcionamiento del sistema software (tal como requisitos de funcionamiento, estándares de calidad, o requisitos del diseño) (50).

Para definir los aspectos no funcionales se utilizó el estándar ISO/IEC 9126. ISO/IEC 9126, que es un estándar de calidad de productos. El objetivo es abarcar todos los aspectos que pueden afectar a la calidad de los productos de software (51).

Atributos de la norma para asegurar la calidad externa e interna

RNF1. Usabilidad

- El módulo estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido.
- Serán reutilizados diseños similares a los que cuenta la Plataforma Educativa ZERA con vistas a aprovechar la experiencia de usuario.

RNF2. Apariencia o Interfaz Externa

- Consistente: las acciones similares deben ejecutarse de forma similar.

RNF3. Seguridad

- El acceso a la información debe estar restringido por usuario, contraseña y rol.

RNF4. Disponibilidad

- Los usuarios según los permisos definidos deben tener acceso en todo momento a la información que sea solicitada.

2.6 Descripción de actores del sistema

En el sistema que se está modelando actúan dos actores denominados *Profesor* y *Administrador*, que son los que interactúan con el sistema.

Tabla 3. Descripción de actores del sistema

Actor	Descripción
Profesor	Es la persona que estructura y organiza el proceso de aprendizaje, facilita los recursos. Además se encarga de crear, eliminar o leer el/los contenidos en el módulo.

2.7 Descripción del proceso de agrupamiento

El proceso de agrupamiento de minería de datos se realiza a través de la metodología CRISP-DM. A continuación, se resumen las tareas genéricas en las que se desglosan cada una de las 6 fases de dicha metodología y las salidas generadas por cada una de ellas.

- **Análisis del problema:** Fase inicial que incluye la comprensión de los objetivos y requerimientos del proyecto desde una perspectiva de negocio, con el fin de convertirlos en objetivos y en una planificación.
- **Análisis de los datos:** Recolección inicial de datos para familiarizarse con ellos, identificar su calidad y descubrir las relaciones entre los más evidentes para las primeras hipótesis de relaciones ocultas entre ellos.
- **Preparación de los datos:** Construcción de la base de datos a partir de datos primarios. Estas tareas se desarrollan en numerosas ocasiones y no de forma muy estructurada. Incluye la selección de tablas, registros y atributos, así como su transformación y preparación para las herramientas de modelado.
- **Modelado:** Se seleccionan y aplican varias técnicas de modelado.
- **Evaluación:** Una vez creado un buen modelo se debe evaluar el rendimiento del mismo y la integridad de todos los pasos sobre todo teniendo en cuenta que se han introducido todos los criterios de negocio.
- **Desarrollo o implementación.**

2.7.1 Análisis del problema

Objetivos del negocio

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el Centro de Tecnologías para la Formación (FORTES), se desarrolla la Plataforma Educativa ZERA, dirigida principalmente a los estudiantes y docentes dentro de la educación en línea. La misma almacena información referente a los estudiantes respecto a la interacción con la plataforma. En la presente investigación se pretende utilizar esta información para describir el comportamiento de los estudiantes que interactúan con la plataforma. Por lo tanto, se tiene como objetivos del negocio:

- Caracterizar el comportamiento de los estudiantes en la Plataforma Educativa ZERA.
- Obtener resultados que ayuden a los profesores a controlar y seguir a los estudiantes en el proceso de enseñanza y aprendizaje de manera individual y grupal a través de las diferentes técnicas de visualización de la información seleccionadas en el presente trabajo.

Evaluación de la situación

Recursos disponibles para la investigación:

- Personal: 2 estudiantes de 5to año de la carrera Ingeniería en Ciencias Informáticas.
- Hardware: 2 PC.
- Software: herramienta de minería de datos WEKA.
- Fuente de datos: base de datos Plataforma Educativa ZERA.
- Fecha terminación: mayo 2016.
- Calidad de los resultados: resultado fácil de entender para el usuario final.
- Seguridad: la información sólo puede ser consultada por las personas autorizadas.

Objetivos del proceso de minería de datos

- Agrupar a los estudiantes en cuanto a: acceso a recursos, actividad en el foro, realización de ejercicios y tareas, calificación, actividad en el glosario, horario de acceso a la plataforma, tiempo de trabajo, comportamiento general en la plataforma.

Plan del proyecto

En esta tarea se definen las actividades a realizar durante el proceso de descubrimiento de conocimiento.

Tabla 4. Plan de proyecto

Actividades	Fecha inicio	Fecha fin
Estudio de la base de datos de ZERA	2/03/2016	13/03/2016
Definir datos a extraer.	16/03/2016	20/03/2016
Extraer datos.	3/04/2016	3/04/2016
Seleccionar técnicas y herramientas de minería de datos	6/04/2016	10/04/2016
Pre-procesar datos.	13/04/2016	17/04/2016
Aplicar algoritmos.	20/04/2016	24/04/2016
Comprobar resultados de la aplicación de los algoritmos.	27/04/2016	30/04/2016

2.7.2 Análisis de los datos

Verificación de calidad de datos

La verificación de la confiabilidad de la información cualitativa es un elemento esencial del diseño y desarrollo del estudio que aumenta la calidad de la información recopilada. No se trata de un grupo de pruebas aplicadas a los datos, sino de verificaciones realizadas antes de empezar la recolección de datos y monitoreadas durante la investigación. Tales verificaciones, en la presente investigación incluyen:

- Cubrimiento de valores.
- Datos duplicados, es decir, existen varias identificaciones del mismo sujeto.
- Equivalencia de datos redundantes.
- Inconsistentes por definición.
- Faltantes, es decir, no existe el dato o es nulo o blanco, cuando debiera existir.

- Inválidos, cuando no cumplen alguna regla de validación; en este caso, se debe especificar la regla violada.
- Incorrectos, cuando no concuerdan con la realidad.

Los datos representan la realidad pues provienen de la interacción real de los estudiantes con la plataforma. Se detectaron campos vacíos. No existe error en los datos.

Recolectar y describir datos iniciales

La recolección de los datos se realiza utilizando la base de datos de la Plataforma Educativa ZERA. Se realiza un estudio del modelo Entidad Relación, el cual cuenta con 223 tablas. De este modelo se seleccionaron las tablas que almacenaban información de interés para la investigación. Los datos a extraer son:

Tabla 5. Datos recolectados de la base de datos de la Plataforma Educativa ZERA

Datos	Tabla
Términos visitados en el glosario.	tb_glosary
Ejercicios realizados de complejidad baja, alta o media.	tb_exercise
Duración en la realización de un ejercicio.	tb_attemp
Fecha de inicio en la realización de un ejercicio.	tb_attemp
Fecha fin en la realización de un ejercicio.	tb_attemp
Número de recursos visitados por tipos.	tb_trace
Número de veces que visita el glosario de términos.	tb_trace
Ejercicios realizados prácticos.	tb_practice
Evaluación.	tb_practice
Fecha de acceso a la plataforma.	tb_session
Fecha en la que abandona la plataforma.	tb_session
Recurso más visitado.	tb_trace
Momento dentro de 24	tb_session

horas en el que más se visita la plataforma.	
Tiempo de conexión en la plataforma por un estudiante.	tb_session
Término más visitado en el glosario.	tb_glosary
Tipo de ejercicio que más realiza un estudiante según su complejidad.	tb_exercise

2.7.3 Preparación de los datos

Seleccionar los datos

En esta tarea se debe decidir qué datos serán usados para el análisis. Los criterios incluyen la importancia a los objetivos de la minería de datos, la calidad, y las restricciones técnicas como límites sobre el volumen de datos o los tipos de datos. Note que la selección de datos cubre la selección de atributos (columnas) así como la selección de registros (filas) en una tabla. El objetivo de esta tarea, es detallar los atributos que serán incluidos o excluidos del proceso de minería de datos. La selección se realizó tomando como referencia los atributos que se utilizan en investigaciones anteriores.

Un glosario, por ejemplo, es una relación de términos explicados y comentados para aquellas personas que desconocen su significado (52) . El uso del mismo permite determinar características de colaboración del estudiante, ya que en *e-learning* se pueden crear glosarios colaborativos, el cual sirve de foco para la colaboración de los alumnos en un curso. Los datos usados para realizar el análisis son: número de palabras que el usuario ha visitado, número de palabras subidas por el usuario, número de aportaciones revisadas, por lo tanto públicas, y bien valoradas o calificadas por sus compañeros/as, palabras más visitadas en el glosario.

La interacción de los usuarios con las plataformas produce una gran cantidad de datos que comprenden tiempo de acceso, materiales visitados, resultados de la actividad académica (resultado de ejercicios, notas de pruebas, etc.) y datos personales.

El uso de las plataformas educativas facilita la administración del aprendizaje en línea, proveyendo un mecanismo de seguimiento del estudiante, su evaluación, acceso a recursos, tiempo de acceso, materiales visitados, resultados de la actividad académica y datos personales (53).

La tabla que se muestra a continuación muestra una selección de atributos propuestos en otras investigaciones sobre minería de datos en plataformas *e-learning* (54), (55).

Tabla 6. Información útil de la interacción de los alumnos con los LMS

<i>Actividad</i>	<i>Descripción</i>
Participación en foros	Número de mensajes leídos por el estudiante en el foro de noticias del curso virtual.
	Número de respuestas dadas por el estudiante en el foro de noticias del curso virtual.
Tiempo de teoría	Tiempo total empleado en componentes teóricos de los contenidos.
Tiempo de tareas	Tiempo total empleado en tareas de enseñanza.
Palabras en glosarios	Número de palabras publicadas en el glosario.
	Número de palabras revisadas en el glosario.
Páginas visitadas	Páginas revisadas.
Recurso visitado	Nombre del recurso más visitado.

Limpiar, estructurar e integrar datos

Con el fin de construir una vista minable se les realizó un pre-procesamiento a los datos utilizando técnicas implementadas en la herramienta de análisis de datos seleccionada.

El tratamiento a los datos fue el reemplazo de valores perdidos. Un valor que falta puede significar una serie de cosas diferentes en sus datos. Tal vez los datos no estaban disponibles o no aplicables o el evento no ocurrió. Podría ser que la persona que haya introducido los datos no sabía el valor correcto o no contestado (56).

En ocasiones se hace necesario realizar modificaciones principalmente sintácticas sobre los datos a ser utilizados para el análisis, pues podría ser requerido por la herramienta de modelado. WEKA trabaja con datos provenientes de bases de datos, archivos y datos que residen en servidores de Internet. En este caso los datos fueron dados a la herramienta a través de un archivo CSV.

2.7.4 Modelado

Selección de una técnica de modelado

La técnica de minería de datos seleccionada fue el agrupamiento, ejecutando varios algoritmos con el fin de comparar los resultados que se obtienen con cada uno de ellos. Se compararán tres algoritmos, DBSCAN, K-means y EM, ofreciéndose los resultados que se obtienen al aplicar cada uno de ellos en el proceso de agrupamiento.

Para seleccionar el algoritmo de agrupamiento fue necesario encontrar una manera de validar la calidad de las particiones.

La validación del agrupamiento, siempre ha sido reconocida como una de las cuestiones esenciales para el éxito de las aplicaciones de esta técnica. La validación en el agrupamiento tiene dos categorías principales: externa e interna. La validación externa utiliza información no presente en los datos mientras que la validación interna mide el agrupamiento basada en la información de los datos (57).

Las medidas de validación interna pueden ser utilizadas para elegir el mejor algoritmo de agrupamiento, así como el número óptimo de clúster sin ningún tipo de información adicional, siendo la más conveniente utilizar en la investigación. En el presente trabajo se utilizó la medida de validación interna Coeficiente de Silhouette el cual establece una relación entre la cohesión y separación de un agrupamiento. El Coeficiente de Silhouette para un punto x del conjunto de datos está definido como:

$$S(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

Siendo:

$a(x)$: distancia promedio de x a todos los demás puntos en el mismo clúster.

$b(x)$: distancia promedio de x a todos los demás puntos en el clúster más cercano.

Donde el valor de $s(x)$ puede variar entre -1 y 1:

- -1: mal agrupamiento.
- 0: indiferente.

- 1: bueno.

El Coeficiente de Silhouette para todo el agrupamiento es:

$$SC = \sum_{i=1}^N S(x), \text{ siendo } N \text{ la cantidad de clúster.}$$

Para aplicar esta medida al resultado de la ejecución de los algoritmos seleccionados fue utilizada la herramienta MOA debido a la complejidad de aplicar el coeficiente de forma manual, sobre una muestra de 500 estudiantes. La misma muestra fue utilizada en la herramienta WEKA para analizar el tiempo de ejecución. La siguiente tabla muestra los valores del Coeficiente de Silhouette correspondiente a cada algoritmo analizado y el tiempo de ejecución en construir el modelo, siendo este último otro factor a tener en cuenta para la elección del algoritmo, debido a que uno de los objetivos de la investigación es propiciar el análisis rápido de la información.

Tabla 7. Coeficiente de Silhouette y tiempo de ejecución.

Algoritmo	Coeficiente de Silhouette	Tiempo de ejecución
K-means	0,932736059	0.01
DBSCAN	0,913176295	0.04
OPTICS	0,913276295	0.32

Al analizar los resultados obtenidos, se evidencia que en cuanto al Coeficiente de Silhouette el K-means posee el valor más alto y a la vez, es el que menor tiempo de ejecución posee, por tanto, se elige este algoritmo para el proceso de agrupamiento.

Construir el modelo

En esta etapa se ejecuta el algoritmo K-means sobre los datos seleccionados, el cual necesita como uno de sus parámetros iniciales el número de clúster. Para seleccionar este parámetro se tuvieron en cuenta dos criterios:

- Ejecutar un algoritmo que determine de forma automática el número de clúster.
- Ejecutar el K-means con diferente número de clúster y comparar para cada ejecución el Coeficiente de Silhouette.

Para el primer criterio se seleccionó el EM, resultando 4 el valor de k. En el segundo criterio los resultados más favorables se obtuvieron también con k =4. La siguiente tabla muestra la ejecución del K-means con diferentes números de clúster:

Tabla 8. Ejecución del K-means con diferente número de k

Número de clúster	Coefficiente de Silhouette
2	0,92339988
3	0,922052239
4	0,932736059
5	0,931440664
6	0,928823812
7	0,817197542

Una vez seleccionado el número de clúster se ejecutó el K-means con diferente número de semilla sobre los datos relacionados a la realización de ejercicios. El número de semillas seleccionado fue 7, pues con este valor se obtiene el menor error cuadrado y el menor tiempo de ejecución.

Tabla 9. Ejecución del K-means con diferente número de semilla

Semillas	Suma del error cuadrado	Tiempo promedio en construir el modelo
3	51.61621641434076	1.89
4	51.59226097208202	1.31
5	51.61621641434076	1.9
6	51.43845527003686	1.26
7	51.32339789519533	1.09
8	51.59325800074741	1.91
9	51.61621641434075	1.96
10	51.59712700442065	1.14
11	51.61621641434075	1.76

12	51.32408628515912	1.11
----	-------------------	------

2.7.5 Desarrollo o implementación

Se propone la implementación de un módulo que permita a los profesores darles control y seguimiento de manera individual y grupal a los estudiantes durante su interacción con la Plataforma Educativa ZERA. El mismo debe ser capaz de agrupar a los estudiantes de acuerdo a la actividad en el glosario, acceso a recursos, realización de tareas, realización de ejercicios y comportamiento general en la plataforma, haciendo uso de técnicas de agrupamiento de minería de datos.

Para realizar estas funcionalidades se seleccionaron un conjunto de datos de la interacción de los estudiantes con la plataforma y de los resultados que obtienen en las evaluaciones que realizan. A los datos se le aplica el algoritmo de agrupamiento K-means. La ejecución del algoritmo se realiza con la herramienta de modelado de datos WEKA.

El resultado de la ejecución del algoritmo será mostrado a través de un gráfico haciendo uso de las librerías jqCloud.js y Morris.js. Además, el profesor tendrá la opción de visualizar los datos de los estudiantes pertenecientes a cada grupo, un resumen del rango de valores que toman los atributos por grupo, una comparación de atributos por grupo y la relación entre atributos. Para guiar el proceso de descubrimiento de conocimiento se hace uso de la metodología CRISP-DM. Para mostrar el resultado del procesamiento en ZERA es necesario integrarla con WEKA.

Integración ZERA-WEKA

Para la integración se requiere conectar WEKA con la base de datos de ZERA. Para ello se necesita:

- Crear un fichero CSV y convertirlo al formato arff, para pasarlo a WEKA por un comando.
- Configurar los parámetros de conexión entre WEKA y el SGBD en el fichero DatabaseUtils.props.

WEKA posee una Interfaz de Línea de Comando (CLI) que permite hacer llamadas a las clases Java que tiene definidas.

PHP dispone de la función `exec()` que se encarga de ejecutar un comando en el shell del sistema operativo, dicha función recibe como parámetros el comando que será ejecutado, un arreglo con el valor de la ejecución del comando y una variable que

almacenará el estado del retorno del comando. Los dos últimos parámetros son opcionales.

Haciendo una llamada a través de la función `exec ()` al shell del sistema y ejecutando comandos de la CLI de WEKA se integran el lenguaje PHP en el que está implementado ZERA y el lenguaje Java de WEKA.

Conclusiones parciales

- La documentación que se obtiene permite un mejor entendimiento del proceso de negocio, a partir del cual se identifican las funcionalidades necesarias para el desarrollo del módulo propuesto.
- CRISP-DM permite definir los objetivos de minería de datos teniendo en cuenta el negocio, la extracción de los datos a procesar, la elección de la herramienta y de los algoritmos de minería de datos.
- Las evaluaciones de los resultados permiten elegir al K-means como algoritmo de agrupamiento.
- El desarrollo de estos elementos servirá como punto de partida para la ejecución de la etapa de modelado del sistema.

Capítulo 3 Desarrollo de la propuesta de solución

3.1 Introducción

El modelado de la metodología AUP comprende las fases de modelado del negocio, requisitos, análisis y diseño. Este modelado es colaborativo, generando los artefactos necesarios para la comprensión del sistema con un mayor nivel de abstracción y una vez obtenidos dichos elementos se da paso a la implementación y las pruebas. En el presente capítulo se desarrollará la propuesta de solución, donde se elaborará el modelado de diseño del sistema debido a que el equipo de trabajo y el cliente poseen una total claridad de los requerimientos, por lo que se generarán los principales diagramas, con esto se implementará el sistema y luego se someterá a evaluación para validarlo.

3.2 Patrón de arquitectura

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos.

Un patrón de diseño puede considerarse como un documento que define una estructura de clases que aborda una situación particular (58).

El patrón MVC (Modelo-Vista-Controladores) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos:

- Modelo: los datos utilizados en la aplicación.
- Vista: cómo se representan los datos al usuario.
- Controlador: cómo se procesa la información en la interfaz del usuario.

3.3 Patrón de diseño

En principio, puede pensarse en un patrón como una manera especialmente inteligente e intuitiva de resolver una clase de problema en particular de modo que la solución puede reutilizarse en diferentes configuraciones. Los patrones de diseño son soluciones probadas a problemas comunes en el desarrollo de software y su aplicación facilita el trabajo en el momento de implementar una aplicación (59).

Patrones GRASP²¹:

- **Bajo Acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas (60). Acoplamiento bajo significa que una clase no depende de muchas clases y se evidencia en el hecho de que los controladores heredan únicamente de la clase `sfProjectConfiguration`.
- **Experto:** Este patrón se basa en la asignación de responsabilidades, las cuales se asignan a las clases que posean la información necesaria para llevarlas a cabo. Es utilizado en la capa de abstracción del modelo de datos. Con el uso del ORM Doctrine, Symfony genera automáticamente las clases que representan las entidades de nuestro modelo de datos. Asociado a cada una de estas clases son generadas un conjunto de funcionalidades que las relacionan de forma directa con la entidad que representan. Estas clases contienen toda la información necesaria de la tabla que representan en la base de datos (61).
- **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento, por lo que es utilizado en los controladores, en ellos se encuentran las acciones definidas para el sistema (60).
- **Controlador:** Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación (60). Dentro del framework el patrón se evidencia en las clases que forman la capa Controlador del patrón arquitectónico MVC, entre las que figuran `sfApplicationConfiguration`, `sfDoctrineRecord` y `sfPluginConfiguration`. En Symfony todas las peticiones son procesadas por un solo controlador frontal, este es el único punto de entrada de una aplicación en un entorno determinado.

3.4 Diagramas de clases del análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos del dominio del problema. Representan las cosas del mundo real no de la implementación automatizada. En general se siguen directrices muy parecidas a las

²¹ GRASP: *patrones generales de software para asignación de responsabilidades*, es el acrónimo de "GRASP (object-oriented design General Responsibility Assignment Software Patterns).

que se usan en la construcción del modelo conceptual, mostrando las clases participantes que se clasifican en tres tipos: interfaz, controladoras y entidades (62).

Clase Interfaz: se utilizan para modelar la interacción entre el sistema y sus actores, cada interfaz debe asociarse con al menos un actor y viceversa.

Clase Control: representan coordinación, secuencia, transacciones y control de otros objetos.

Clase Entidad: se utilizan para modelar la información que posee una larga vida y que es a menudo persistente.

Seguidamente se muestra el diagrama de clase del análisis correspondiente a la HU Visualizar palabras en el glosario, el resto de los diagramas se encuentran en el Anexo 2.

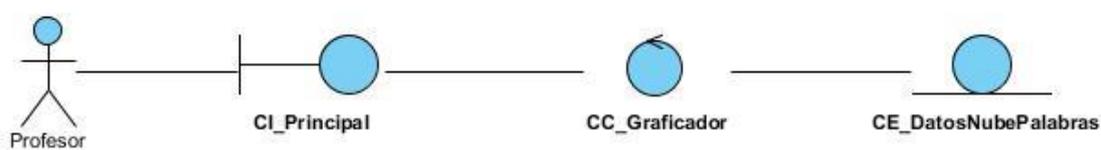


Figura 2. Diagrama de clases del análisis: HU Visualizar palabras visitadas en el glosario.

3.5 Diagramas de colaboración del análisis

Los diagramas de colaboración permiten observar adecuadamente la interacción de un objeto con respecto de los demás. Describen la perspectiva estática del sistema dado por las relaciones existentes entre los objetos y la vista dinámica de la interacción indicada por el envío de mensajes a través de los enlaces existentes entre los objetos (63).

A continuación, se muestra el diagrama de colaboración de la HU correspondiente al requisito funcional Visualizar palabras visitadas en el glosario, el resto de los diagramas se encuentran en el Anexo 3.

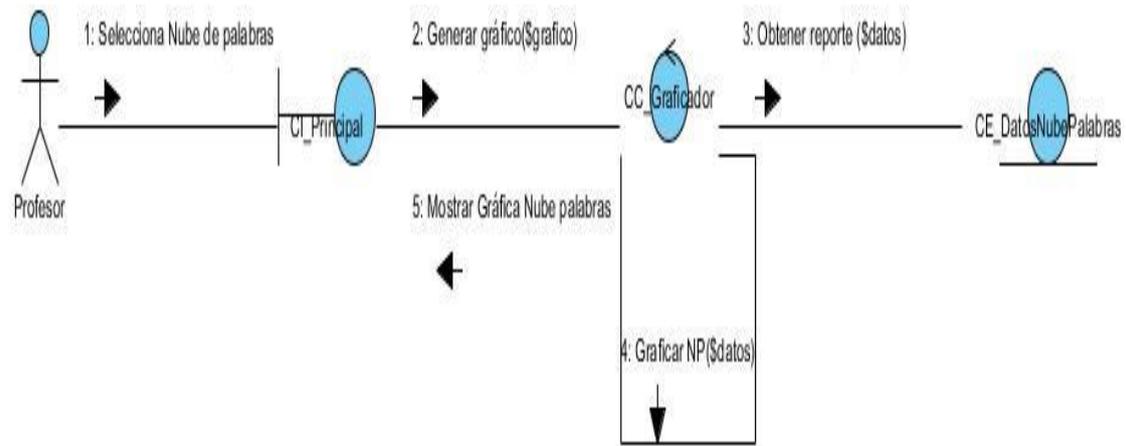


Figura 3. Diagrama de colaboración: HU Visualizar palabras visitadas en el glosario.

3.5.1 Diagramas clases del diseño

Los diagramas de clases del diseño son artefactos que especifican la estructura de clases de un sistema, así como sus relaciones. Definen de forma correcta las relaciones de dependencia, generalización y asociación de las clases que constituyen el sistema.

En la Figuras 4 se representa el diagrama de clases del diseño correspondiente a la HU Visualizar palabras visitadas en el glosario.

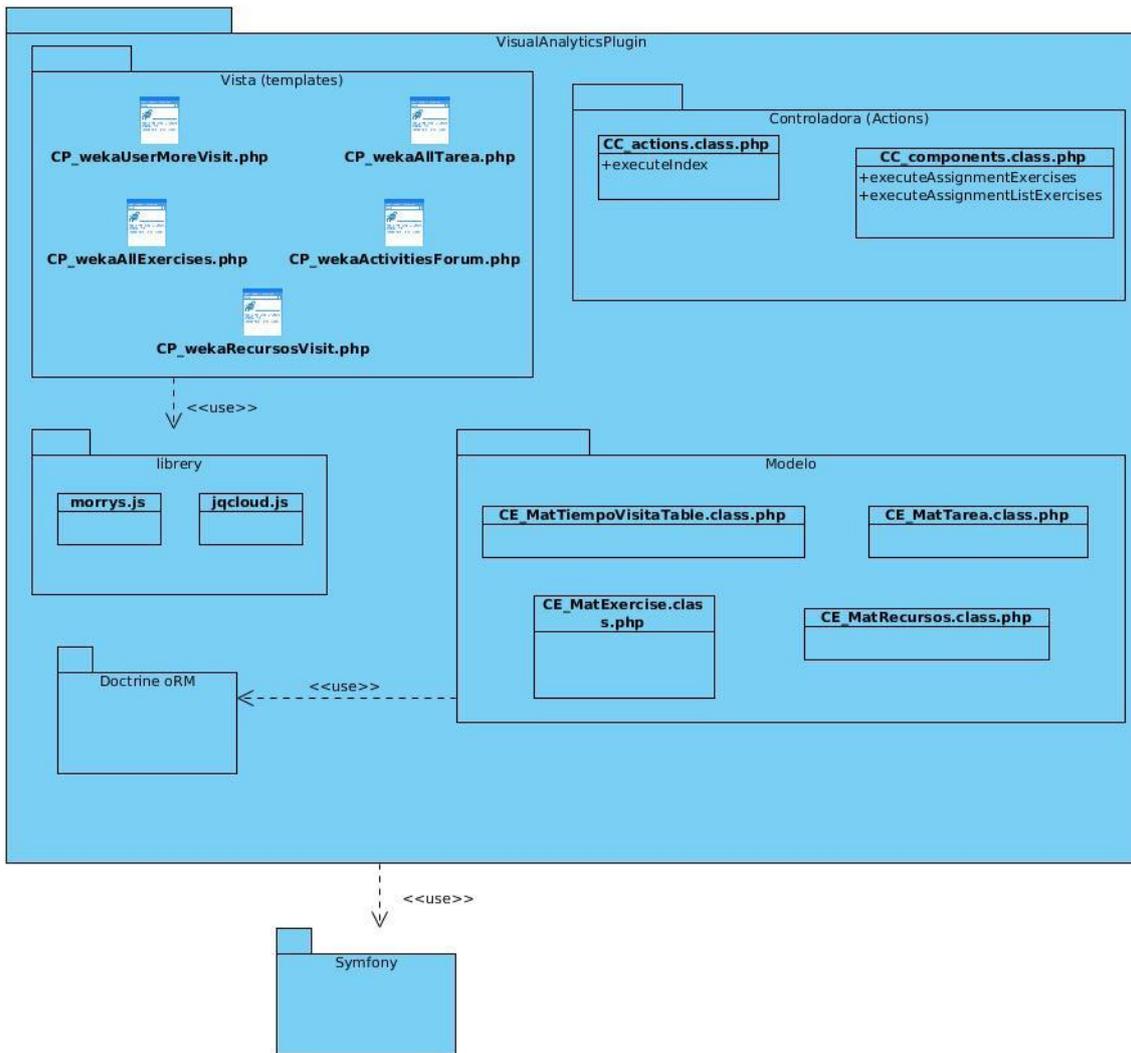


Figura 4. Diagrama de Clases del Diseño.

3.5.2 Diagramas de secuencia

En el flujo de diseño se utilizan los diagramas de interacción, estos incluyen las interacciones entre las clases del diseño a través de mensajes, que describen las operaciones que realiza cada clase para colaborar con otras con el objetivo de dar cumplimiento a la petición del usuario (64).

A continuación, se muestra el diagrama de secuencia correspondiente a la HU Visualizar palabras visitadas en el glosario, el resto de los diagramas se encuentran en el Anexo 4.

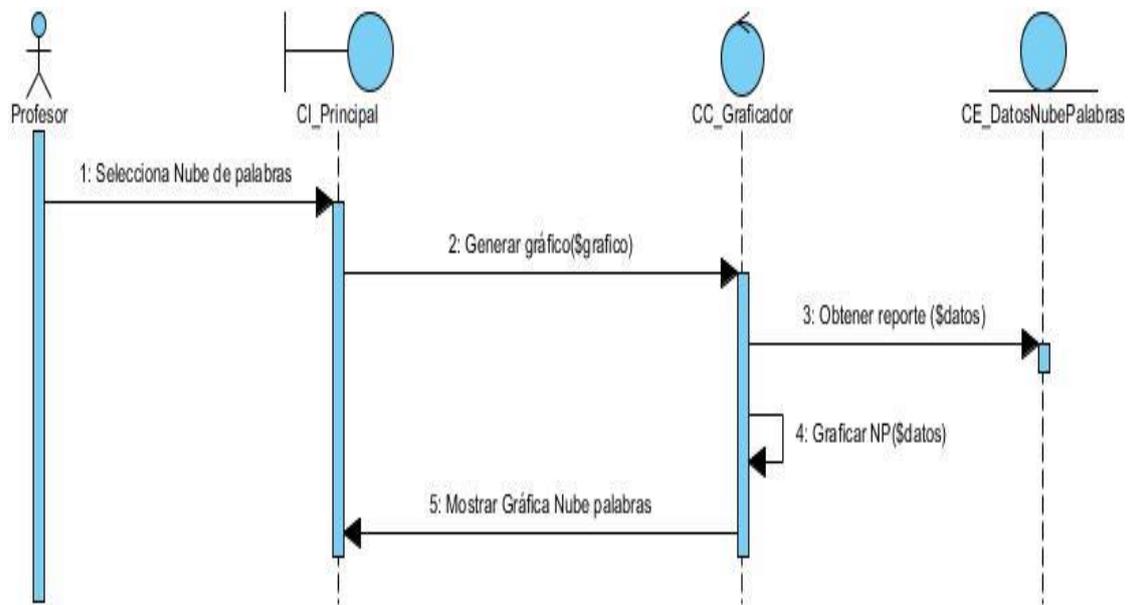


Figura 5. Diagrama de Secuencia: HU Filtrar palabras visitadas en el glosario.

3.5.3 Diagrama de componentes

Un diagrama de componentes muestra los elementos de un diseño de un sistema de software. Permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces (65).

Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

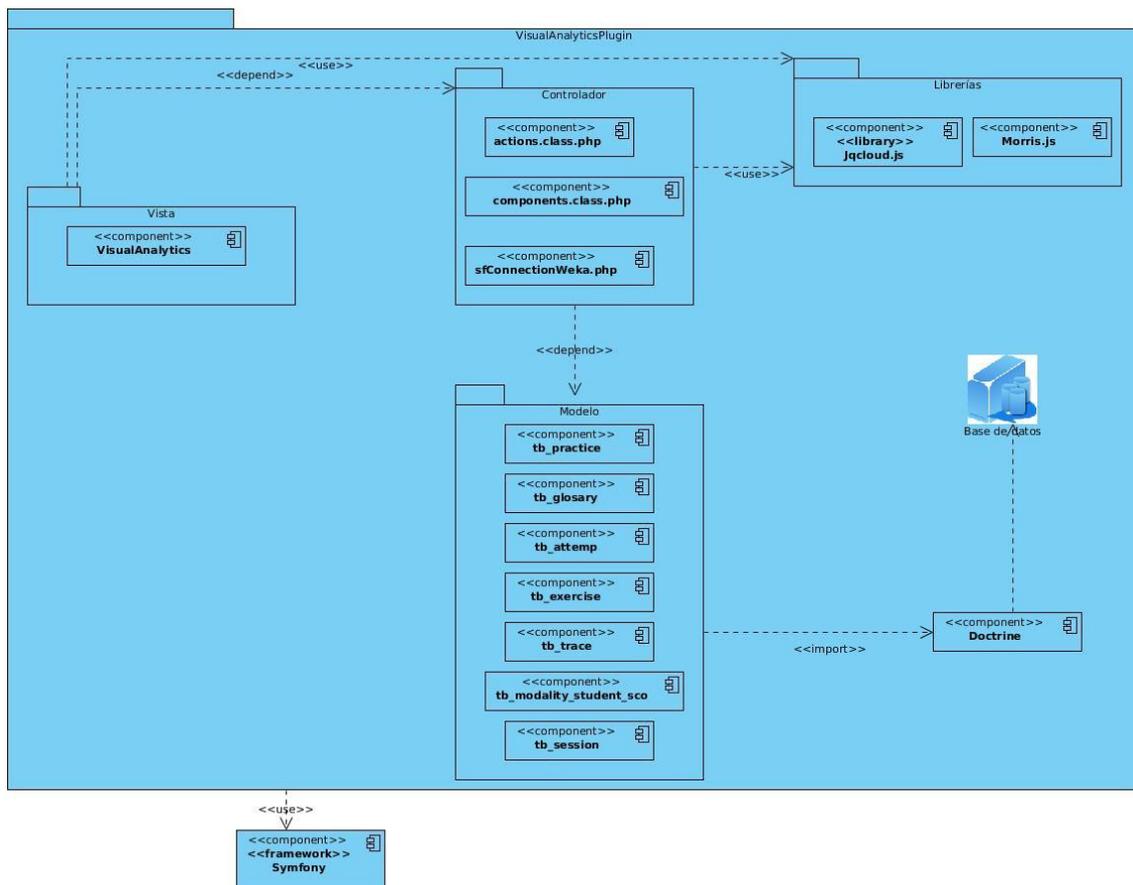


Figura 9. Diagrama de componentes.

3.6 Modelado de los datos

Otra de las etapas del diseño es la elaboración del modelo de datos, con el propósito de garantizar que los datos persistentes sean almacenados coherente y eficazmente y definir el comportamiento que debe ser implementado en la base de datos (66).

El modelo de datos correspondiente al módulo que se desarrolla está compuesto por 20 tablas. A continuación se enuncian algunas tablas correspondientes al módulo.



Figura 10. Modelado de datos.

3.7 Descripción de las tablas de la base de datos

La recolección de los datos se realiza utilizando la base de datos de la Plataforma Educativa ZERA. Se realiza un estudio del modelo Entidad Relación, el cual cuenta con 223 tablas. De este modelo se seleccionaron las tablas que almacenaban información de interés para la investigación, trabajando finalmente con 31. A continuación se describirán las tablas `tb_practice` y `tb_glosary`, el resto de las tablas se describen en el Anexo 5.

Tabla 10. Descripción de la tabla `tb_practice` de la base de datos

<i>tb_practice</i>		
Descripción: en la siguiente tabla se agrupa la información correspondiente a los datos de los ejercicios realizados por un estudiante.		
Atributo	Tipo	Descripción

id	bigint	Etiqueta única que identifica al objeto en la tabla.
evaluation	carácter varying (128)	Almacena la evaluación del ejercicio.
percent	double precision	Almacena el por ciento del ejercicio realizado.
exercise_id	bigint	Almacena el identificador único del ejercicio.
study_program_id	bigint	Almacena el identificador único del programa de estudio.
student_id	bigint	Almacena el identificador único del estudiante.
way_to_do_id	bigint	Almacena el identificador único del camino en la realización del ejercicio.
begin_date	Timestamp without time zone	Almacena la fecha de inicio de la realización del ejercicio.
end_date	Timestamp without time zone	Almacena la fecha de fin de la realización del ejercicio.
node_id	Carácter varying(128)	Almacena el identificador único del nodo.
created_at	Timestamp without time zone	Almacena la fecha de creación.
updated_at	Timestamp without time zone	Almacena la fecha de

	zone	actualización.
deleted_at	Timestamp without time zone	Almacena la fecha de eliminación.

Tabla 11. Descripción de la tabla tb_glosary de la base de datos

tb_glosary		
Descripción: en la siguiente tabla se agrupa la información correspondiente al glosario, donde se almacenan los términos visitados por los estudiantes.		
Atributo	Tipo	Descripción
id	bigint	Etiqueta única que identifica al objeto en la tabla.
term	character varying(128)	Almacena el término visitado
node_id	character varying(128)	Almacena el identificador único del nodo.
created_at	timestamp without time zone	Almacena la fecha de creación.
updated_at	timestamp without time zone	Almacena la fecha de actualización.
deleted_at	timestamp without time zone	Almacena la fecha de eliminación.

3.8 Pruebas de software

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se

usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba (67).

Al ejecutarse esta actividad se persigue descubrir defectos en el sistema que hagan que este tenga un comportamiento incorrecto o no deseable.

3.8.1 Niveles de prueba

El proceso de pruebas se realiza en varios niveles, estos se enfocan a determinados objetivos y están estrechamente relacionados con los tipos de pruebas. En cada uno de estos niveles de prueba, se podrán ejecutar diferentes tipos de prueba tales como: pruebas funcionales, no funcionales, de arquitectura y asociadas al cambio de los productos (68).

A continuación, una breve descripción de cada nivel de prueba (68):

- **Pruebas unitarias:** este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.
- **Pruebas de integración:** este tipo de pruebas son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.
- **Pruebas de sistema:** este tipo de pruebas deben ser ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo, una buena práctica en este punto corresponde a la tercerización de esta responsabilidad. La obligación de este equipo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto.

Para la ejecución de las actividades de prueba no se aplicarán las pruebas de integración debido a que al sistema no se le incorporarán componentes o módulos desarrollados independientemente de la solución. Serán aplicadas las pruebas unitarias para poder validar las funcionalidades con las que cuenta el sistema mediante Lime. Se ejecutarán las pruebas de sistema una

vez que esté concluido, para verificar que cumple con los requerimientos funcionales y no funcionales.

3.8.2 Métodos de prueba

Las pruebas son de gran importancia en la garantía de la calidad del software. Los objetivos principales de realizar una prueba son:

- Detectar un error
- Tener un buen caso de prueba
- Descubrir un error no descubierto antes

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo.

3.8.3 Diseño de casos de prueba (CP)

Se trata de diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y de tiempo. Cualquier producto de ingeniería se puede probar de dos formas (69):

- **Pruebas de caja negra:** Realizar pruebas de forma que se compruebe que cada función es operativa.
- **Pruebas de caja blanca:** Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

A continuación, se muestra el diseño de caso de prueba correspondiente a la HU Visualizar palabras visitadas en el glosario. El resto de los casos de pruebas se encuentran en el Anexo 6.

Tabla 12. Diseño del caso de prueba de la UH Visualizar palabras visitadas en el glosario

CP Visualizar palabras visitadas en el glosario			
Descripción general			
Permitir que el usuario consulte las palabras visitadas en el glosario.			
Condiciones de ejecución			
-Estar autenticado en el sistema con el rol profesor.			
-Tener los permisos para poder acceder a esta funcionalidad.			
SC 1 Visualizar palabras visitadas en el glosario			
Escenario	Descripción	Respuesta	del Flujo

		<i>sistema</i>	<i>central</i>
EC 1 Visualizar palabras visitadas en el glosario	Selecciona la opción Nube de palabras y consulta un gráfico con aquellas palabras visitadas en el glosario	Muestra las palabras visitadas en el glosario de términos. Permite además observar las palabras en diferentes tamaños según cantidad de consultas realizadas.	Módulo Analítica Visual/Nube de palabras

3.8.3.1 Resultados de las pruebas de caja negra

En la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta (69). Para la realización de las pruebas del sistema se aplicó esta técnica sobre las interfaces gráficas del sistema, con la ejecución de las mismas se pudo constatar el cumplimiento de los requisitos funcionales y no funcionales del módulo obtenido. A continuación, se presenta el resultado de las pruebas realizadas:

Tabla 13. Resultados de las pruebas de caja negra

<i>Iteraciones</i>	<i>Cantidad de casos de prueba</i>	<i>No conformidades detectadas</i>			
		<i>Alta</i>	<i>Media</i>	<i>Baja</i>	<i>Total</i>
1	14	7	5	2	14
2	14	3	3	1	7
3	14	2	2	1	5

3.8.4 Pruebas unitarias

Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Teniendo en cuenta esto se hizo uso de Lime para realizar las pruebas. A continuación, se muestra el resultado obtenido de la ejecución de las pruebas unitarias utilizando Lime:

```
flako@flako-pc ~/Escritorio/tesis_2015_2016 $ php symfony test:unit sfConnectionWeka
1..3
ok 1
ok 2
not ok 3
# Failed test (./test/unit/sfConnectionWekaTest.php at line 27)
# Looks like you failed 1 tests of 3.
flako@flako-pc ~/Escritorio/tesis_2015_2016 $
```

Figura 6. Resultados fallidos en la ejecución de Lime

```
eka.php on line 131
java.io.IOException: No source has been specified
    at weka.core.converters.CSVLoader.getStructure(CSVLoader.java:962)
    at weka.core.converters.AbstractFileLoader.runFileLoader(AbstractFileLoader.java:332)
    at weka.core.converters.CSVLoader.main(CSVLoader.java:220)
ok 3
# Looks like everything went fine.
flako@flako-pc ~/Escritorio/tesis_2015_2016 $
```

Figura 7. Resultados correctos en la ejecución de Lime

Conclusiones parciales

- El diseño propuesto, a partir del uso del patrón Modelo-Vista-Controlador, permitió una mejor comprensión de la estructura del módulo.
- Para la validación del software se realizaron las pruebas al sistema mediante la técnica de caja negra, con esta se detectaron a tiempo un grupo de no conformidades en cada iteración, las cuales fueron resueltas al final de cada una.

Conclusiones generales

Con el desarrollo del módulo se arriba a las siguientes conclusiones:

- El marco teórico elaborado permitió fundamentar y elaborar teóricamente los conceptos relacionados con la investigación.
- La herramienta WEKA y la metodología CRISP-DM permitieron llevar a cabo el procesamiento de los datos.
- El estudio de los elementos asociados al tema de tesis, haciendo uso de métodos científicos, permitió definir el agrupamiento como técnica de minería de datos a utilizar.
- Se realizó un estudio sobre analítica visual y el uso de la misma en LMS a nivel nacional e internacional, evidenciándose la necesidad de crear una solución que cumpla con las técnicas más novedosas de analítica visual.
- Se desarrolló el módulo Analítica Visual para la Plataforma Educativa ZERA, obteniéndose como resultado el desarrollo de las funcionalidades que dan solución a los requisitos definidos.
- La realización de pruebas permitió comprobar el correcto funcionamiento del módulo.

Recomendaciones

Para el mejoramiento de este trabajo es importante optimizar un grupo de elementos que no fueron tomados en cuenta por cuestiones de tiempo, dentro de ellos se señalan los siguientes:

- Establecer nuevas técnicas de visualización de la información que aún no se han tratado.
- Aplicar técnicas que permitan realizar recomendaciones a los alumnos durante su interacción con la plataforma para poder mejorar su aprendizaje.

Bibliografía

1. *Estilos de aprendizaje y e-learning. Hacia un mayor rendimiento académico.* **Martínez Caro, Eva y Gallego Rodríguez, Alejandrino.** 7, s.l. : RED: Revista de Educación a Distancia, 2003. ISSN-e 1578-7680.
2. *Estado actual de los sistemas e-learning .* **García Peñalvo, Francisco José.** 2, Salamanca : Ediciones Universidad de Salamanca, 2005, Vol. 6. ISSN-e 1138-9737.
3. *Agrupamiento de datos.* **Monteserin, Ariel.** 2009.
4. *Data Mining. Concepts, models, method and algorithms.* **Kantardzic, Mehmed.** s.l. : IEEE press, 2003.
5. *Algoritmos TDIDT aplicados a la Minería de Datos Inteligente.* **Serventes, M.** Universidad de Buenos Aires: Facultad de Ingeniería, Laboratorio de Sistemas Inteligentes : s.n., 2008.
6. *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* s.l. : Brief, Issue, April 10, 2012.
7. **Padhraic, Smyth, Heikki, Mannila y David, Hand.***Principles of Data Mining.* s.l. : The MIT Press, 2008.
8. **Hall, Mark A, Frank, Eibe y Witten, Ian H.***Data Mining Practical Machine Learning Tools and Techniques.* Tercera edición. 2011.
9. **XU, Rui y WUNSCH, DONALD C.***Clustering. IEEE Press Series on Computational Intelligence.* [sin fecha]. ISBN 978-0-470-27680-8.
10. **SÁNCHEZ, S., PLA, F. y PASCUAL, D.***Algoritmos de agrupamiento.*
11. **Ian H, Frank, Witten y Eibe.***Data Mining: Practical machine learning tools with Java implementation.* San Francisco : s.n., 2011.

12. **Pérez Aramillo, Miguel Antonio y Cardoso García, Yanet.** La Habana: Universidad de las Ciencias Informáticas : s.n.
13. *Sistemas Inteligentes de Gestión. Guión de Prácticas de Minería de Datos. Práctica 1. Herramientas de Minería de Datos.* **Berzal, Fernando y Cubero, Juan Carlos.** Granada: Dpto de Ciencias de la Computación en I.A : s.n.
14. moa.cms.waikato.ac.nz. [En línea] 2010. <http://moa.cms.waikato.ac.nz/>.
15. **Cigliuti, P., Pollo-Cattaneo, F. y García-Martínez, R.** PROCESOS DE IDENTIFICACIÓN DE COMPORTAMIENTO DE COMUNIDADES EDUACTIVAS CENTRADAS EN EVEAs. Buenos Aires : s.n., 2016.
16. **Samaniego Erazo, Gonzalo y Marqués Molías, Luis.** Patrones de uso docente de EVAs en la Escuela de Ingeniería en Sistemas Informáticos de la ESPOCH . Chimborazo : s.n., 2015.
17. **Núñez Cardenas, Felipe de Jesús, y otros.** Identificación de estilos de aprendizaje en alumnos universitarios de computación de la Huasteca Hidalguense mediante técnicas de minería de datos.
18. **Gómez Aguilar, Diego Alonso; García Peñalvo, Francisco José; Therón, Roberto.** El profesional de la información. [En línea] 2014. <http://www.elprofesionaldelainformacion.com/contenidos/2014/mayo/03.pdf>.
19. **Sociedad Latinoamericana para la Calidad.** Educar Chile. [En línea] 2000. ww2.educarchile.cl/UserFiles/P0001/File/radar.pdf.
20. *Visualización de Grafos Web.* **Medrano, José Federico, Alonso Berrocal, José Luis y G. Figuerola, Carlos.** Salamanca : s.n., 2000.
21. *Visualización y Recuperación de Información.* **Hassan Montero, Yusef.** Granada : s.n., 2006.
22. **Hermenegildo, Romero.** <http://es.slideshare.net>. [En línea] 7 de Febrero de 2012. [Citado el: 20 de Enero de 2015.] <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>.

23. IBM. [En línea] 22 de Noviembre de 2010. [Citado el: 22 de Enero de 2016.] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa>.
24. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. **Letelier, Patricio y Penadés, Ma Carmen**. 26, Valencia : s.n., 2006, Vol. 5. ISSN 1666-1680.
25. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education, 2000. 84-7829-036-2.
26. docs.kde.org. [En línea] [Citado el: 23 de Enero de 2016.] <https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-basics.html>.
27. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady**. *The Unified Modeling Language Reference Manual Second Edition*. Boston : Pearson Education, 2004. ISBN 0-321-24562-8.
28. *Herramientas CASE para ingeniería de Requisitos*. **Alarcón, Andrea y Sandoval, Erika**. 6, Boyacá : s.n., 2008.
29. *Extensión de Visual Paradigm for UML para el desarrollo dirigido*. **Cabrera González, Lianet y Roberto Pompa Torres, Enrique**. 10, La Lisa : s.n., 15 de 10 de 2012, Serie Científica. Universidad de las Ciencias Informáticas., Vol. 5, pág. 6. ISSN: | RNPS.
30. **Morales, Ricardo**. Colombia Digital. [En línea] 1 de Septiembre de 2014. [Citado el: 2 de febrero de 2016.] <https://colombiadigital.net/actualidad/articulos-informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html>.
31. php. [En línea] <http://php.net/manual/es/intro-what-is.php>.
32. **Potencier, Fabien y Zaninotto, Francois**. librosweb. [En línea] http://librosweb.es/libro/symfony_1_4/.
33. **Guardado, Iván**. web.ontuts. [En línea] 6 de julio de 2010. <http://web.ontuts.com/tutoriales/utilizando-doctrine-como-orm-en-php/>.

34. **Potencier, Fabien y Zaninotto, François.** *Symfony la guía definitiva*. 2006.
35. **De la Torre, Anibal.** adelat. [En línea] 2006.
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
36. **html5a.** [En línea] 2012.
<http://www.html5a.com.ar/temarios/descripcion.php?cod=68&punto=1>.
37. **James Gosling, Bill Joy, Guy Steele, Gilad Bracha.** *The Java language specification*. Tercera edición. 2005.
38. **Fernández, Lorenzo y Escribano, Pacheco.** *JavaScript. Ejercicios prácticos tutorizados*. s.l.: LULU Press, 2015, 2015. 1409204715, 9781409204718.
39. **Álvarez, Miguel Ángel.** *Manual de jQuery*. 2010.
40. **fergarcia.** [En línea] 25 de Enero de 2013.
<https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.
41. *Herramienta de Desarrollo Netbeans*. **Mendoza González, Geovanny**.
42. **Martínez, Rafael.** postgresql-es. [En línea] 2 de Octubre de 2010.
http://www.postgresql.org.es/sobre_postgresql.
43. pgadmin. [En línea] <http://www.pgadmin.org/features.php>.
44. **ACENS, [prod.]**. *Servidor web Nginx, una clara alternativa a APACHE*. [Documento]
45. jscharts. [En línea] <http://www.jscharts.com/>.
46. highcharts. [En línea] <http://www.highcharts.com/>.
47. **Larman, Craig y Hall, Prentice.** *MODELO DEL DOMINIO. UML y Patrones*. 2003.

48. **Fernández, Eduardo, Giorgi, Marcelo y Lorenzo, Tomás.** *VIEG, una herramienta para la Visualización de Información mediante Grafos.* Montevideo : s.n.
49. **Beas, J.M.** *Historias de Usuario.* [Blog] 2011.
50. **Pytel, P., y otros.** *INGENIERÍA DE REQUISITOS BASADA EN TÉCNICAS DE INGENIERÍA DEL CONOCIMIENTO.* [Documento] 2011. ISBN: 978-950-673-892-1 .
51. **Gómez, David Alejandro.** *ISO 9126.* [Blog] 2009.
52. Moodle. [En línea] 3 de Abril de 2007. https://docs.moodle.org/all/es/Caracter%C3%ADsticas_del_Glosario.
53. *Minería de Datos Educacional en Ambientes.* **R. Huapaya, Constanza, y otros.** Mar del Plata : s.n., 2012. 223-4816600 int 259.
54. *Selección de atributos predictivos del rendimiento académico de estudiantes en un modelo de B-learning.* **Cobo Ortega, Ángel, Rocha Blanco, Rocio y Alvarez Díaz, Yurlenis.** 37, 2011.
55. *Aplicando minería de datos para descubrir rutas de aprendizaje frecuentes en Moodle.* **Bogarín Vega, Alejandro, Romero Morales, Cristóbal y Cerezo Menéndez, Rebeca.** 5, s.l. : edmetec, 2016. E-ISSN: 2254-0059.
56. **Sayad, Dr. Saed.** Saedsayad. [En línea] http://www.saedsayad.com/missing_values.htm.
57. **Sosa García, Joan, Vega Pons, Sandro y Ruiz Shulcloper, José.** *Algoritmos de agrupamiento difuso, índices de validación: un estado del arte.* . Habana : CENATAV, 2012. ISSN 2072-6287.
58. *El patrón de diseño Modelo-Vista-Controlador y su implementación en Java Swing.* **Bascón Pantoja, Ernesto.** Junio de 2011.
59. **SOMMERVILLE, Ian.** *Ingeniería de Software.* [Documento] s.l. : Pearson Educación, 2011. ISBN 978-607-32-0603-7.

60. **Visconti, Marcello y Astudillo, Hernán.** *Fundamentos de Ingeniería de Software*. [Documento] Santa María : s.n.
61. **ZANINOTTO, Francois y POTENCIER, Fabien.** *Symfony 1.4, la guía definitiva*. ISBN 9782918390305.
62. **González Hernández, Daimara y Acosta Miranda, Yaniosky.** *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas: Sistema Informatizado para la Gestión de la Información de los Pacientes en Rehabilitación*. 2008.
63. **Booch, Grady, Rumbaugh, Jim y Jacobson, Ivar.** *UML El Lenguaje Unificado de Modelado*. 1999.
64. **Larman, Craig.** *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Prentice Hall.
65. msdn.microsoft.com. [En línea] 2015. <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
66. **JACOBSON, Ivar.** *A PPLYING UML IN T HE U NIFIED P ROCESS*. s.l. : Rational Software Corporation, 1999.
67. pruebasdesoftware. [En línea] 2005. <http://www.pruebasdesoftware.com/laspruebasdesoftware.htm>.
68. **Javier Zapata Sánchez.** pruebasdelsoftware.wordpress.com. [En línea] Enero de 2013. <https://pruebasdelsoftware.wordpress.com/>.
69. **Lemus, Guillermo.** es.slideshare.net. [En línea] 27 de abril de 2012. <http://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>.