Universidad de las Ciencias Informáticas

Facultad 4

Aplicación de técnicas de agrupamiento al comportamiento de los estudiantes en la Plataforma Educativa ZERA.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Noralys Almeida Milanés

Roberto Martínez Navarro

Tutores: Ing. Irina Ivis Santiesteban Pérez

Ing. Adrián García Sánchez

La Habana, junio de 2015

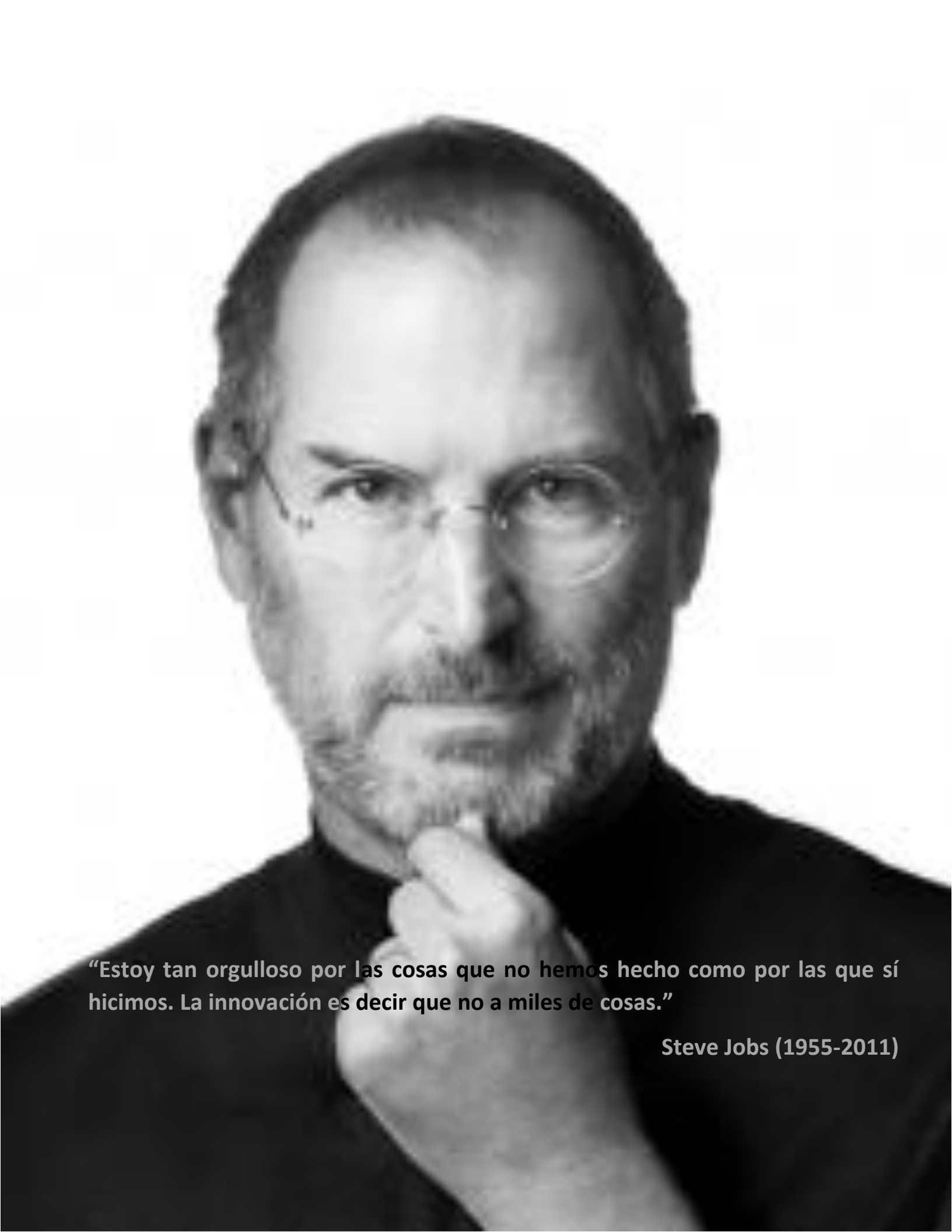
"Año 57 de la Revolución"

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores	Firma	Tutores	Firma
Noralys Almeida Milanés	_____	Ing. Irina Ivis Santiesteban Pérez	_____
Roberto Martínez Navarro	_____	Ing. Adrián García Sánchez	_____


“Estoy tan orgulloso por las cosas que no hemos hecho como por las que sí hicimos. La innovación es decir que no a miles de cosas.”
Steve Jobs (1955-2011)

Dedicatoria

De Noralys:

A ti abuela porque donde quiera que estés sé que siempre me acompañas, por tu amor, tu ternura y por tu forma de ser.

A mi mamá porque es una de mis razones de ser, es mi sostén, mi amiga y mi mejor ejemplo.

De Roberto:

Con amor y cariño dedico este trabajo a los seres que concibieron mi existencia, a los que han estado junto a mí en los momentos buenos y malos, los que han hecho realidad mis sueños y me han guiado siempre por el camino correcto, a ustedes, con ternura mi corazón por siempre.

Mamá, papá y abuelos.

Agradecimientos

De Noralys:

A Dios por permitirnos vivir este momento.

A mi mamá por su amor, sacrificio y entrega absoluta.

A mi tía Mercy por acogerme como una hija y por ser mi apoyo incondicional en estos 5 años.

A mi papá Salva por su cariño y por siempre estar presente.

A mi familia por enseñarme que no importan los contratiempos si sabemos tolerarnos y aceptarnos tal cual somos.

A la familia de Roberto por su cariño sincero.

A Maritza por su amistad y a Xiomara por tener esos detalles conmigo que hacen que recuerde a mi abuela.

A Roberto por su cariño, su paciencia y su amistad. Estoy muy feliz de haberte conocido.

A mi madre blanca Betty y a Maylin, gracias por darme la oportunidad de pertenecer a su familia.

A los amigos de la universidad y a las niñas del apartamento por los momentos compartidos.

A los tutores por siempre confiar en nosotros.

De Roberto:

A Dios por permitirme estar en este mundo tan hermoso.

A toda mi familia por el amor, cariño y apoyo que me han dado desde que nací, en especial a mis padres y mis abuelos que han sido mis guías durante toda mi vida.

A mi tía Maryanis y a mi tío Fernando, por acogerme como un hijo.

A mi compañera de tesis, mi novia Noralys Almeida Milanés, que siempre ha estado conmigo en los momentos buenos y malos; su amor, cariño y dedicación han sido cada día más fuerte y bello.

A la familia de mi novia por el gran afecto y apego que me tienen, en especial Elsa, Mercy y Salva.

A los tutores por la orientación y el apoyo brindado durante la realización de este trabajo.

A todos mis amigos, que de una forma u otra han ofrecido su ayuda en cualquier momento, principalmente los que vienen conmigo desde otro nivel de enseñanza y los que me han acompañado en estos 5 años de esfuerzos y sacrificios.

Resumen

El empleo de las Tecnologías de la Información y las Comunicaciones en el campo educativo se evidencia con la existencia de diversas herramientas que sirven de apoyo en el proceso de enseñanza y aprendizaje. La Plataforma Educativa ZERA es una de estas herramientas creada para ofrecer múltiples ventajas a profesores y alumnos. ZERA almacena información de las actividades que realizan los estudiantes y de los resultados académicos que obtienen. Esta información se muestra al profesor a través de un conjunto de reportes. A pesar de los reportes le resulta difícil al profesor realizar un análisis rápido del comportamiento de los estudiantes, por lo que la plataforma necesita de funcionalidades que resuelvan esta dificultad. Es por ello que se decide desarrollar un módulo haciendo uso de técnicas de agrupamiento de minería de datos, para caracterizar el comportamiento de los estudiantes en la plataforma. A partir de este objetivo se realizó un estudio de las herramientas, tecnologías y metodologías a emplear en la solución que contribuyó a su posterior elección. Se seleccionaron la metodología XP para el desarrollo de *software* y CRISP-DM para la minería de datos. Como lenguajes del lado del cliente se utilizaron 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, Apache como servidor web y PostgreSQL como gestor de bases de datos.

Palabras claves: agrupamiento, comportamiento, minería de datos.

Índice

Índice de tabla

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) influyen actualmente en todas las esferas sociales. En el campo educativo se pone de manifiesto con la introducción del *e-learning* o aprendizaje electrónico, definido como *"modalidad de enseñanza y aprendizaje que consiste en el diseño, puesta en práctica y evaluación de un curso o plan formativo desarrollado a través de redes de ordenadores y puede definirse como una educación o formación ofrecida a individuos que están geográficamente dispersos o separados o que interactúan en tiempos diferidos del docente empleando los recursos informáticos y de telecomunicaciones. Lo característico del e-learning es que el proceso formativo tiene lugar totalmente o en parte a través de una especie de aula o entorno virtual en el cual tiene lugar la interacción profesores-alumnos*[1] *así como las actividades de los estudiantes con los materiales de aprendizaje".*(1)

La adopción del *e-learning* supone una apuesta por un modelo pedagógico en el que el alumnado toma una mayor responsabilidad en su educación, contribuyendo al desarrollo de la eficiencia en el proceso de enseñanza y aprendizaje, y por ende, a la mejora cualitativa del modelo educativo.(2)

Para favorecer el desarrollo del *e-learning* han surgido diversas herramientas como los Sistemas de Gestión de Aprendizaje (LMS en inglés *Learning Management System*). Un LMS *"es un software instalado generalmente en un servidor web, que se emplea para crear, aprobar, administrar, almacenar, distribuir y gestionar las actividades de formación virtual (puede utilizarse como complemento de clases presenciales o para el aprendizaje a distancia)"* (3).

Los LMS tienen la característica de integrar herramientas comunes de la web con otras más específicas para brindar un ambiente controlado y de fácil utilización que permite esquematizar y coordinar procesos de enseñanza y aprendizaje según diversas corrientes pedagógicas.(4)

Las plataformas educativas[2] deben integrar módulos que ofrecen múltiples servicios a administradores, docentes y alumnos, los cuales pueden agruparse en las siguientes categorías (5):

- Herramientas de gestión de contenidos (CMS en inglés *Content Management System*)

[1] Interacción profesor-alumno, alumno-alumno.

[2] *Software* que permite la interacción asíncrona entre docentes y alumnos mediante la tecnología web, también se les identifica como EVA (Entornos Virtuales de Aprendizaje) o como LMS.

- Herramientas de comunicación y colaboración (foros, blogs, chats, correo electrónico).
- Herramientas de seguimiento y evaluación.
- Herramientas de administración y asignación de permisos.
- Herramientas complementarias, como portafolio, bloc de notas y sistemas de búsquedas de contenidos y materiales educativos.

Estos módulos permiten almacenar un conjunto de información relacionada a las actividades realizadas por los estudiantes. Actualmente esta información es aprovechada para extraer nuevo y útil conocimiento. Para el análisis de la información se utilizan diversas técnicas dentro de las que se encuentra la minería de datos, que aplicada a la educación ha propiciado el surgimiento de lo que es conocido como Minería de Datos Educacional (MDE).

La Universidad de las Ciencias Informáticas (UCI) tiene como una de sus líneas de desarrollo la referente a tecnologías para la formación, la cual abarca *software* educativo, tanto para los niveles curriculares de la enseñanza inicial, media o preuniversitaria, así como tecnologías de formación a distancia y semipresencial. Ejemplo de esto es el Centro de Tecnologías para la Formación (FORTES) *"dedicado al desarrollo de tecnologías que permiten ofrecer servicios y productos para las soluciones educativas, aplicando las TIC a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas, garantizando la calidad de las soluciones y la capacitación de los recursos humanos a partir de investigaciones que combinen los elementos pedagógicos y tecnológicos más avanzados, integrando así los procesos de enseñanza y aprendizaje, producción e investigación"* (6).

FORTES tiene como uno de sus productos la Plataforma Educativa ZERA la cual tiene como uno de sus objetivos ayudar al profesor con el análisis de los resultados académicos de los estudiantes y la interacción de estos con la plataforma. Actualmente la plataforma brinda a los profesores un conjunto de reportes entre los que se encuentran la calificación del estudiante en las diferentes evaluaciones que realiza, tiempo que demoró el estudiante en responder una tarea, tiempo que demoró en estudiarse el contenido, recursos a los que accedió, participación en el foro, etc.

Cuando el volumen de información que almacena la plataforma aumenta, resulta difícil para el profesor realizar un análisis de cada estudiante y conocer de forma rápida similitudes de uso de la plataforma. Además le es trabajoso establecer características similares y cuáles predominan e identificar

agrupaciones de alumnos que le permita trazar estrategias en el proceso de enseñanza y aprendizaje. Otro aspecto a tener en cuenta es que en ocasiones información valiosa pasa desapercibida pudiendo ayudar en la toma de decisiones. En estos momentos ZERA carece de funcionalidades que le permitan obtener esta información de forma rápida. Además, desaprovecha la oportunidad de darle valor agregado a los datos que almacena.

Atendiendo a lo planteado anteriormente se tiene como **problema a resolver:** ¿Cómo caracterizar el comportamiento de los estudiantes en la Plataforma Educativa ZERA?

Se define como **objeto de estudio:** aplicación de técnicas descriptivas de minería de datos a plataformas educativas y como **campo de acción:** aplicación de técnicas de agrupamiento en plataformas educativas.

Para resolver el problema planteado se tiene como **objetivo general:** Desarrollar un módulo que aplique técnicas de agrupamiento para caracterizar el comportamiento de los estudiantes en la Plataforma Educativa ZERA.

Teniendo en cuenta el objetivo general planteado se definen los siguientes **objetivos específicos:**

- Investigar los aspectos teóricos fundamentales que sustentan la investigación.
- Desarrollar el módulo para el análisis de los datos de ZERA.
- Probar las funcionalidades del módulo.

Para cumplir los objetivos específicos se tienen como **tareas:**

- Análisis de las técnicas, algoritmos, herramientas y metodologías empleadas en la minería de datos.
- Análisis de las soluciones existentes.
- Selección de las herramientas y tecnologías para la creación del módulo.
- Selección de una metodología de desarrollo de *software* que sirva de guía en el proceso de creación del módulo.
- Caracterización e integración de los datos que serán empleados en la investigación.
- Implementación del módulo que muestre el resultado de la aplicación de algoritmos de agrupamiento en la Plataforma Educativa ZERA.
- Comparación de los resultados obtenidos por los algoritmos aplicados.
- Realización de pruebas que evidencien el correcto funcionamiento del módulo.

Se tiene como **hipótesis:** El desarrollo de un módulo donde se apliquen técnicas de agrupamiento permitirá caracterizar el comportamiento de los estudiantes en la Plataforma Educativa ZERA.

En la investigación se utilizan los siguientes métodos teóricos:
Análisis-síntesis: para el estudio de las fuentes bibliográficas existentes referente al objeto de estudio, identificando los elementos más importantes y necesarios para dar solución al problema planteado.

Inductivo-deductivo: para el estudio de la utilización de técnicas de minería de datos en plataformas educativas y de las principales herramientas que implementan estas técnicas, con el objetivo de determinar qué alternativas pueden ser incorporadas en la presente investigación.

Hipotético-deductivo: para formular la hipótesis de la investigación de forma tal que tribute a la solución.

Histórico-lógico: con el fin de realizar un estudio de cómo se ha comportado el uso de la minería de datos en el campo educativo y las tendencias actuales de la aplicación de técnicas de minería de datos en plataformas educativas.

Análisis documental: en la consulta de la literatura especializada en las temáticas afines de la investigación.

Como método empírico:
Entrevista: Utilizado con el objetivo de conocer la aplicación de minería de datos en investigaciones realizadas en la universidad. Haciendo énfasis en el uso de herramientas para el análisis de datos y la integración de estas con otras aplicaciones, los algoritmos utilizados y la forma de representar los resultados.

La tesis está estructurada en tres capítulos:
Capítulo 1 Fundamentación teórica: En este capítulo se definen los elementos asociadas a la investigación y se analizan estudios anteriores relacionados con el tema para elaborar el marco teórico de la investigación. Además de describir las herramientas, metodologías y tecnologías a utilizar en la solución.

Capítulo 2 Propuesta de solución: Este capítulo describe la propuesta de solución relacionada a la aplicación de técnicas de agrupamiento a la Plataforma Educativa ZERA. Además muestra la utilización

de la metodología de desarrollo XP y la metodología de minería de datos CRISP-DM enfocadas a la investigación.

Capítulo 3 Implementación y prueba: El capítulo expone aspectos a tener en cuenta en la implementación del módulo y los métodos utilizados para probar su funcionamiento.

Para finalizar se presentan las conclusiones y recomendaciones derivadas de la investigación, la bibliografía consultada y los anexos con elementos notorios del proceso investigativo.

Capítulo 1 Fundamentación teórica

1.1 Introducción

El primer paso en un proceso investigativo es realizar un análisis de los principales aspectos relacionados al problema que se pretende dar solución. Para el presente trabajo se hizo necesario el estudio de conceptos, técnicas y herramientas asociados a la minería de datos. Cuando la investigación es en el mundo de la informática y se espera resolver el problema con un producto de *software* entonces hay que definir las tecnologías, herramientas, metodologías y lenguajes a utilizar.

Descubrimiento de conocimiento en bases de datos

El uso de las TIC resulta indispensable para recopilar, almacenar, transformar, analizar y visualizar la gran cantidad de información que es generada por las disímiles instituciones. El resumir datos para la toma de decisiones ha sido el campo tradicional de la estadística, pero hoy en día existen nuevas técnicas, que revelan patrones o asociaciones que usualmente nos eran desconocidas y se le ha llamado descubrimiento de conocimiento en bases de datos (KDD en inglés *Knowledge Discovery in Databases*). KDD está organizado en 5 fases (7):

- **Recopilación e integración:** en esta fase se seleccionan las distintas fuentes de información y se transforman los datos a un formato y unidad de medida comunes generando un almacén de datos.
- **Limpieza, selección y transformación:** en esta fase se eliminan o se corrigen los valores faltantes/erróneos y se seleccionan los atributos más relevantes o se generan nuevos atributos a partir de los existentes para reducir la complejidad de la fase de minería de datos. También se puede reducir la cantidad de instancias.
- **Minería de datos:** esta es la fase donde se eligen el trabajo a realizar (clasificación, agrupamiento, etc.) y el método a utilizar.
- **Evaluación e interpretación:** en este punto se analizan y evalúan los patrones obtenidos y en caso de ser necesario se retorna a alguna de las fases anteriores.
- **Difusión y uso (presentación):** en esta fase se hace uso de los resultados obtenidos y se difunden entre todos los potenciales usuarios.

1.2 Minería de datos

Una de las etapas del KDD es la Minería de datos, la cual ha sido definida de la siguiente forma:

Búsqueda de información nueva, valiosa y no trivial en grandes volúmenes de datos.(8)

Proceso analítico diseñado para explorar grandes cantidades de datos en búsqueda de patrones consistentes y/o relaciones sistemáticas entre variables y luego validar el hallazgo aplicando los patrones encontrados a nuevos conjuntos de datos.(9)

La minería de datos se ha empleado en numerosos campos cómo la economía, la medicina y la educación. En el campo educativo se ha incrementado el interés de utilizar minería de datos, centrándose en la aplicación de métodos de descubrimiento que utilicen los datos educacionales para comprender mejor a los estudiantes y el entorno en el que aprenden.

1.2.1 Minería de Datos Educacional

Al aplicar minería de datos en el área educativa surge lo que se conoce como Minería de Datos Educacional (MDE), término que ha sido definido como:

Toda minería de datos realizada sobre bases de datos relacionados con la educación, o bien, a aquella orientada hacia dicho sector.(10)

Es una disciplina emergente, interesada en el desarrollo de métodos para la exploración de datos provenientes del contexto educativo y el uso de esos métodos para comprender mejor a los estudiantes, y hacer ajustes en su aprendizaje.(11)

Una de las fuentes de análisis en la MDE son los datos que almacenan las plataformas educativas, que además de brindar múltiples servicios guardan información de las acciones de los estudiantes y de los resultados de las evaluaciones que realizan. A partir de estos datos se puede conocer el uso que hacen los estudiantes de la plataforma y establecer la relación entre resultados académicos y actividades realizadas.

Comportamiento

Un término fundamental utilizado en la investigación es comportamiento. Por tal motivo es necesario dejar claro su significado teniendo en cuenta el contexto en el que será empleado.

El comportamiento es la manera de comportarse (conducirse, portarse). Se trata de la forma de proceder de las personas u organismos frente a los estímulos y en relación con el entorno.(12)

El comportamiento remite a las acciones de una persona y a los hechos que muestra en la rutina cotidiana.(13)

A partir de los conceptos anteriores, en la investigación se define comportamiento como las acciones que realizan los estudiantes en la plataforma.

La MDE también ha sido usada para determinar patrones de comportamiento de estudiantes en plataformas educativas y para ello se han empleado diversas técnicas de minería de datos.

1.2.2 Modelos de minería de datos

Las técnicas de minería de datos 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 grandes categorías: supervisadas o predictivas y no supervisadas o descriptivas (14):

- **Descriptivas o no supervisadas:** este modelo aspira a descubrir patrones y tendencias sobre el conjunto de datos sin tener ningún tipo de conocimiento previo de la situación a la cual se quiere llegar. Descubre patrones en los datos analizados. Proporciona información sobre las relaciones entre los mismos.
- **Predictivas o supervisadas:** crean un modelo de una situación donde las respuestas son conocidas y luego, lo aplica en otra situación de la cual se desconoce la respuesta. Conociendo y analizando un conjunto de datos, intentan predecir el valor de un atributo, estableciendo relaciones entre ellos.

Cada modelo de minería de datos incluye un conjunto de tareas que pueden ser utilizadas en dependencia del objetivo del análisis a realizar.

1.2.3 Tareas de minería de datos

Los modelos de minería de datos se clasifican de acuerdo a las tareas que realizan. Dentro de las tareas del modelo descriptivo están: agrupamiento, reglas de asociación y descubrimiento de secuencia. Las tareas predictivas son: clasificación, regresión y series de tiempo.

Para cada una de las tareas de minería de datos existen un conjunto de algoritmos que son los que utilizan los datos para su ejecución. En la investigación se seleccionaron los de agrupamiento pues lo que se pretende es formar grupos de alumnos con características similares y describir su comportamiento.

1.2.4 Algoritmos de agrupamiento (*clustering*)

El problema del agrupamiento puede definirse como sigue: dados *n* puntos en un espacio *n-dimensional* particionar los mismos en *k* grupos tales que los puntos dentro de un grupo son más similares que cada uno a los de los otros grupos, dicha similaridad se mide atendiendo a alguna función distancia (función de disimilaridad) o alguna función de similaridad.

Los métodos de agrupamiento pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad.(8)

Los algoritmos jerárquicos son aquellos en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente , se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo aglomerativo o divisivo.(8)

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo.(8)

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.(8)

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-means

El algoritmo *K-means* es uno de los algoritmos de agrupamiento más conocidos y populares. *K-means* busca una partición óptima de los datos al minimizar la suma de criterio de error al cuadrado con un procedimiento de optimización iterativo, que pertenece a la categoría de algoritmos particionales. 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.(15)

Ventajas (16):

- Complejidad asintótica polinomial de grado 1 $(O(n*k*t))$

n: número de objetos a agrupar.

k: número de particiones a encontrar.

t: número de iteraciones.

- Adecuado en grupos compactos y bien separados.
- Con un gran número de variables, *K-means* puede ser computacionalmente más rápido que la agrupación jerárquica.

EM (Expectation Maximization)

Algoritmo de *clustering* probabilístico que clasifica dentro de los particionales. Se trata de obtener la FDP (Función de Densidad de Probabilidad) desconocida a la que pertenecen el conjunto completo de datos. Esta FDP se puede aproximar mediante una combinación lineal de NC componentes, definidas a falta de una serie de parámetros, que son los que hay que averiguar. El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el *likelihood* de los datos. Se trataría entonces de estimar los parámetros buscados, maximizando este *likelihood* (este criterio se conoce como ML-Maximun *Likelihood*). El algoritmo EM, procede en dos pasos que se repiten de forma iterativa:

- *Expectation*: Utiliza los valores de los parámetros, iniciales o proporcionados por el paso *Maximization* de la iteración anterior, obteniendo diferentes formas de la FDP buscada.
- *Maximization*: Obtiene nuevos valores de los parámetros a partir de los datos proporcionados por el paso anterior.

Después de una serie de iteraciones, el algoritmo EM tiende a un máximo local de la función L[3]. Finalmente se obtendrá un conjunto de clústeres que agrupan el conjunto de proyectos original. Cada uno

[3] Función l*og-likelihood* que calcula el logaritmo del *likelihood*.

de estos cluster estará definido por los parámetros de una distribución normal. Este algoritmo da resultados de gran utilidad para el conjunto de datos del mundo real. Se recomienda usarlo en una pequeña porción de datos. Su mayor desventaja es que es de naturaleza compleja.(17)

***Density Based Spatial Clustering of Applications with Noise* (DBSCAN[4])**

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 denso-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 borde. De esta forma DBSCAN construye grupos en los que sus puntos son o puntos centrales o puntos borde, un grupo puede tener más de un punto central.(18)

***Ordering Points to Identify the Clustering Structure* (OPTICS[5])**

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.(18)

1.2.5 Metodologías para aplicar minería de datos

Para concretar proyectos de minería de datos existen diversas metodologías y cada una de ellas tiene características y etapas o fases que indican cómo lograr el proceso de descubrimiento de conocimiento.

[4] Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido.
[5] Orden de Puntos para Identificar la Estructura del Agrupamiento.

Las metodologías permiten llevar a cabo el proceso de minería de datos en forma sistemática y no trivial. Ayudan a entender el proceso de descubrimiento de conocimiento y proveen una guía para la planificación y ejecución de los proyectos.(19)

Algunas de las metodologías existentes son: KDD *Process,* CRISP-DM[6], SEMMA[7] y Catalyst (conocida como P3TQ). CRISP-DM se ha convertido en la metodología más utilizada, según un estudio publicado en el año 2014 por la comunidad KDnuggets (*Data Mining Community's Top Resource*).

Con el objetivo de seleccionar una de las metodologías de minería de datos mencionadas se realiza un análisis comparativo considerando los siguientes aspectos:

Escenarios y puntos de partida considerados para el proyecto:
SEMMA y KDD *Process* inician el proyecto de minería a partir del conjunto de datos. CRISP-DM y P3TQ comienzan con un análisis del negocio y del problema organizacional.

Estructura de las fases del proceso:

Tabla 1: Estructura de las fases del proceso.

Fases	Proceso KDD	CRISP-DM	SEMMA	P3TQ
Análisis y comprensión del negocio		Contempla análisis y comprensión del problema.		Contempla análisis y comprensión del problema.
Selección y preparación de los datos	Se contempla la selección y preparación de los datos.	Se contempla la selección y preparación de los datos.	Se contempla la selección y preparación de los datos.	Se contempla la selección y preparación de los datos.
Modelado	Se aplican las técnicas de minería.	Se aplican las técnicas de minería.	Se aplican las técnicas de minería.	Se aplican las técnicas de minería.

[6] Cross-Industry Standard Process for Data Mining.

[7] *Sample (Muestreo)*, *Explore (Exploración)*, *Modify (Modificación)*, *Model (Modelo)* and *Assess (Valoración).*

Evaluación	Se realiza en función de la utilidad que se aporta al dominio de aplicación o problema organizacional.	Se realiza en función de la utilidad que se aporta al dominio de aplicación o problema organizacional.	Se realiza sobre el desempeño del modelo.	Se realiza en función de la utilidad que se aporta al dominio de aplicación o problema organizacional.
Implementación	Utilización del nuevo conocimiento.	Despliegue.		Comunicación.

Nivel de detalle en las tareas de cada fase:

KDD y SEMMA proponen sólo los pasos generales del proyecto de minería de datos mientras que CRISP-DM y P3TQ, especifican con mayor detalle las actividades del proceso.

Para la investigación se escoge la metodología CRISP-DM teniendo en cuenta los siguientes motivos (19):

- Concibe el proyecto de minería de datos estrechamente relacionado al negocio en cuestión.
- Puede ser empleada independientemente de la herramienta que se utilice para el desarrollo del proyecto.
- Es de libre distribución y se encuentra en constante perfeccionamiento por parte de la comunidad internacional.
- Establece un conjunto de tareas y actividades para cada fase del proyecto.
- Es la que cuenta con mayor aceptación por parte de los desarrolladores de procesos de extracción de conocimientos a partir de datos.
- Suficientemente amplia y flexible para aplicarla a proyectos de cualquier tamaño.

CRISP-DM

CRISP-DM es una metodología estándar que ha sido desarrollada para la construcción de proyectos de minería de datos. Creada por el grupo de empresas SPSS, NCR y Daimer Chrysler en el año 2000, es actualmente la guía de referencia más utilizada en el desarrollo de proyectos de minería de datos.

Estructura el proceso en seis fases: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación e Implantación (19).

1.2.6 Herramientas de minería de datos

Actualmente existen múltiples herramientas para el análisis de datos. Uno de los objetivos de esta investigación es utilizar una de estas herramientas para aplicar técnicas de pre-procesamiento y minería a los datos que almacena la Plataforma Educativa ZERA. La selección de la herramienta fue hecha teniendo en cuenta los algoritmos de agrupamiento implementados y la posibilidad de integración con otras aplicaciones. Las herramientas analizadas fueron RapidMiner, R y WEKA.

RapidMiner

RapidMiner es un *software* de código abierto con licencia GNU GPL para el descubrimiento de conocimiento y minería de datos. Es un entorno con muchos algoritmos de aprendizaje y otras utilidades añadidas, está desarrollada sobre el lenguaje Java y funciona en los sistemas operativos más conocidos.(20)

R

R es un conjunto integrado de servicios de *software* para la manipulación de datos, cálculo y representación gráfica. Se compila y ejecuta en una amplia variedad de plataformas UNIX, Windows y MacOS. R ofrece una amplia variedad de técnicas estadísticas (análisis de series de tiempo, modelado lineal y no lineal, pruebas estadísticas clásicas, clasificación y agrupación) y gráficas, y puede ser extendido a través de paquetes. R está disponible como *software* libre bajo los términos de la Licencia Pública General GNU de la *Free Software Foundation* en forma de código fuente.(21)

WEKA

Las iniciales WEKA responden a *Waikato Environment for Knowledge Analysis*, se trata de una herramienta de libre distribución desarrollada en la Universidad de Waikato (Nueva Zelanda), escrita en lenguaje Java y que permite realizar multitud de análisis. La herramienta está aplicada a procesos de minería de datos, por lo que agrupa diferentes técnicas: pre-procesado, agrupamiento o *clustering*, clasificación, generación de reglas de asociación, etc. También incluye facilidades para la visualización de los datos.(22)

La herramienta dispone de cuatro interfaces distintas (22):

1. **Interfaz Simple CLI:** permite la introducción de todo tipo de comandos, pero no es posible realizar representaciones gráficas, el interfaz en modo texto permite instanciar las distintas clases Java definidas en el programa WEKA.
2. **Interfaz *Explorer*:** es el interfaz gráfico básico, en él se pueden mostrar gráficamente tanto las características de los datos de partida como los resultados de los análisis. Permite introducir los comandos con ayuda del *mouse*, seleccionando los operadores adecuados en menús desplegables.
3. **Interfaz *Experimenter*:** se trata de un interfaz gráfico más avanzado, en el que no solo se pueden realizar análisis sobre los datos, sino que además es posible comparar el funcionamiento de diferentes algoritmos (por ejemplo, diferentes clasificadores) o bien comparar distintos ficheros de datos.
4. **Interfaz *KnowledgeFlow*:** este último interfaz permite representar como una red de operadores en cascada los procesos a realizar sobre los datos.

Para seleccionar la herramienta a utilizar se tuvieron en cuenta los siguientes aspectos:

Algoritmos de agrupamiento implementados

Hay algoritmos que están implementados en la propia herramienta y otros pueden ser añadidos según la opción brindada por cada una. R puede ser extendido a través de paquetes. Hay alrededor de ocho paquetes suministrados con la distribución R y muchos más están disponibles a través de la familia CRAN[8] de sitios de Internet que cubren una amplia gama de estadísticas modernas. WEKA y RapidMiner permiten añadir algoritmos haciendo uso de sus APIs.(20) (23)

Integración con otras aplicaciones

RapidMiner ejecuta un flujo de trabajo que integra la carga de datos, transformación de datos y ejecución del algoritmo. Además de RapidMiner existe RapidAnalytics el cual utiliza RapidMiner como motor y ofrece, entre otras cosas, la ejecución remota y programada de los procesos de análisis. El resultado de estos procesos puede ser consumido a través de servicios web por otras aplicaciones.(20)

[8] Comprehensive R Archive Network.

Además de R existe RStudio y RStudio Server, el primero es un entorno de desarrollo integrado (IDE) para R y el segundo un servidor que permite ejecutar los procesos de RStudio para que sean consumidos a través de servicios web.(24)

Con WEKA es posible introducir su código Java en una aplicación Java propia, de modo que se puede crear un programa específico, que haga uso de las utilidades de WEKA que se desee. En segundo lugar, también es posible hacer llamados a las clases Java que tiene definida desde el shell del sistema operativo del que se disponga (sea este Windows, Linux o Macintosh).(25)

Se decide escoger WEKA porque ofrece la vía más directa de integrar y puede ser usada con fines comerciales. Para usar RapidMiner o R sería necesario instalar una herramienta adicional.

Herramienta para validar resultados del agrupamiento

MOA (Massive Análisis on-line) es un *framework* de código abierto para la minería flujo de datos escrito en Java. Incluye herramientas para la evaluación y una colección de algoritmos de aprendizaje automático (clasificación, regresión, agrupamiento, detección de valores atípicos y sistemas de recomendación). MOA permite realizar los experimentos a través de una interfaz gráfica de usuario (GUI) o la ejecución de la línea de comandos.

Aplicación de técnicas de minería de datos a plataformas educativas

El uso de LMS como apoyo a la labor educativa es cada vez más frecuente. Ejemplo de estos son Moodle, Dokeos, Claroline, Webct, ILIAS y Blackboard. Estos sistemas almacenan un conjunto de información de la interacción de los usuarios que hoy en día es aprovechada para aplicar técnicas de minería de datos. Las técnicas más utilizadas en la minería de datos aplicada a los sistemas de *e-learning* son: clasificación y agrupamiento, descubrimiento de reglas de asociación y, análisis de secuencias. (26) A continuación se mencionan un conjunto de investigaciones relacionadas a la aplicación de técnicas de minería de datos en plataformas educativas.

La Universidad Técnica Particular de Loja (UTPL) posee una plataforma virtual que es utilizada como herramienta de soporte a la educación en sus dos modalidades: abierta y presencial. Aprovechando la información que se almacena de la interacción de estudiantes y profesores se realizó un análisis de minería de datos sobre la plataforma virtual de la UTPL. El propósito era conocer cómo los estudiantes

interactuaban con las herramientas de la plataforma, para así presentar recomendaciones para la mejora de estos servicios y aprovechar realmente las ventajas que ofrecen este tipo de sistemas.(27)

En la investigación "*Clustering Learners according to their Collaboration*" presentan la aplicación del algoritmo EM para construir clústeres, el cual usa indicadores estadísticos de la interacción de los estudiantes en los foros para la obtención de grupos de estudiantes de acuerdo a su nivel de colaboración.(28)

La utilización de técnicas de agrupamiento es utilizada por Gord McCalla y Tiffany Tang para formar grupos de usuarios basándose en su comportamiento de navegación.(26)

Los estudios similares sirven de referencia para conocer las técnicas y herramientas de minería de datos más utilizadas en el contexto educativo y los objetivos hacia los que se enfoca el uso de minería de datos en plataformas educativas. En las investigaciones estudiadas hacen uso de herramientas de minería de datos pero no exponen cómo integrar estas herramientas a plataformas educativas.

1.3 Descripción de las metodologías, herramientas, lenguajes y tecnologías

En el proceso de desarrollo de un *software* es imperioso el uso de las metodologías, herramientas y tecnologías que apoyen y favorezcan el avance de forma rápida y eficiente, estas facilitan el trabajo y agregan funcionalidades que permiten al equipo de desarrollo la obtención de un producto final que satisfaga las expectativas del cliente e incluya las nuevas tendencias en el campo de la informática. Para ello se ha realizado un estudio de las metodologías, herramientas, lenguajes y tecnologías que serán usadas en la construcción de la solución propuesta en la investigación.

1.3.1 Metodología de desarrollo de *software*

Las metodologías de desarrollo de *software* tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelamiento de sistemas que permitan desarrollar *software* de alta calidad. Una metodología de desarrollo de *software* es usada para estructurar, planear y controlar el proceso de desarrollo (29). Las metodologías de desarrollo de *software* se clasifican en tradicionales o ágiles. Las tradicionales se focalizan en documentación, planificación y procesos, o sea, llevar una documentación exhaustiva de todo el proyecto y cumplir con el plan de proyecto. Sin embargo, las ágiles se basan en

retrasar las decisiones y la planificación adaptativa, permitiendo potenciar aún más el desarrollo de *software* a gran escala.(30)

Programación Extrema

Programación Extrema (XP en inglés Extreme Programming) es un enfoque ágil para el desarrollo de *software*. Fue concebido y desarrollado para hacer frente a las necesidades específicas de desarrollo de *software* llevada a cabo por los equipos pequeños. Enfatiza la implicación del cliente y las pruebas, además de que se adapta a los requerimientos que cambian rápidamente. XP incluye (31):

- Una filosofía de desarrollo de *software* basada en valores de comunicación, retroalimentación, simplicidad, coraje y respeto.
- Un conjunto de prácticas de probada utilidad para mejorar el desarrollo de *software*. Prácticas que se complementan entre sí, amplificando sus efectos.
- Un conjunto de principios complementarios.

Para definir las responsabilidades en la investigación se hizo uso de los roles propuestos por XP. Los roles utilizados son (31):

- Programador
- Cliente
- Encargado de pruebas (Tester)

De las prácticas que propone XP en la investigación se utilizaron las siguientes (31):

- El juego de la planificación
- Entregas pequeñas
- Diseño simple
- Pruebas
- Refactorización
- Programación en parejas
- Integración continua
- Cliente in-situ
- Estándares de programación

El ciclo de vida de XP consiste de seis fases: Exploración, Planificación, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.(31) En la investigación solo serán documentadas las fases de Exploración, Planificación e Iteraciones.

Teniendo en cuenta el tiempo de duración para la entrega del producto, la cantidad de funcionalidades a implementar, la posibilidad de constantes cambios y la facilidad de interacción con el cliente se escoge como metodología de desarrollo XP.

1.3.2 Lenguajes y tecnologías del lado del cliente

Un programa cliente es el *software* necesario en el equipo cliente para tener acceso al servicio que ofrece el programa servidor.(32)

HTML

HTML es el lenguaje que se emplea para el desarrollo de páginas web, el cual nos permite representar el contenido y también referenciar otros recursos (imágenes, etc.), enlaces a otros documentos, mostrar formularios para posteriormente procesarlos, etc. HTML no es propiamente un lenguaje de programación sino un sistema de etiquetas. Se diseñó con el objetivo de estructurar documentos y mostrarlos en forma de hipertexto y permite establecer relaciones unidireccionales entre ellos.(33) La versión que será empleada es la 5.0.

CSS

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.(34) La versión que será empleada es la 3.0, debido a que incorpora nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas.

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Este es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.(35)

Framework jQuery

jQuery es un framework para el lenguaje JavaScript que implementa una serie de clases (de programación orientada a objetos) que permite programar sin preocuparse del navegador que está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales. Este framework nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Ejemplo de ello, obtener ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc.(36) La versión que será empleada es la 1.10.2.

1.3.3 Lenguajes y tecnologías del lado del servidor

Un programa servidor es el que debe estar ejecutándose en el equipo servidor para que este pueda ofrecer un servicio.(32)

PHP

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Cuenta con una gran librería de funciones, una amplia documentación y sobre todo es un lenguaje del lado del servidor; es decir, se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente.(37)

Entre sus principales características cabe destacar su facilidad de aprendizaje, es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. Capacidad de expandir su potencial al utilizar una enorme cantidad de módulos (llamados ext's o extensiones), posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Permite las técnicas de Programación Orientada a Objetos.(37)

Framework Symfony

Es un *framework* diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiéndole al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.(38)

Symfony es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix y Linux como en plataformas Windows.(38)

Características (38):

- Independiente del SGBD.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Código fácil de leer y permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Se publica bajo licencia MIT[9], con la que se pueden desarrollar aplicaciones web comerciales, gratuitas y/o de *software* libre.

La versión que será empleada es la 1.4, pues la Plataforma Educativa ZERA se encuentra desarrollada en esa versión.

Doctrine

ORM[10] (*Object Relational Mapping*) que se encarga de gestionar el modelo de datos en Symfony. Permite que el acceso y la modificación de los datos almacenados en la base de datos se realicen mediante objetos, así nunca se accede de forma explícita a la base de datos permitiendo un alto nivel de abstracción.(39)

Lime

[9] Massachusetts Institute of Technology.
[10] Interfaz que traduce la lógica de los objetos a la lógica relacional.

En el ámbito de PHP existen muchos *frameworks* para crear pruebas unitarias, siendo los más conocidos PhpUnit y SimpleTest. Symfony incluye su propio *framework* llamado Lime. Las pruebas unitarias de Symfony son archivos PHP normales cuyo nombre termina en Test.php y que se encuentran en el directorio test/unit/ de la aplicación. Su sintaxis es sencilla y fácil de leer.(40)

Lime proporciona el soporte para las pruebas unitarias y tiene las siguientes ventajas (40):

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los *frameworks* de pruebas garantizan un entorno de ejecución "limpio" para cada prueba.
- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- 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.

Para ejecutar las pruebas unitarias Lime propone los siguientes métodos (40):

Tabla 2: Métodos Lime para pruebas unitarias.

Método	Descripción
diag($mensaje)	Muestra un comentario, pero no ejecuta ninguna prueba.
ok($condicion[, $mensaje])	Si la condición que se indica es true, la prueba tiene éxito.
is($valor1, $valor2[, $mensaje])	Compara dos valores y la prueba pasa si los dos son iguales (==).
isnt($valor1, $valor2[, $mensaje])	Compara dos valores y la prueba pasa si no son iguales.
like($cadena, $expresionRegular[, $mensaje])	Prueba que una cadena cumpla con el patrón de una expresión regular.
unlike($cadena, $expresionRegular[, $mensaje])	Prueba que una cadena no cumpla con el patrón de una expresión regular.

cmp_ok($valor1, $operador, $valor2[, $mensaje])	Compara dos valores mediante el operador que se indica.
isa_ok($variable, $tipo[, $mensaje])	Comprueba si la variable que se le pasa es del tipo que se indica.
isa_ok($objeto, $clase[, $mensaje])	Comprueba si el objeto que se le pasa es de la clase que se indica.
can_ok($objeto, $metodo[, $mensaje])	Comprueba si el objeto que se le pasa dispone del método que se indica.
is_deeply($array1, $array2[, $mensaje])	Comprueba que dos arreglos tengan los mismos valores.
include_ok($archivo[, $mensaje])	Valida que un archivo existe y que ha sido incluido correctamente.
fail([$mensaje])	Provoca que la prueba siempre falle (es útil para las excepciones).
pass([$mensaje])	Provoca que la prueba siempre se pase (es útil para las excepciones).
skip([$mensaje, $numeroPruebas])	Cuenta como si fueran *$numeroPruebas* pruebas (es útil para las pruebas condicionales).
todo([$mensaje])	Cuenta como si fuera una prueba (es útil para las pruebas que todavía no se han escrito).
comment($mensaje)	Muestra el comentario indicado pero no cuenta como ninguna prueba.
error($mensaje)	Muestra el mensaje de error indicado pero no cuenta como ninguna prueba.
info($mensaje)	Muestra el mensaje informativo indicado pero no cuenta como ninguna prueba.

Entorno de Desarrollo Integrado NetBeans

El Entorno de Desarrollo Integrado NetBeans es una herramienta pensada para escribir, compilar, depurar y ejecutar programas. Consta de una gran comunidad de usuarios en constante crecimiento, lo que le ha

permitido, al igual que muchos otros sistemas libres, el progreso paulatino de sus prestaciones y la eliminación de errores que pudiesen existir.(41)

Ofrece todas las herramientas necesarias para crear aplicaciones web con lenguajes dinámicos como PHP y JavaScript. Es fácil de instalar y se puede ejecutar tanto en Windows como en Linux. NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de *software*. Brinda una barra de navegación para el acceso rápido a funciones en una clase muy extensa, detecta errores de sintaxis en tiempo real, además de un completamiento de código fuente eficiente y seguro.(41) La versión que será empleada es la 8.0.

Sistema Gestor de Bases de Datos

Un Sistema Gestor de Base de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Está compuesto por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, además de garantizar la seguridad e integridad de los mismos.(42)

PostgreSQL

PostgreSQL es un SGBD relacional, orientado a objetos y de código abierto, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Dentro de sus principales ventajas se encuentran (42):

- Soporta distintos tipos de datos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendaria.
- Es extensible a través del código fuente, disponible sin costos adicionales.
- Soporte nativo para los lenguajes más populares del medio: PHP, C++, Perl, Python, entre otros.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos especiales, minería de datos, entre otros.
- Es multiplataforma, disponible en Linux, Unix, Mac Os X y Windows, entre otros.

Se seleccionó PostgreSQL 9.2.4.1, debido a que es multiplataforma, confiable, estable y con funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad. Además es el que emplea ZERA.

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 SQL 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)

Servidor Web Apache

Constituye uno de los servidores web más populares del mercado, y uno de los más utilizados actualmente, de código abierto y gratuito, disponible para Windows y GNU/Linux. Alcanzó su máxima cuota de mercado en 2005 al ser el servidor empleado en el 70% de los sitios web en el mundo. Es considerado el servidor web por excelencia, pues desde su surgimiento ha demostrado que es estable y que tiene mejor rendimiento que sus competidores.(44)

Ventajas (44):

- Fiabilidad: Más del 90% de los servidores con más alta disponibilidad funcionan bajo un servidor Apache.
- *Software* libre: El servidor Apache es gratuito y es distribuido bajo la licencia de Apache en la cual se permite realizarle cambios al código fuente.
- Extensibilidad: Se pueden añadir módulos para ampliar aún más las capacidades del servidor.

Se decide utilizar como servidor web, Apache en su versión 2.4.7, porque fue creado para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Es un *software* libre y multiplataforma que permite a clientes o instituciones construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización.

1.4 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, JS Chart y Flot.

JS Charts

JS Charts es un generador gráfico basado JavaScript que requiere poca o ninguna codificación. Con JS Charts dibujar gráficos es una tarea simple y fácil, ya que sólo tiene que utilizar secuencias de comandos del lado del cliente (es decir, realizada por el navegador web). No se requieren plugins adicionales o módulos del servidor. JS Charts permite crear gráficos en diferentes plantillas como gráficos de barras, gráficos circulares o gráficos de líneas simples.(45) Sus principales características pueden resumirse de la siguiente forma (45):

- Fácil de integrar.
- Personalizable.
- Pueden crearse los tres tipos más comunes de gráficos: barras, circulares y de línea.
- Es compatible con la mayoría de los navegadores web.
- Es gratis.

Morris js

Es un plugin de gráficos basado en jQuery, creado por Olly Smith. Es libre bajo la licencia BSD simplificada. Puede generar gráficos de línea, área, barras, ranura y tablas Donut. Las plataformas en las que es compatible son: Firefox, Internet Explorer, Google Chrome, Safari, Opera, iPhone, iPad.(46)

Flot

Es una librería gráfica JavaScript para jQuery, con un enfoque en el uso simple, atractiva apariencia y características interactivas. Se caracteriza por (47):

- Funciona con Internet Explorer 6+, Chrome, Firefox 2 +, Safari 3+ y Opera 9.5+.
- Flot soporta líneas, puntos, zonas llenas, barras y cualquier combinación de éstos, en el mismo gráfico y hasta en la misma serie de datos.
- Puede configurar los puntos en los ejes, la leyenda, el tipo de gráfico, etc.

Después de ver las posibilidades que ofrecen cada una de las librerías se seleccionan Flot y Morris js porque poseen los tipos de gráficos que se necesitan representar.

1.5 Lenguajes y tecnologías de modelado

En el proceso de desarrollo de *software* se generan artefactos que facilitan la construcción de la solución. Actualmente existen lenguajes y herramientas utilizadas para modelar estos artefactos.

Lenguaje Unificado de Modelado

UML es un lenguaje de modelado visual, está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, y puede ser usado en todas las fases de desarrollo del *software*. Su función principal es especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Se utiliza para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. No garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios, razón por la cual se escoge para generar el Modelo del Dominio.(48)

Visual Paradigm for UML

Visual Paradigm para UML es una herramienta para especificar, visualizar y documentar los distintos aspectos del sistema de *software*. El *software* de modelado UML ayuda a una más rápida construcción de aplicaciones con calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.(49)

De manera general, esta herramienta de modelado ofrece (49):

- Entorno de creación de diagramas para UML.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Disponibilidad en múltiples plataformas.

La versión que será empleada es la 8.0.

1.6 Conclusiones parciales

- El uso de la minería de datos en el campo educativo se incrementa como área de investigación.
- La selección de la metodología de minería de datos CRISP-DM permite entender el proceso de descubrimiento de conocimiento y es una guía para la planificación y ejecución del proyecto.
- La técnica de agrupamiento es una de las más empleadas en la MDE.

- Las metodologías, herramientas y tecnologías seleccionadas favorecen la construcción de la solución.

Capítulo 2 Propuesta de solución

2.1 Introducción

El uso de una metodología en el desarrollo de *software* es recomendable pues constituye una guía en todo el proceso. Una de estas metodologías es XP la cual propone un conjunto de fases que generan artefactos que tributan en la construcción de la solución. Por otro lado, cuando el objetivo del producto es aplicar minería de datos, resulta apropiado el uso de una metodología que muestre como extraer conocimiento en un conjunto de datos. En este caso se describen las fases y tareas propuestas por CRISP-DM.

2.2 Modelo conceptual

Un modelo conceptual explica los conceptos significativos en un dominio del problema. Es un diccionario visual de términos importantes del dominio, utiliza la notación UML de diagrama de estructura estática. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema de *software*.(50)

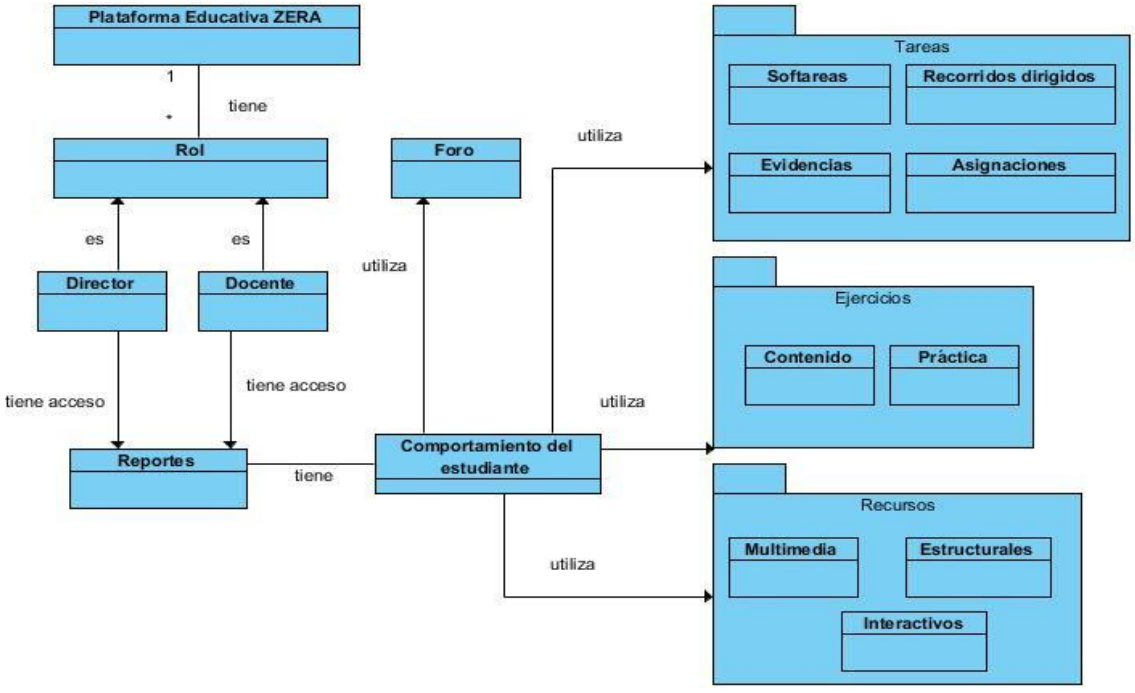


Figura 1: Modelo conceptual.

Términos del modelo conceptual

- **Plataforma Educativa ZERA:** destinada a apoyar el proceso de enseñanza y aprendizaje.
- **Rol:** constituye el nivel de acceso de los usuarios en la plataforma.
- **Docente (Profesor):** es la persona que guía el proceso de enseñanza del estudiante, le asigna actividades al grupo o individualmente, lleva el control de las tareas de los estudiantes, hace un seguimiento a las actividades de todos los estudiantes que estén a su cargo.
- **Director:** accede a todos los grupos de su institución para supervisar el trabajo de los docentes y estudiantes.
- **Reportes:** es el módulo dentro de la plataforma destinado a los reportes.
- **Comportamiento del estudiante:** son los reportes relacionados a la actividad en el foro, el acceso a recursos, realización de ejercicios, realización de tareas y comportamiento general en la plataforma.
- **Softareas:** modalidad de tarea que realiza el estudiante.
- **Recorridos dirigidos:** modalidad de tarea que realiza el estudiante.
- **Evidencias:** modalidad de tarea que realiza el estudiante.
- **Asignaciones:** modalidad de tarea que realiza el estudiante.
- **Contenido:** ejercicios que realiza el estudiante en la sección Contenido.
- **Prácticas:** ejercicios que realiza el estudiante en la sección Prácticas.
- **Multimedia:** recursos multimedia a los que accede el estudiante.
- **Interactivos:** recursos interactivos a los que accede el estudiante.
- **Estructurales:** recursos estructurales a los que accede el estudiante.

2.3 Descripción del proceso de agrupamiento

El proceso de agrupamiento de minería de datos se lleva a cabo teniendo en cuenta las siguientes fases de la metodología CRISP-DM:

- Comprensión del negocio
- Comprensión de los datos
- Preparación de los datos
- Modelado
- Evaluación

2.3.1 Comprensión del negocio

En la comprensión del negocio se definen los objetivos del negocio y de minería de datos, a partir de eso se crea un plan con las tareas de minería de datos a realizar durante todo el proceso de extracción de conocimiento.

Objetivos del negocio

ZERA es una plataforma educativa creada en la UCI y utilizada como apoyo en el proceso de enseñanza y aprendizaje. Almacena un conjunto de información que puede ser aprovechada desde varios puntos de vista. En el caso de la presente investigación se necesita para describir el comportamiento de los estudiantes que interactúan con la plataforma aportándole elementos al profesor en el proceso de toma de decisiones. A raíz de esta situación se tiene como objetivos de negocio:

- Caracterizar el comportamiento de los estudiantes en la Plataforma Educativa ZERA.
- Obtener resultados que ayuden a los profesores a trazar estrategias en el proceso de enseñanza y aprendizaje.

Evaluación de la situación

Recursos disponibles para la investigación:

- Personal: 2 estudiantes de 5to año de la carrera Ingeniería en Ciencias Informáticas.
- Hardware: 1 laptop, 1 PC.
- *Software*: herramienta de minería de datos WEKA.
- Fuente de datos: base de datos Plataforma Educativa ZERA.
- Fecha terminación: mayo 2015.
- 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, 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 3 Plan del proyecto.

Actividades	Fecha inicio	Fecha fin
Estudio de la base de datos de ZERA.	2/03/2015	13/03/2015
Definir datos a extraer.	16/03/2015	20/03/2015
Extraer datos.	23/03/2015	3/04/2015
Seleccionar técnicas y herramientas de minería de datos.	6/04/2015	10/04/2015
Pre-procesar datos.	13/04/2015	17/04/2015
Aplicar algoritmos.	20/04/2015	24/04/2015
Comprobar resultados de la aplicación de los algoritmos.	27/04/2015	30/04/2015

2.3.2 Comprensión de los datos

En esta etapa el principal objetivo es extraer los datos que se necesitan para realizar el análisis. Además se verifica qué condiciones presentan los datos a los que se le aplicará el algoritmo de minería.

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 245 tablas. De este modelo se seleccionaron las tablas que almacenaban información de interés para la investigación, trabajando finalmente con 47. Los datos a extraer son:

Tabla 4: Datos recolectados.

Datos	Tabla	Tipo
Ejercicios		
Número de ejercicios realizados (Prácticas).	tb_practice	Numérico
Número de ejercicios realizados (Contenido).	tb_practice	Numérico
Tiempo que dedica a realizar ejercicios de	tb_attemp	Numérico

práctica.		
Tiempo que dedica a realizar ejercicios de contenido.	tb_attemp	Numérico
Cantidad de ejercicios realizados de complejidad fácil.	tb_exercise	Numérico
Cantidad de ejercicios realizados de complejidad media.	tb_exercise	Numérico
Cantidad de ejercicios realizados de complejidad alta.	tb_exercise	Numérico
Promedio de calificación de ejercicios (Prácticas) y (Contenido).	tb_practice	Numérico
Recursos		
Número de recursos visitados por tipo.	tb_trace	Numérico
Número de veces que consulta el glosario de términos.	tb_trace	Numérico
Tareas		
Número de tareas que realiza el estudiante.	tb_modality_student	Numérico
Número de tareas que asigna el profesor.	tb_modality_student	Numérico
Cantidad de tareas que realiza el estudiante de las que asigna el profesor.	tb_modality_student	Numérico
Promedio de calificación por tarea.	tb_evaluation_student	Numérico
Foro		
Número de veces que el estudiante visita el foro.	r_topic_user	Numérico
Número de temas de discusión que crea el estudiante en el foro.	tb_topic	Numérico
General		
Género.	sf_guard_user	Nominal
Edad.	sf_guard_user	Nominal
Porciento de asistencia del estudiante.	tb_assistance	Numérico

Número de accesos a la plataforma.	tb_session	Numérico
Último acceso o visita.	sf_guard_user	Numérico
Cantidad de días sin visitar el sitio.	sf_guard_user	Numérico
Tiempo total de duración en la plataforma.	tb_session	Numérico
Número de páginas que el usuario ha visitado.	tb_trace	Numérico

Verificación de calidad de datos

En esta tarea se analiza la calidad de los datos. Para la presente investigación los criterios a tener en cuenta fueron:

- Representación de la realidad.
- Error en los datos.
- Existencia de campos vacíos.

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.

2.3.3 Preparación de los datos

La preparación de los datos define cuales van a ser utilizados realmente, se realizan las transformaciones necesarias y se les da el formato adecuado.

Seleccionar los datos

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.

Los foros, por ejemplo, son un servicio de comunicación ampliamente usado en ambientes *e-learning*, por lo que la información obtenida de la interacción en los mismos permite determinar características de colaboración del estudiante. Los datos usados para realizar el análisis son: número de foros que el usuario ha visitado, número de mensajes que el usuario ha enviado a los foros, número de conversaciones que el usuario inicia.(28)

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.(51)

La tabla que se muestra a continuación contiene atributos propuestos en investigaciones anteriores sobre minería de datos en plataformas *e-learning*.(52) (53) (54) (55) (56) (57) (58) (59)

Tabla 5: Información útil de la interacción del alumno con los LMS.

Información
Tiempo de estudio (acceso a páginas, por sesión o tema).
Tiempo empleado en responder una cuestión o tarea.
Páginas visitadas (revisiones).
Número de páginas leídas.
Preguntas contestadas.
Participación en foros.
Recursos más interesantes (demandados o descargados).

De los atributos listados en la tabla 2 se excluyen:

- Género, Edad: no se seleccionan estos datos para el modelo porque el objetivo es conocer el comportamiento general en la plataforma, sin tener en cuenta género o edad. Además muchos estudiantes tenían vacío estos campos pudiendo afectar el agrupamiento.
- Último acceso o visita: este dato estuvo mal elaborado en su concepción inicial ya que persigue el mismo objetivo que Cantidad de días sin visitar el sitio.

Limpiar, estructurar e integrar datos

Con el fin de construir una vista minable se le 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. Los valores perdidos son valores del conjunto de datos desconocidos, sin recopilar o incorrectamente introducidos.(60)

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 consultas SQL hechas a la base de datos de ZERA.

2.3.4 Modelado

En el modelado es donde se seleccionan y aplican los algoritmos y herramientas para procesar los datos seleccionados.

Seleccionar una técnica de modelado

La técnica de minería de datos seleccionada fue el agrupamiento, ejecutando varios algoritmos con el objetivo de escoger el que arroje mejores resultados. Para definir el algoritmo a utilizar se tuvo en cuenta el tipo de dato a procesar, la calidad de las particiones y el tiempo de ejecución. Los algoritmos analizados son: K-means, EM, DBSCAN y OPTICS.

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. (61) Para cada categoría de validación existen un conjunto de medidas que demuestran cuán bueno fue el agrupamiento de un algoritmo sobre un conjunto de datos.

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 a utilizar en la investigación. Las medidas de validación interna se basan en los siguientes dos criterios (61):

- **Cohesión**: mide cuán estrechamente relacionados están los objetos en un grupo.
- **Separación**: mide lo distinto o bien separado que es un grupo de otros.

Una de las medidas de validación interna es el 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 se *s(x)* puede variar entre -1 y 1:

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

El Coeficiente de Silhouette para todo el agrupamiento es:

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

Para aplicar esta medida al resultado de la ejecución de los algoritmos seleccionados fue utilizada la herramienta MOA, 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 6: Coeficiente de Silhouette y tiempo de ejecución.

Algoritmo	Coeficiente de Silhouette	Tiempo de ejecución
K-means	0,932736059	0.01
EM	0,938020152	1.78
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 EM muestra el valor más alto seguido del K-means con una diferencia de 0,005284093. Teniendo en cuenta que esta diferencia no es significativa y que se necesita elegir un algoritmo que agrupe bien y a la vez sea rápido, se selecciona el K-means para ser utilizado.

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 7: Ejecución del K-means con diferente número de k.

Número de clúster	Coeficiente 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 8: 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

2.4 Propuesta de solución

Se propone la implementación de un módulo que permita mostrar a los profesores el comportamiento de los estudiantes en la Plataforma Educativa ZERA. El módulo debe ser capaz de agrupar a los estudiantes de acuerdo a la actividad en el foro, 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 Flot 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:

- El conector java o driver para la conexión entre WEKA y el SGBD que se esté usando.

- 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.

2.5 Fase de Exploración

En la fase de Exploración se define el alcance general del proyecto. El cliente define lo que necesita mediante la redacción de HU y los programadores estiman los tiempos de desarrollo en base a esta información. El resultado es una visión general del sistema, y un plazo total estimado.

2.5.1 Historias de Usuario (HU)

Las HU constituyen el artefacto empleado por XP para especificar los requisitos de la aplicación o sistema a implementar. Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas (62). Para documentar las HU se utiliza una plantilla con los aspectos definidos a continuación:

- Número: posee el número asignado a la HU.
- Nombre de HU: atributo que contiene el nombre de la HU.
- Usuario: usuario del sistema que utiliza o protagoniza la HU.
- Prioridad en el negocio: evidencia el nivel de prioridad de la HU en el negocio.
- Puntos estimados: estimación hecha por el equipo de desarrollo del tiempo de duración de la HU.
- Descripción: posee una descripción de lo que realizará la HU.
- Observaciones: describe aspectos a tener en cuenta para acceder a la funcionalidad.

La descripción de las demás HU puede encontrarse en el Anexo 1.

Tabla 9: Actividad de los estudiantes en el foro.

Historias de Usuario	
Número: 1	**Nombre:** Actividad de los estudiantes en el foro.
Usuario: Docente, Director	
Prioridad en el negocio: Alta	
Iteraciones asignadas: 1	**Puntos de estimación:** 2 semanas
Descripción: El usuario podrá consultar un reporte correspondiente a la actividad de los estudiantes en el foro, con opción de visualizar a través de gráficos las agrupaciones que se forman. El usuario también tiene la opción de ver una tabla con los datos de los estudiantes de cada agrupación y un resumen de los valores de los atributos por agrupación. Además podrá cambiar el tipo de gráfico entre barra y pastel.	
Observaciones: Para visualizar un reporte el usuario debe estar autenticado en el sistema y tener los permisos para poder acceder a esta funcionalidad.	

2.5.2 Aspectos no funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.(62)

Para definir los aspectos no funcionales se utilizó el estándar ISO/IEC 9126. Este estándar ha sido desarrollado en un intento de identificar los atributos claves de calidad para un producto de *software*. Identifica seis características básicas de calidad que pueden estar presentes en cualquier producto de *software*. El estándar provee una descomposición de las características en subcaracterísticas, que se muestran a continuación.(63)

Tabla 10: Propuesta de aspectos no funcionales estándar ISO/IEC 9126.

Característica	Subcaracterísticas
Funcionalidad	Adecuación Exactitud Interoperabilidad Seguridad
Confiabilidad	Madurez

	Tolerancia a fallas Recuperabilidad
Usabilidad	Comprensibilidad Capacidad de aprendizaje Operabilidad
Eficiencia	Comportamiento en tiempo Comportamiento de recursos
Mantenibilidad	Analizabilidad Modificabilidad Estabilidad Capacidad de pruebas
Portabilidad	Adaptabilidad Instalabilidad Reemplazabilidad

De lo propuesto por el estándar ISO/IEC 9126 se utilizan las características y subcaracterísticas que interesan en la investigación. Para documentar los requisitos se propone la nomenclatura definida por Len Bass, Paul Clements y Rick Kazman en el libro *"Software Architecture in Practice"*. De la nomenclatura se utilizan los aspectos definidos a continuación:

Fuente (Origen): se describe la entidad fuente que provocó el estímulo. Puede ser un humano, un sistema o cualquier otro actor.

Artefacto: se listan el/los artefacto(s) que es estimulado. Puede ser todo el sistema o alguna parte. Ejemplos: canales de comunicación, interfaz de usuario, el Sistema, el código.

Estímulo: se describe la condición que debe ser considerada como una acción externa o interna capaz de provocar una reacción en el sistema.

Respuesta: es la actividad que se realiza después de la llegada del estímulo.

La siguiente tabla muestra la descripción de uno de los aspectos no funcionales, los restantes se visualizan en el Anexo 2:

Tabla 11: ANF[11] de Comprensibilidad.

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Comprensibilidad
Objetivo	Facilidad para entender el contenido del módulo.
Origen	Humano
Artefacto	Interfaz de usuario
Estímulo	**Respuesta: flujo de eventos (Escenarios)**
1.a Reportes/Comportamiento del estudiante	
Gráficos	Descripción del contenido de los gráficos.
1.b Reportes/Comportamiento del estudiante	
Leyenda	Los gráficos poseen leyenda.
1.c Reportes/Comportamiento del estudiante	
Términos	Los nombres de las opciones del módulo están asociados a la información que se muestra.

2.6 Fase de Planificación

La Planificación es una fase corta, en la que el cliente y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y las entregas de las mismas. El resultado de esta fase es un Plan de Entregas.

2.6.1 Estimación de esfuerzo por HU

La estimación de esfuerzos por HU es una actividad de la fase de planificación. Es aquí donde los desarrolladores determinan el tiempo estimado que durará la implementación de cada HU.

Tabla 12: Estimación de esfuerzo por HU.

Historia de Usuario	Puntos de estimación
Actividad de los estudiantes en el foro.	2 semanas
Comportamiento de los estudiantes en la realización de ejercicios.	1 ½ semanas

[11] Aspecto no funcional.

Comportamiento de los estudiantes en la realización de tareas.	1 ½ semanas
Comportamiento de los estudiantes en el acceso a recursos.	2 semanas
Comportamiento general de los estudiantes en la plataforma.	2 semanas
Filtrar estudiantes.	2 semanas

2.6.2 Plan de Iteraciones

La elaboración de las HU y la estimación de los esfuerzos para la realización de cada una de ellas, sirven de base para la planificación de la implementación del sistema. Un plan de iteraciones, es el artefacto a través del cual se seleccionan las HU que serán implementadas en cada iteración del sistema.

Tabla 13: Plan de iteraciones.

Iteración	Descripción
1	Se implementará la HU_1
2	Se implementarán las HU_2, HU_3
3	Se implementará la HU_4
4	Se implementará la HU_5
5	Se implementará la HU_6

2.6.3 Plan de duración de las iteraciones

El plan de duración de las iteraciones recoge el orden de las HU que se van a desarrollar en cada iteración. Para esto se tiene en cuenta la prioridad definida por el cliente y el nivel de complejidad de las mismas. Este plan es utilizado en la presente investigación con el objetivo de organizar el trabajo.

Tabla 14: Plan de duración de las iteraciones.

Iteración	Orden de las HU	Duración Total
1	Actividad de los estudiantes en el foro.	2 semanas
2	Comportamiento de los estudiantes en la realización de ejercicios. Comportamiento de los estudiantes en la realización de tareas.	3 semanas
3	Comportamiento de los estudiantes en el acceso a recursos.	2 semanas
4	Comportamiento general de los estudiantes en la plataforma.	2 semanas
5	Filtrar estudiantes.	2 semanas

2.6.4 Plan de entregas

La fase de planificación concluye con el compromiso final del equipo de desarrollo con el cliente, generando el plan de entregas. Este plan retorna un cronograma donde se unen las funcionalidades referentes a un mismo tema o módulo, para lograr un mayor entendimiento en la implementación del sistema.

Tabla 15: Plan de entregas.

Número de Iteración	Duración Iteración	Fecha inicio	Fecha fin
Iteración 1	2 semanas	16/02/2015	28/02/2015
Iteración 2	3 semanas	2/03/2015	20/03/2015
Iteración 3	2 semanas	23/03/2015	3/04/2015
Iteración 4	2 semanas	13/04/2015	25/04/2015
Iteración 5	1 semana	27/04/2015	2/05/2015

2.7 Diseño del módulo

XP propone realizar diseños simples y sencillos, hacerlo todo lo menos complejo posible para lograr que sea entendible y fácil de implementar. Realizar una correcta especificación de los nombres de métodos y clases, ayuda a comprender mejor lo diseñado y facilita las posteriores ampliaciones y la reutilización del código.(31)

2.7.1 Tarjetas Clase-Responsabilidades-Colaboradores (CRC)

Las tarjetas CRC trabajan con la técnica de modelado basada en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades. Fueron usadas para la representación de las diferentes entidades presentes en el módulo desarrollado en este trabajo. Con ellas se estructuró las diferentes entidades así como sus responsabilidades y relaciones entre ellas. Las tarjetas están compuestas por el nombre de la clase colocado como título, en la parte izquierda se colocan las responsabilidades (funcionalidades) y en la parte derecha las clases que se implican en cada funcionalidad. La siguiente tabla muestra la tarjeta CRC_1, las demás pueden ser consultadas en el Anexo 3.(64)

Tabla 16: CRC_1.

Clase: sfReportDataMining

Responsabilidades	Colaboradores
extractDataWeka() *extractDataPercent()* *joinArrays()* *sqlGeneral()* *sqlExercise()* *sqlResources()* *sqlTask()* *sqlForum()* *max()* *min()* *compareClusterDatas()* *avg()* *auxAvg()* *percentData()* *percentToScaleNote()* *percentToScaleNoteGroup()* *rangeScaleEvaluations()* *cualitativeTrue()* *sumResourcesForGroup()* *sumExercisesForStudentGroup()*	*BacherlorReatingScale*

2.7.2 Patrón de arquitectura

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), que está formado por tres niveles (40):

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio (la base de datos pertenece a esta capa). Es aquí donde actúa el ORM Doctrine, permitiendo al desarrollador abstraerse de la base de datos. Las clases y archivos relacionados con el modelo se guardan en el directorio *lib/model/* del proyecto.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella (un motor de plantillas es parte de esta capa), o sea, es principalmente la capa de plantillas PHP. Estas son guardadas en varios directorios *templates/* dentro de las aplicaciones del proyecto.

- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Es la pieza de código que llama al modelo para obtener algunos datos que le pasa a la vista para la presentación al cliente. Todas las solicitudes son gestionadas por los controladores frontales (*index.php* y *frontend_dev.php* son los controladores frontales por defecto). Estos controladores frontales delegan la verdadera labor a las acciones de los módulos de las aplicaciones.

2.7.3 Patrones de diseño

El patrón es una descripción del problema y la esencia de su solución, 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. Los patrones de diseño empleados en la solución pertenecen al conjunto de patrones GRASP y GOF.(62) A continuación se explican los que fueron utilizados:

Patrones GRASP[12]

- **Bajo Acoplamiento:** Un elemento con bajo (o débil) acoplamiento no depende de demasiados elementos (50). El bajo acoplamiento se evidencia en el hecho de que los controladores heredan únicamente de la clase *sfActions*. Además las clases que implementan la lógica del negocio y de acceso a datos no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia entre las clases, en este caso, sea baja.(40)
- **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.(40)
- **Creador:** Este patrón indica a qué clase se le asigna la responsabilidad de la creación de instancias, puesto que esta posee la información necesaria para la creación de este objeto. Es utilizado en los controladores, en ellos se encuentran las acciones definidas para el sistema. En la

[12] Patrones Generales de Software para Asignación de Responsabilidades *(General Responsibility Assignment Software Patterns).*

implementación de las acciones se crean instancias de las clases del modelo y de los formularios que representan a estas clases.(40)

- **Controlador:** Este patrón se encarga de que una clase actúe como intermediaria para el manejo de eventos. Un controlador sirve como intermediario entre una interfaz y la acción que se desee ejecutar (50). Dentro del *framework* el patrón se evidencia en las clases que forman la capa Controlador del patrón arquitectónico MVC, entre las que figuran *sfController, sfWebController, sfFrontWebController, sfContex, sfAction* y quienes heredan de la misma (los *action* o controladores de cada módulo), y en el *index.php* del ambiente que se está ejecutando. 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.(40)

Patrones GOF

- **Singlenton[13]:** Es un patrón de tipo creación, ya que abstrae el proceso de creación de instancias. Resuelve el problema de que exista una instancia única de una clase, proporcionando un punto de acceso global a la misma (50). En Symfony se hace visible en clases como *sfContext* la cual permite interactuar con los objetos que son únicos en el núcleo del *framework*, ya que posee una referencia a cada uno de ellos; a través del método *getInstance* que posee esta clase es posible obtener una instancia de la misma. Otro ejemplo lo constituye la clase *sfRouting* la cual se encarga de enrutar todas las peticiones que se hagan a la aplicación, la misma permite a través del método *getInstance* que se obtenga la instancia única de ella existente en el núcleo del framework.(40)
- **Decorator:** Es un patrón de tipo estructura, ya que permite que clases y objetos sean utilizados para componer estructuras de mayor tamaño. Resuelve el problema de añadir responsabilidades adicionales a un objeto dinámicamente. En Symfony este patrón es utilizado en la capa Vista del patrón arquitectónico MVC, ejemplo de ello lo constituye la clase *sfView* que es padre de todas las vistas de la aplicación. El fichero *layout.php* es la plantilla global que utiliza el *framework* para decorar las plantillas asociadas a cada una de las acciones de la aplicación.(40)

2.8 Conclusiones parciales

- El uso de la metodología 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

[13] Conocido como instancia única.

los algoritmos de minería de datos, así como la evaluación de los resultados y su puesta en práctica.

- La metodología XP sirve de guía para definir los requisitos del cliente, planificar el tiempo de duración de cumplir estos requisitos e implementarlos, y probar que están hechos correctamente y de la manera que el cliente quiere.
- La evaluación de los resultados permite comprobar que los algoritmos escogidos arrojaron resultados correctos que pueden ser utilizados.

Capítulo3: Implementación y prueba

3.1 Introducción

La implementación del *software* es el resultado de todo un trabajo de investigación y análisis previo. Es esta la etapa donde se hacen realidad los requisitos del cliente. Para acompañar un desarrollo exitoso hay que hacer uso de pruebas que demuestren que lo que se implementa es correcto y que está acorde a lo que el cliente quiere. Este capítulo hace referencia a los principales elementos de estas etapas del desarrollo de *software* en la investigación.

3.2 Fase de iteraciones

Es en esta fase donde XP propone la implementación del *software*. Para esto incluye varias iteraciones sobre el sistema antes de ser entregado, generando en cada iteración el conjunto de HU que se han seleccionado para la misma. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.(31)

3.2.1 Tareas de ingeniería

Los objetivos de una iteración se cumplen al completar las funcionalidades planteadas en las HU que la componen. Las HU se derivan en una serie de tareas que marcan los pasos a seguir durante el ciclo de vida de cada iteración y, por consiguiente, del sistema en general. Estas tareas se conocen como tareas de ingeniería y constituyen un artefacto generado por la metodología XP.(31) A continuación se muestran las tareas de ingeniería asociadas a la HU_1. Las tareas correspondientes a las otras HU pueden encontrarse en el Anexo 4.

Tabla 17: TI[14] Representar grupos.

Tarea de Ingeniería	
Número: 1	**Número de HU:** 1
Nombre: Representar grupos	
Tipo de tarea: Desarrollo	**Estimación:** 4 días
Fecha de inicio: 16/02/2015	Fecha fin: 19/02/2015
Programador responsable: Roberto Martínez Navarro	

[14] Tarea de Ingeniería.

Descripción: El sistema muestra una gráfica de pastel con la cantidad de agrupaciones formadas, permitiendo que al poner el cursor en cada partición se muestre el porciento de estudiantes que pertenecen a la agrupación señalada. Además permite visualizar las agrupaciones en una gráfica de barra.

Tabla 18: TI Mostrar datos de estudiantes.

Tarea de Ingeniería	
Número: 2	**Número de HU:** 1
Nombre: Mostrar datos de estudiantes	
Tipo de tarea: Desarrollo	**Estimación:** 3 días
Fecha de inicio: 20/02/2015	**Fecha fin:** 24/02/2015
Programador responsable: Roberto Martínez Navarro	
Descripción: El sistema muestra una tabla con los datos de los estudiantes por cada agrupación formada.	

Tabla 19: TI Mostrar tabla resumen.

Tarea de Ingeniería	
Número: 3	**Número de HU:** 1
Nombre: Mostrar tabla resumen	
Tipo de tarea: Desarrollo	**Estimación:** 3 días
Fecha de inicio: 25/02/2015	**Fecha fin:** 27/02/2015
Programador responsable: Roberto Martínez Navarro	
Descripción: El sistema muestra una tabla con el rango de valores que toman los atributos en cada agrupación.	

Tabla 20: TI Realizar pruebas unitarias.

Tarea de Ingeniería	
Número: 4	**Número de HU:** 1
Nombre: Realizar pruebas unitarias.	
Tipo de tarea: Prueba	**Estimación:** 1 día

Fecha de inicio: 16/02/2015	**Fecha fin:** 27/02/2015
Responsable: Noralys Almeida Milanés	
Descripción: Se realizan pruebas unitarias a las funcionalidades implementadas.	

Tabla 21: Diseñar casos de prueba de aceptación.

Tarea de Ingeniería	
Número: 5	**Número de HU:** 1
Nombre: Diseñar casos de prueba de aceptación.	
Tipo de tarea: Prueba	**Estimación:** 1 día
Fecha de inicio: 23/02/2015	**Fecha fin:** 26/02/2015
Responsable: Noralys Almeida Milanés	
Descripción: Se diseña el caso de prueba de aceptación correspondiente a la HU_1.	

Tabla 22: Realizar pruebas de aceptación.

Tarea de Ingeniería	
Número: 6	**Número de HU:** 1
Nombre: Realizar pruebas de aceptación.	
Tipo de tarea: Prueba	**Estimación:** 1 día
Fecha de inicio: 28/02/2015	**Fecha fin:** 28/02/2015
Responsable: Noralys Almeida Milanés	
Descripción: Se realizan las pruebas de aceptación.	

3.2.2 Estándares de codificación

Son convenios para escribir código fuente en ciertos lenguajes de programación. Estos estándares facilitan el mantenimiento del código, sirven como punto de referencia para los programadores, mantienen un estilo de programación y ayudan a mejorar el proceso de codificación, haciéndolo, entre otras cosas, más eficiente. Para definir el estándar de codificación se toma como referencia lo definido en *"Pautas de Codificación, Centro FORTES"*.

- Los nombres de las clases deben declararse en notación *StudlyCaps*[15].
- Las llaves de las clases se deben abrir en la línea siguiente a donde se comenzó a declarar esta, y cerrarse en una línea después del cuerpo de la clase.
- Los nombres de los métodos deben declararse en notación *camelCase*[16].
- Las llaves de los métodos se deben abrir en la línea siguiente a donde se comenzó a declarar este, y cerrarse en una línea después del cuerpo del método.
- Las estructuras de control deben tener un espacio detrás, los métodos y las llamadas de estos no deberían tenerlo.
- Las llaves de las estructuras de control se deben abrir en la misma línea donde se comenzaron a declarar estás, y cerrarse en una línea después del cuerpo de la estructura.
- Los paréntesis de apertura en las estructuras de control no deben tener un espacio después de estos, y tampoco deben tener un espacio antes del paréntesis de cerrado.
- Las líneas en blanco se deben añadir para mejorar la legibilidad del código y para separar bloques relacionados de códigos.
- No debe existir más de una instrucción por línea.
- Se deben utilizar 4 espacios para indentar el código.
- Las palabras claves deben estar en minúscula, al igual que las constantes php *true*, *false* y *null*.
- Las instrucciones *extends* e *implements* deben declararse en la misma línea de donde se declara la clase.
- Todos los métodos deben declarar su visibilidad.
- En la lista de argumentos de los métodos no debe haber un espacio delante de las comas, y debe haber un espacio después de cada una.

3.3 Pruebas de *software*

Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. El proceso de prueba tiene dos metas distintas (62):

- Demostrar al desarrollador y al cliente que el *software* cumple con los requerimientos.

[15] Es una notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúscula.

[16] Es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúscula exceptuando la primera palabra.

- Encontrar situaciones donde el comportamiento del *software* sea incorrecto, indeseable o no esté de acuerdo con su especificación.

XP propone un desarrollo dirigido por pruebas, haciendo énfasis en la realización de pruebas unitarias encargadas de verificar el código y pruebas de aceptación orientadas a probar las funcionalidades del sistema.(31)

3.3.1 Pruebas unitarias

Las pruebas unitarias aseguran que un único componente de la aplicación produzca una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto. Las pruebas unitarias son un procedimiento para validar que una porción del código del sistema funciona apropiadamente de manera aislada.(63) XP propone que estas pruebas deben realizarse a medida que se va implementando y en el menor tiempo posible, por eso recomienda automatizarlas (31). 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:

Tabla 23: Resultado de las pruebas unitarias.

	Iteración 1	**Iteración 2**	**Iteración 3**	**Iteración 4**	**Iteración 5**	**Total**
Cantidad de pruebas	5	3	2	2	1	13

Lime notifica el resultado de las pruebas. Las siguientes imágenes muestran la ejecución de las pruebas en la iteración 1.

```
sfReports/DataMiningTest.....................................................not ok
    Failed tests: 3, 4
Failed Test                    Stat  Total   Fail  Errors  List of Failed
-------------------------------------------------------------------------------
sfReports/DataMiningTest          0      5      2       0  3 4
Failed 1/1 test scripts, 0.00% okay. 2/5 subtests failed, 60.00% okay.
```

Figura 2: Resultado de las pruebas unitarias fallidas.

Figura 3: Resultado de las pruebas unitarias correctas.

3.3.2 Pruebas de aceptación

Las pruebas de aceptación verifican el comportamiento completo del sistema o de la aplicación en cuestión. Típicamente corresponden a escenarios de uno o más casos de uso, características o historias de usuario. Pueden ser automatizadas por desarrolladores, pero la característica clave es que el usuario final debe ser capaz de reconocer el comportamiento especificado por la prueba.(63) Los casos de prueba de aceptación se encuentran en el Anexo 5, sólo se muestra el caso de prueba correspondiente a la HU Actividad en el foro.

Tabla 24: CPA[17] Actividad en el foro.

Caso de Prueba de Aceptación	
Código: CP_1	**Historia de Usuario: HU_1**
Nombre: Actividad en el foro.	
Descripción: Muestra los resultados del análisis de la actividad en el foro.	
Condiciones de Ejecución: El usuario debe estar autenticado en la plataforma.	
Pasos de ejecución	**Resultado Esperado**
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo	El sistema muestra una lista desplegable con las opciones Programa de estudio y Grupo.
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo	El sistema muestra una lista desplegable para seleccionar el nombre del Programa de estudio o el Grupo.
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo/Buscar	El sistema muestra un gráfico de pastel con las agrupaciones que se forman. Cuando el usuario pasa el cursor por el gráfico se muestra el número de la agrupación y el porciento de alumnos de esa agrupación.

[17] Caso de Prueba de Aceptación.

Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo/Buscar/Barras	El sistema permite visualizar la información en un gráfico de barras. Cuando el usuario pasa el cursor por el gráfico se muestra el número de la agrupación y el porciento de alumnos de esa agrupación.
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo/Buscar/Estudiantes por agrupación	El sistema muestra una tabla con las características de los estudiantes por agrupación.
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo/Buscar/Características generales por agrupación	El sistema muestra una tabla con el rango de valores que toma cada atributo por agrupación.
Reportes/Comportamiento del estudiante/Foro/Programa de estudio y/o Grupo/Nombre Programa de estudio o Grupo/Buscar	El sistema muestra al final de la página un gráfico con una comparación de atributos por agrupación.

3.3.3 Análisis de los resultados de las pruebas

Al concluir cada una de las iteraciones planificadas para el desarrollo de la propuesta de solución, fueron realizadas las pruebas pertinentes para realizar las entregas pactadas con el cliente. Las no conformidades detectadas fueron resueltas. La siguiente tabla muestra el resultado de dichas pruebas en cada una de las iteraciones.

Tabla 25: Cantidad de No Conformidades por iteración.

	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
Cantidad	11	8	4	5	2

3.4 Conclusiones parciales

- La definición del estilo de código a utilizar facilitó la implementación.
- La implementación hizo uso de las tareas de ingeniería definidas.

- La realización de pruebas unitarias y de aceptación constituyó una medida de avance en el desarrollo del módulo.

Conclusiones generales

- 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 de las metodologías, herramientas, lenguajes y tecnologías que permitió determinar la utilización de XP como metodología de desarrollo de *software*, CRISP-DM como metodología de minería de datos, WEKA como herramienta para el análisis de datos, PHP como lenguaje de programación, Symfony como *framework* para el desarrollo y PostgreSQL como gestor de base de datos.
- La implementación de los requisitos definidos dio como resultado un módulo que aplica técnicas de agrupamiento para describir el comportamiento de los estudiantes en la Plataforma Educativa ZERA.
- La realización de pruebas permitió comprobar el correcto funcionamiento del módulo.
- La evaluación de los algoritmos escogidos permitió seleccionar el de mejores resultados para ser utilizado en la investigación.

Recomendaciones

A partir de la investigación realizada se recomienda:

- Aplicar técnicas que permitan realizar recomendaciones a los alumnos durante su interacción con la plataforma para poder mejorar su aprendizaje.
- Utilizar técnicas que permitan predecir resultados académicos de los estudiantes.

Referencias bibliográficas

1. MOREIRA, Manuel Area and SEGURA, Jordi Adell. E-learning: enseñar y aprender en Espacios Virtuales. . 2009.

2. ÁLVAREZ, Roberto Baelo. EL e-learning, una respuesta educativa a las demandas de las sociedades del siglo XXI. *Revista de Medios y Educación*. July 2009. No. 35.

3. CLARENC, Claudio Ariel. *Analizamos 19 plataformas de e-Learning: Investigación colaborativa sobre LMS* [online]. 2013. [Accessed 4 March 2015]. ISBN 978-1-291-53343-9. Available from: https://books.google.com.cu

4. DANS, Enrique. Educación online: plataformas educativas y el dilema de la apertura. *Revista de Universidad y Sociedad del Conocimiento.* 2009.

5. DÍAZ, Sebastián Becerro. Plataformas Educativas, un entorno para profesores y alumnos. *Revista Temas para la Educación*. May 2009. No. 2.

6. FORTES. *Documento Visión del centro FORTES*. 2010.

7. SPOSITTO, Osvaldo M., ETCHEVERRY, Martín E., RYCKEBOER, Hugo L. and BOSSERO, Julio. *Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil*. 2009.

8. KANTARDZIC, Mehmed. *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, 2011. ISBN 9780470890455.

9. CORSO, Cynthia Lorena and ALFARO, Sofía Lorena. Algoritmos de Data Mining aplicados la enseñanza basada en la Web. In : . ISBN 978-987-24967-3-9.

10. RAMÍREZ, Rafael Reséndiz. *UN MODELO DE MINERÍA DE DATOS PARA LA TOMA DE DECISIONES EN POLÍTICA EDUCATIVA A PARTIR DE LAS EVALUACIONES DE EJECUCIÓN MÁXIMA A GRAN ESCALA.*

11. Educational Data Mining. [online]. [Accessed 15 March 2015]. Available from: http://www.educationaldatamining.org/

12. Definición de comportamiento. [online]. [Accessed 20 March 2015]. Available from: http://definicion.de/comportamiento/

13. Comportamiento. *Definición MX* [online]. [Accessed 12 April 2015]. Available from: http://definicion.mx/comportamiento/Definición de Comportamiento

14. LÓPEZ, César Pérez and GONZÁLEZ, Daniel Santín. *Minería de datos: técnicas y herramientas.* Paraninfo, 2007. ISBN 9788497324922.

15. XU, Rui and WUNSCH, DONALD C. *Clustering. IEEE Press Series on Computacional Intelligence.* Board, [no date]. ISBN 978-0-470-27680-8.

16. HAN, Jiawei, KAMBER, Micheline and PEI, Jian. *Data Mining, Southeast Asia Edition: Concepts and Techniques*. 2. Morgan Kaufmann, 2006. ISBN 9780080475585.

17. GARRE, Miguel, CUADRADO, Juan José, SICILIA, Miguel A., RODRÍGUEZ, Daniel and REJAS, Ricardo. Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. *Revista Española de Innovación*. 2007. Vol. 3, no. 1.

18. PASCUAL, D., PLA, F. and SÁNCHEZ, S. *Algoritmos de agrupamiento.*

19. MEJIA, Juan Camilo Giraldo and BUILES, Jovani Alberto Jiménez. Caracterización del Proceso de Obtención de Conocimiento y Algunas Metodologías para Crear Proyectos de Minería de Datos. *Revista Latinoamericana de Ingeniería de Software*. 2013.

20. Predictive Analytics, Data Mining, Self-service, Open source - RapidMiner. *rapidminer* [online]. [Accessed 20 May 2015]. Available from: https://rapidminer.com/

21. Introduction to R. *R: The R Project for Statistical Computing* [online]. [Accessed 23 March 2015]. Available from: http://www.r-project.org/

22. Weka: Data Mining with Open Source Machine Learning Software in Java. *WEKA, The University of Waikato* [online]. [Accessed 23 March 2015]. Available from: http://www.cs.waikato.ac.nz/ml/weka/

23. Machine Learning Project at the University of Waikato in New Zealand. [online]. [Accessed 20 May 2015]. Available from: http://www.cs.waikato.ac.nz/ml/index.html

24. RStudio. *RStudio* [online]. [Accessed 2 March 2015]. Available from: http://www.rstudio.com/products/rstudio/#Server

25. ARMAS, Elvismary Molina de, MERIÑO, Mario Pupo, VÁZQUEZ, Maikel Y. Leyva, CHÁVEZ, María del Carmen and GRAU, Ricardo. SLD096 EXPERIENCIA DE DESARROLLO DE UN SOFTWARE A LA MEDIDA PARA LA CLASIFICACIÓN EN DOMINIOS BIOLÓGICOS Y BIOMÉDICOS UTILIZANDO WEKA. In : . La Habana, Cuba, 2013. ISBN 978-959-7213-02-4.

26. MORALES, Cristóbal Romero, SOTO, Sebastián Ventura and MARTÍNEZ, Cesar Hervás. Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web. In : . Escuela Politécnica Superior. Universidad de Córdoba., 2005. ISBN 84-9732-449-8.

27. ALBERCA, Greyson P, AGILA, Martha V, LOJA, Fausto J, VALDIVIEZO, Priscila M and JIMÉNEZ, Juan C. *Recolección de datos de interacción de alumnos en una plataforma E-learning para obtener indicadores de interés de su actividad aplicando técnicas de aprendizaje automático.* 2009.

28. ANAYA, A. and BOTICARIO, J.G. Clustering Learners according to their Collaboration. In : *Proceedings of the 13th international conference on computer supported cooperative work in design (CSCWD 2009)*. IEEE Computer Society Press, 2009.

29. PRESSMAN and ROGER S. *Ingeniería del software: un enfoque práctico* [online]. Mikel Angoar, 1997. [Accessed 2 February 2015]. Available from: http://www.google.com.cu/books?hl=en&lr=&id=8UV5jxkuBZIC&oi=fnd&pg=PP3&dq=metodologias+de+desarrollo+de+software&ots=wJPq-SMoDJ&sig=JpJuctheai5dJiCdMAdPE24q2v8&redir_esc=y#v=onepage&q=metodologias%20de%20desarrollo%20de%20software&f=false

30. FIGUEROA, Roberth G., SOLÍS, Camilo J. and CABRERA, Armando A. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.* Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, [no date].

31. BECK, Kent. *Extreme Programming Explained*. 1. 1999. ISBN 0201616416.

32. COBO, Ángel. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web.* Ediciones Díaz de Santos, 2005. ISBN 9788479787066.

33. MATEU, Carles. *Desarrollo de aplicaciones web.* [online]. 1. Universitat Oberta de Catalunya. Barcelona., 2004. [Accessed 3 February 2015]. ISBN 84-9788-118-4. Available from: http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo_de_Aplicaciones_Web.pdf

34. ALVAREZ, Miguel Angel. *Manual de CSS 3.* [online]. [Accessed 3 February 2015]. Available from: http://www.desarrolloweb.com/manuales/css3.html

35. JAVIER EGUILUZ. *Introducción a JavaScript.* [online]. 2009. [Accessed 3 February 2015]. Available from: http://www.librosweb.es/javascript

36. What is jQuery? *jQuery* [online]. [Accessed 24 February 2015]. Available from: https://jquery.com/

37. PHP: ¿Qué es PHP? - Manual. *PHP: Hypertext Preprocessor* [online]. [Accessed 20 February 2015]. Available from: http://php.net/manual/es/intro-whatis.php

38. Introducing Symfony. [online]. [Accessed 24 May 2015]. Available from: http://symfony.com/legacy/doc/gentle-introduction/1_4/en/01-Introducing-Symfony

39. Doctrine Project. *Doctrine* [online]. [Accessed 4 February 2015]. Available from: http://www.doctrine-project.org/index.html

40. POTENCIER, Fabien and ZANINOTTO, Francois. *Symfony 1.4, la guía definitiva*. [no date]. ISBN 9782918390305.

41. NetBeans IDE Features. *NetBeans* [online]. [Accessed 24 May 2015]. Available from: https://netbeans.org/features/index.html

42. Advantages. *PostgreSQL* [online]. [Accessed 24 May 2015]. Available from: http://www.postgresql.org/about/advantages/

43. Features. *pgAdmin* [online]. [Accessed 24 May 2015]. Available from: http://www.pgadmin.org/features.php

44. About the Apache HTTP Server Project. *Apache HTTP Server Project* [online]. [Accessed 24 May 2015]. Available from: http://httpd.apache.org/ABOUT_APACHE.html

45. What is JS Charts? *JS Charts* [online]. [Accessed 20 May 2015]. Available from: http://www.jscharts.com/

46. Compare Everything. *SocialCompare* [online]. [Accessed 20 May 2015]. Available from: http://socialcompare.com/en

47. Flot Examples. *Flot* [online]. [Accessed 24 May 2015]. Available from: http://www.flotcharts.org/flot/examples/

48. Introduction to UML. *Unified Modeling Language (UML)* [online]. [Accessed 24 May 2015]. Available from: http://www.uml.org/

49. *Sitio Oficial de Visual Paradigm* [online]. [Accessed 10 December 2014]. Available from: http://www.visual-paradigm.com

50. LARMAN, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Pearson Educación, 2003. ISBN 9788420534381.

51. HUAPAYA, Constanza R., LIZARRALDE, Francisco A., ARONA, Graciela M. and MASSA, Stella M. *Minería de Datos Educacional en Ambientes Virtuales de Aprendizaje*. 2012.

52. HUANG, Chenn-Jung, LIU, Ming-Chou, CHU, San-Shine and CHENG, Chih-Lun. An intelligent learning diagnosis system for Web-based thematic learning platform. *Computers & Education*. May 2007. Vol. 48, no. 4, p. 658–679. DOI 10.1016/j.compedu.2005.04.016.

53. YEH, Shiou-Wen and LO, Jia-Jiunn. Assessing metacognitive knowledge in web-based CALL: a neural network approach. *Computers & Education*. February 2005. Vol. 44, no. 2, p. 97–113. DOI 10.1016/j.compedu.2003.12.019.

54. HUANG, Mu-Jung, HUANG, Hwa-Shan and CHEN, Mu-Yen. Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications*. October 2007. Vol. 33, no. 3, p. 551–564. DOI 10.1016/j.eswa.2006.05.019.

55. PIRAMUTHU, S. Knowledge-based web-enabled agents and intelligent tutoring systems. *IEEE Transactions on Education*. November 2005. Vol. 48, no. 4, p. 750–756. DOI 10.1109/TE.2005.854574.

56. PAQUETTE, Gilbert, LEONARD, Michel, LUNDGREN-CAYROL, Karin, MIHAILA, Stefan and GAREAU, Denis. Learning design based on graphical knowledge-modeling. *Journal of Educational Technology and Society*. 2006. Vol. Special issue on Learning Design, January 2006, p. 97–112.

57. CHEN, Chih-Ming and DUH, Ling-Jiun. Personalized web-based tutoring system based on fuzzy item response theory. *Expert Systems with Applications*. May 2008. Vol. 34, no. 4, p. 2298–2315. DOI 10.1016/j.eswa.2007.03.010.

58. MINAEI-BIDGOLI, B., KASHY, D.A., KORTEMEYER, G. and PUNCH, W.F. Predicting student performance: an application of data mining methods with an educational Web-based system. In : *Frontiers in Education, 2003. FIE 2003 33rd Annual*. November 2003. p. T2A–13.

59. JING, Yongjun, ZHONG, Shaochun, LI, Xin, LI, Jinan and CHENG, Xiaochun. Using Instruction Strategy for a Web-Based Intelligent Tutoring System. In : *Technologies for E-Learning and Digital Entertainment* [online]. Springer Berlin Heidelberg, 2006. p. 132–139. Lecture Notes in Computer Science, 3942. [Accessed 24 April 2015]. ISBN 978-3-540-33423-1. Available from: http://link.springer.com/chapter/10.1007/11736639_18

60. Análisis de valores perdidos. *IBM Knowledge Center* [online]. 1 January 2013. [Accessed 20 May 2015]. Available from: http://www-01.ibm.com/support/knowledgecenter/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/mva/idh_miss.htm?lang=es

61. LIU, Yanchi, LI, Zhongmou, XIONG, Hui, GAO, Xuedong and WU, Junjie. Understanding of Internal Clustering Validation Measures. In : *IEEE International Conference on Data Mining*. 2010. ISBN 15504786.

62. SOMMERVILLE, Ian. *Ingeniería de Software*. 9. Pearson Educación, 2011. ISBN 978-607-32-0603-7.

63. PRESSMAN, Roger S. *Ingeniería del software. Un enfoque práctico*. 6. Estados Unidos, 2005. ISBN 9701054733.

64. GÓMEZ, Alveiro Rosado, DUARTE, Alexander Quintero and GUEVARA, Cesar Daniel Meneses. Desarrollo ágil de software aplicando programación extrema. . 2014. Vol. 5, no. 1.