

Universidad de las Ciencias Informáticas
Facultad 4



Título: “Solución informática para la extracción de conocimiento en los registros del módulo Resultados de la colección El Navegante”

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

Autores:

Gilberto Benítez Gautier
Yunior Baez Quevedo

Tutor(es):

Msc. Liana Isabel Araujo Pérez
Ing. Osmany Montes de Oca Rodríguez
Ing. Angel Alberto Vazquez Sánchez

La Habana, Cuba
Año 55 de la Revolución

Declaración de la Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 11 días del mes de junio del año 2013.

Gilberto Benítez Gautier

Autor

Yunior Baez Quevedo

Autor

Msc. Liana Isabel Araujo Pérez

Tutor

Ing. Osmany Montes de Oca Rodríguez

Tutor

Ing. Angel Alberto Vazquez Sánchez

Tutor

Dedicatoria

De Gilberto Benítez Gautier:

A la memoria de mi abuela Paula, por ser la persona más importante en mi vida, por ser mi guía, ahora y siempre.

A mi mamá, por ser mi ejemplo, por darme su amor, comprensión, apoyo y ayudarme a convertirme en lo que soy.

A mi hermanita, por todo su apoyo en los momentos difíciles, por creer en mí, por ayudarme a ser mejor persona.

Al resto de mi familia, porque me ayudaron muchísimo, y me apoyaron en todo momento.

De Yunior Baez Quevedo:

A toda mi familia por apoyarme de forma incondicional.

En especial a mis padres, por guiarme al camino correcto, por darme la educación adecuada para ser hoy lo que soy y ser las personas más importante en mi vida.

Agradecimientos

A mis padres y mi hermano por hacer posible esta graduación y darme su apoyo incondicional en cualquier decisión que tomada.

A mi tío Níco por ser un ejemplo a seguir a lo largo de toda mi vida.

A mi tía Martha, Evelio, Evelito y Mayelín por ayudarme y acogerme en su hogar.

A mis abuelos y mis tíos Omar, Xiomara, Rafael, y Keki por preocuparse tanto por mí y ayudarme en todo lo que han podido.

A mi novia Yanisleydis por acompañarme en gran parte de la carrera y darme todo el amor del mundo, por darme muchas fuerzas para seguir adelante siempre que lo necesité, te quiero mucho.

A mis suegros por acogerme en su familia como un miembro más y darme su afecto y cariño.

A mi compañero de tesis que más que un compañero ha sido un amigo que me ha ayudado desde el inicio de la carrera.

A mis compañeros de apartamento por ser mi familia durante estos años y ayudarme cuando lo necesité.

A todos mis compañeros de aulas con los que compartí momentos difíciles e inolvidables y que me ofrecieron su ayuda incondicional.

A TODOS ELLOS MUCHAS GRACIAS

Yunior

Agradecimientos

A los dos amores de mi vida, mi mamá y mi hermanita, gracias a ellas llegué tan lejos.

A mi tío Tony, mi tía Mayra, mis primas Mayliet y Maibet, por ayudarme en todo momento.

A mis tíos y primos, a Maido, Pedri, Emilín, tío Nené, tía Nora, y el resto de mi familia por su ayuda incondicional.

A mi compañero de tesis por ser mi amigo y mi hermano en todo momento durante estos 5 años.

A mis compañeros de apartamento por ser como mi familia de la UCI.

A mi amigo Rafe, todos mis amigos y amigas que han estado conmigo en las buenas y en las malas.

A todos mis compañeros de aula, que vivieron conmigo inolvidables momentos de preocupaciones y alegrías.

Muchas gracias a todos

Gilberto

A los tutores Liana, Osmany y Angel, por ayudarnos tanto y ser parte de cada momento difícil por el que pasamos.

A Katia por su ayuda incondicional y su apoyo en el momento que más lo necesitamos.

A nuestros profesores a lo largo de estos cinco años que contribuyeron a nuestra formación.

A todo los que de una forma u otra hicieron posible la realización de este trabajo.

Autores

Resumen

Con el adelanto de la informatización en varias esferas de la sociedad y el perfeccionamiento acelerado de las Tecnologías de la Información y las Comunicaciones (TIC), el desarrollo de *software* se ha convertido en un elemento de gran importancia para la sociedad actual. Su impacto en la educación ha transformado los métodos tradicionales de enseñanza, aprovechando los beneficios que brindan las TIC para un mejor desarrollo del proceso de enseñanza-aprendizaje (PEA). Muchos de estos productos almacenan en sus registros la interacción de los estudiantes con la aplicación. El volumen de estos registros dificulta la obtención de información necesaria para la toma de decisiones de los profesores. La minería de datos, se presenta como una tecnología de apoyo para extraer información y conocimiento útil contenido en grandes volúmenes de datos. En este sentido, la presente investigación aborda el desarrollo de una solución informática que permita a una persona, con conocimientos mínimos de minería de datos, realizar un proceso de extracción de conocimiento (KDD, *Knowledge Discovery in Databases*), que sirva de ayuda en la toma de decisiones. Se utilizó como metodología para guiar el KDD a Crisp-DM, apoyados en la herramienta Weka para la extracción de conocimientos. El proceso de desarrollo del sistema fue conducido por la metodología RUP, transitando por los flujos de trabajo: modelo del negocio, requisitos, análisis y diseño, implementación y pruebas; obteniéndose cada uno de los artefactos propuestos.

Palabras claves: KDD, minería de datos, proceso de enseñanza-aprendizaje.

Índice de Contenidos

Introducción	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción	5
1.2 Módulo Resultados	5
1.3 Definición de minería de datos	6
1.4 Características de la minería de datos.....	7
1.5 Fases del proceso de minería de datos	8
1.5.1 Filtrado de datos.....	8
1.5.2 Selección de variables.....	8
1.5.3 Algoritmos de Extracción de Conocimiento.....	8
1.5.4 Interpretación y evaluación.....	9
1.6 Tareas realizadas en el proceso de minería de datos.....	9
Clasificación	10
Regresión o estimación	10
Agrupamiento o <i>clustering</i>	10
Correlaciones	10
Reglas de asociación	10
Reglas de asociación secuenciales.....	11
1.7 Análisis de soluciones similares existentes.....	11
1.7.1 Minería de datos aplicada a entornos educativos	11
1.7.2 Minería de datos relacionada a Entornos Educativos en el mundo	13
1.7.3 Minería de datos relacionada a Entornos Educativos en Cuba	14
1.8 Metodología para el desarrollo de proyectos de minería de datos	15
1.9 Metodología para el desarrollo de <i>software</i>	17
1.10 Tecnologías y herramientas a utilizar.....	18
1.10.1 Herramientas usadas en el proceso de minería de datos	18
1.10.2 <i>Frameworks</i> de desarrollo	21

Índice de contenidos

1.10.3 Lenguaje de modelado	22
1.10.4 Herramienta Case	22
1.10.5 Lenguajes de programación web:	23
1.10.5.1 Lenguajes del lado del cliente:	23
1.10.5.2 Lenguajes del lado del servidor:	24
1.10.6 Servidor web	25
1.10.7 Sistemas de gestión de bases de datos	26
1.10.8 Entorno de desarrollo integrado (IDE)	26
1.11 Conclusiones del capítulo	27
CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA.....	29
2.1 Introducción	29
2.2 Modelo de dominio.....	29
2.2.1 Análisis de los conceptos del dominio	29
2.2.2 Diagrama del modelo de dominio	31
2.3 Especificación de requisitos.....	31
2.3.1 Requisitos funcionales.....	31
2.3.2 Requisitos no funcionales.....	32
2.4 Modelo de casos de uso del sistema	34
2.4.1 Actores del sistema	34
2.4.2 Diagrama de casos de uso	35
2.5 Descripción de los casos de uso del sistema	35
2.6 Fundamentación de la propuesta de solución	36
2.6.1 Comprensión del negocio	37
2.6.2 Comprensión de los datos	37
2.6.3 Preparación de los datos	38
2.7 Resultados y evaluación	39
2.7.1 Modelado.....	40
2.7.2 Evaluación.....	44
2.7.3 Despliegue	45

Índice de contenidos

2.8 Conclusiones del capítulo	45
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	46
3.1 Introducción	46
3.2 Análisis del sistema.....	46
3.2.1 Modelo de análisis	46
3.2.2 Diagrama de clases del análisis	46
3.3 Diagrama de colaboración	47
3.4 Arquitectura del diseño	48
3.4.1 Patrones arquitectónicos	48
3.4.2 Patrones de diseño.....	48
3.5 Modelo de diseño.....	49
3.5.1 Diagrama de clases del diseño.....	49
3.6 Diagrama de secuencia	52
3.7 Diseño de la base de datos	55
3.7.1 Modelo Entidad-Relación.....	55
3.8 Conclusiones del capítulo	56
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.....	57
4.1 Introducción	57
4.2 Modelo de implementación	57
4.2.1 Diagrama de componentes.....	57
4.2.2 Diagrama de despliegue.....	60
4.3 Estándares de codificación	61
4.3.1 Convenciones de nombres (Funciones y Clases)	61
4.3.2 Indentación y espacios en blanco.....	62
4.3.3 Condicionales if, for, while, switch	62
4.3.4 Comentarios	62
4.4 Interfaces de usuario	63
4.5 Pruebas de <i>software</i>	63

Índice de contenidos

4.5.1 Niveles y tipos de prueba	63
4.5.2 Métodos de pruebas	64
4.5.2.1 Pruebas de caja blanca.....	64
4.5.2.2 Pruebas de caja negra	64
4.5.3 Técnicas de pruebas	64
4.6 Conclusiones del capítulo	65
Conclusiones generales.....	67
Recomendaciones	68
Bibliografía.....	69
Glosario de términos.....	73
Anexos.....	74

Índice de figuras:

Ilustración 1: Ciclo interactivo del PEA.....	12
Ilustración 2: Modelo de dominio.....	31
Ilustración 3: Diagrama de casos de uso del sistema.....	35
Ilustración 4: Ejemplo de fichero ARFF	39
Ilustración 5: CU1: Realizar proceso de extracción de conocimientos.....	47
Ilustración 6: Diagrama de colaboración CU1: Realizar proceso de extracción de conocimiento	47
Ilustración 7: Diagrama de clases del diseño: CU Realizar proceso de extracción de conocimiento	51
Ilustración 8: Diagrama de clases del diseño: CU Mostrar resultados Agrupamiento	51
Ilustración 9: Diagrama de clases del diseño: CU Mostrar resultados Reglas	52
Ilustración 10: Diagrama de clases del diseño: CU Generar reporte	52
Ilustración 11: Diagrama de secuencia CU Realizar proceso de extracción de conocimiento	53
Ilustración 12: Diagrama de secuencia CU Mostrar Resultados Agrupamiento: Escenario 1	53
Ilustración 13: Diagrama de secuencia CU Mostrar Resultados Agrupamiento: Escenario 2	54
Ilustración 14: Diagrama de secuencia CU Mostrar Resultados Reglas.....	54
Ilustración 15: Diagrama de secuencia CU Generar reporte	55
Ilustración 16: Diagrama Entidad-Relación	56
Ilustración 17: Diagrama de componentes del sistema	58
Ilustración 18: Diagrama de componentes del paquete actions.....	58
Ilustración 19: Diagrama de componentes del paquete Clases Js	58
Ilustración 20: Diagrama de componentes del paquete CSS	59
Ilustración 21: Diagrama de componentes del paquete HTML	59
Ilustración 22: Diagrama de componentes del paquete Lib	60
Ilustración 23: Diagrama de despliegue	61
Ilustración 24: CU2: Mostrar resultados Agrupamiento	82
Ilustración 25: CU3: Mostrar resultados Reglas	82
Ilustración 26: CU4: Generar reporte	82
Ilustración 27: Diagrama de colaboración CU2: Mostrar resultados Agrupamiento	83
Ilustración 28: Diagrama de colaboración CU3: Mostrar resultados Reglas	83
Ilustración 29: Diagrama de colaboración CU4: Generar reportes	84
Ilustración 30: Interfaz de selección de los algoritmos	85
Ilustración 31: Interfaz del algoritmo Kmeans.....	86
Ilustración 32: Interfaz del algoritmo Apriori	87

Índice de tablas

Tabla 1: Ejemplo de minería de datos en los entornos educativos	15
Tabla 2: Actores del sistema	34
Tabla 3: CU Realizar proceso de extracción de conocimiento.....	35
Tabla 4: CU Mostrar resultados Agrupamiento	36
Tabla 5: CU Mostrar resultados Reglas	36
Tabla 6: CU Generar reportes.....	36
Tabla 7: Clases del diseño.....	50
Tabla 8: Estrategia de pruebas	65
Tabla 9: No conformidades encontradas.....	65
Tabla 10: Descripción de CU 1 Realizar proceso de extracción de conocimiento	78
Tabla 11: Descripción de CU 2 Mostrar resultados Agrupamiento	79
Tabla 12: Descripción de CU 3 Mostrar resultados Reglas	80
Tabla 13: Descripción de CU 4 Generar reporte	81
Tabla 14: Descripción de variables	88
Tabla 15: Caso de prueba CU1 Realizar proceso de extracción de conocimiento.....	90
Tabla 16: Caso de prueba CU1 Realizar proceso de extracción de conocimiento.....	91
Tabla 17: Caso de prueba CU1 Realizar proceso de extracción de conocimiento.....	93
Tabla 18: Caso de prueba CU1 Realizar proceso de extracción de conocimiento.....	94
Tabla 19: Caso de prueba CU 2 Mostrar resultados Agrupamiento	96
Tabla 20: Caso de prueba CU 3 Mostrar resultados Reglas	97
Tabla 21: Caso de prueba CU 4 Generar reporte.....	97

Introducción

En las últimas décadas, con el adelanto de la informatización en las diferentes esferas de la sociedad y el perfeccionamiento acelerado y continuo de las Tecnologías de la Información y las Comunicaciones (TIC), el desarrollo de *software* se ha convertido en un elemento de gran importancia para la sociedad actual. Su impacto se manifiesta en diversos ámbitos de la actividad humana, contribuyendo al mejoramiento de la calidad de vida de las personas y cambiando su forma de actuar para con la sociedad. El desarrollo de *software* para la educación ha transformado los métodos tradicionales de enseñanza, aprovechando los beneficios que brindan las TIC, para un mejor desempeño del proceso de enseñanza-aprendizaje y la obtención de resultados superiores en el ámbito educacional.

En Cuba una de las estrategias para el progreso tecnológico paulatino de la sociedad está dirigida hacia la creación de productos de *software* que aporten soluciones novedosas y constituyan pasos importantes hacia la digitalización del país. Para lograrlo cuenta con la Universidad de las Ciencias Informáticas (UCI) la cual tiene entre sus principales objetivos proveer productos informáticos que posibiliten el creciente desarrollo tecnológico del país. La universidad se encuentra organizada en varios centros de desarrollo, entre ellos se encuentra el Centro de Tecnologías para la Formación (FORTES), encargado del desarrollo de soluciones informáticas que apoyen el proceso de formación. Un ejemplo lo constituye el proyecto Multisaber-Navegante cuyo objetivo es la creación de las colecciones de *software* educativo Multisaber, dirigida a los estudiantes de la enseñanza primaria y El Navegante destinada a la enseñanza secundaria, del Sistema Educativo Cubano.

El Navegante, específicamente en su versión multiplataforma, está basado en el concepto de hiperentorno de aprendizaje definido como: "...un sistema informático basado en tecnología hipertexto que contiene una mezcla o elementos representativos de diversas tipologías de *software* educativo que, utilizados de forma oportuna y acertada, pueden contribuir al manejo de diferentes estructuras lingüísticas" (1). La colección está formada por 10 productos que se especializan en varias materias, cada uno de estos integrado por seis módulos, brindando la facilidad de realizar diversas acciones y mezclando los diferentes elementos que representan las tipologías de *software* educativo.

Uno de los componentes más importantes de la colección es el módulo Resultados, encargado de administrar la información para medir el avance de los estudiantes a través de reportes, mostrando una serie de datos e informaciones sobre cómo se ha comportado la interacción de los alumnos con cada uno de los restantes módulos de la colección. A través de estos reportes se realizan análisis estadísticos

utilizando gráficas comparativas de efectividad. De esta manera el profesor puede consultar las evaluaciones y mostrar el recorrido de sus estudiantes.

La colección El Navegante permite almacenar grandes volúmenes de registros de la interacción de los estudiantes con cada módulo de la aplicación. Dichos registros pueden contener información valiosa como patrones de comportamiento y características de los estudiantes, pero imperceptible a las personas debido a los grandes volúmenes de información.

A pesar de las funcionalidades presentes en el módulo Resultados, este no cuenta con una funcionalidad que facilite la realización de un KDD, que permita extraer patrones de comportamiento y características a partir de grandes volúmenes de registros presentes en la aplicación. Esto no les permite a los educadores realizar un análisis exhaustivo de los registros de los estudiantes en la aplicación, que les posibilitaría tener un mayor conocimiento del comportamiento de los alumnos, lo cual serviría de apoyo para la preparación de los profesores en el proceso docente educativo.

A partir de la problemática descrita surge como **problema a resolver** en la investigación la siguiente interrogante: ¿Cómo realizar un proceso de extracción de conocimiento, que permita obtener patrones de comportamiento y características a partir de grandes volúmenes de registros presentes en la colección El Navegante?

Para resolver dicho problema se ha identificado como **objetivo general** desarrollar una solución informática para la extracción de conocimiento en los registros del módulo Resultados de la colección El Navegante.

Para lograr el objetivo general se tendrán en cuenta los siguientes objetivos específicos:

1. Analizar los aspectos teóricos que sustenten el procesamiento de datos, abarcando las principales técnicas, herramientas y procesos que se utilizan a nivel mundial.
2. Implementar una solución informática que permita obtener patrones de comportamiento y características a partir de los registros almacenados en El Navegante.
3. Evaluar la calidad de la solución desarrollada.

A partir del problema identificado se ha centrado el estudio en el procesamiento de los datos almacenados mediante la utilización del proceso de minería de datos, lo cual constituye el **objeto de estudio**.

Al haber planteado los elementos fundamentales que describen el problema a resolver y los objetivos sobre el cual se le dará solución, se puede determinar que el **campo de acción** tendrá su incidencia en el

procesamiento de los datos almacenados mediante la utilización del proceso de minería de datos aplicado a los hiperentornos de aprendizaje.

Luego de un análisis de la problemática, se plantea la siguiente **idea a defender**: El desarrollo de una solución informática que facilite la ejecución de un proceso de extracción de conocimiento, utilizando técnicas de minería de datos adecuadas, a partir de los datos generados en la colección El Navegante, permitirá a los profesores identificar patrones de comportamiento, tendencias y características de los estudiantes que ayuden en la toma de decisiones.

Las **tareas de la investigación** que se proponen para dar solución a los objetivos anteriormente planteados son:

- Estudio de las técnicas, herramientas y metodologías a emplear en el procesamiento de datos.
- Análisis de las soluciones similares existentes.
- Identificación de una técnica para solucionar el problema en cuestión.
- Análisis y diseño de la solución propuesta.
- Implementación de la solución propuesta.
- Diseño de pruebas de funcionalidad.
- Validación funcional del resultado obtenido, mediante las pruebas diseñadas para el sistema.

Durante el desarrollo de la investigación la aplicación de **métodos científicos** contribuyó a la obtención de nuevos conocimientos y su aplicación para la elaboración de la propuesta de solución.

Métodos Teóricos:

- **Análítico-sintético:** Como parte fundamental de la investigación fue necesario realizar un análisis de los conceptos fundamentales relacionados con el procesamiento de los datos almacenados mediante la utilización del proceso de minería de datos, así como, la revisión de soluciones similares.

- **Análisis histórico-lógico:** Se realiza un análisis de la trayectoria de la colección El Navegante, específicamente el módulo Resultados, enfocada fundamentalmente en sus características y reportes.

El documento se estructura en cuatro capítulos en los que se describen los procesos de la investigación y desarrollo del sistema. Además incluye la bibliografía consultada, el glosario de términos y los anexos.

Capítulo 1: Fundamentación teórica. En este capítulo se realiza un análisis de las características y los reportes generados por el módulo Resultados, así como, los conceptos fundamentales relacionados con el objeto de estudio. Se analizan además, las soluciones similares existentes en el ámbito nacional e internacional, las tecnologías y herramientas a utilizar, durante la investigación y desarrollo del sistema.

Capítulo 2: Descripción del sistema. Se describen los conceptos fundamentales relacionados con los procesos del dominio reflejados a través del diagrama del modelo de dominio. Se puntualizan los requisitos funcionales y no funcionales del sistema, así como, los casos de uso definidos y sus correspondientes descripciones. Se muestra además la fundamentación de la propuesta de solución reflejada en el desarrollo de la metodología Crisp-DM a través de las diferentes fases que esta propone.

Capítulo 3: Análisis y diseño del sistema. En este capítulo se exponen los elementos fundamentales propuestos por el flujo de trabajo de análisis y diseño de la metodología RUP. Además, se realiza un análisis de los patrones arquitectónicos y de diseño presentes en la aplicación, así como, el diseño de la base de datos de la colección.

Capítulo 4: Implementación y pruebas. En este capítulo se describen los elementos relacionados con el desarrollo de la propuesta de solución, apoyados en artefactos como diagramas de componentes y de despliegue. Se exponen los estándares de código utilizados en la implementación del sistema y las principales interfaces de la aplicación. Se realizan además las pruebas al sistema que permitan identificar errores en el funcionamiento del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se exponen las ideas y funcionalidades principales asociadas al módulo Resultados de la colección El Navegante, destacando los reportes generados por el mismo y sus aplicaciones. Además se brinda una visión general de las diferentes técnicas para el análisis de datos; enmarcándose en un contexto de inteligencia artificial enfocado específicamente en la minería de datos, así como se abordan conceptos asociados al dominio del problema que permiten tener una mejor comprensión del mismo. Se realiza un análisis de las aplicaciones basadas en técnicas de minería de datos en el ámbito nacional e internacional. Se procede a caracterizar los fundamentos y fases de los procesos de minería de datos; así como, tareas, métodos, componentes y principales herramientas a utilizar en estos procesos.

En este capítulo se expone además la relación de la minería de datos con la esfera educacional y las principales aplicaciones en dicha rama. Además se muestran elementos que caracterizan la metodología de desarrollo, tecnologías y herramientas utilizadas para el desarrollo de la propuesta.

1.2 Módulo Resultados

El módulo Resultados de la colección El Navegante fue desarrollado con el fin de brindar funcionalidades para facilitar un mejor seguimiento y control de la trayectoria de los estudiantes en su interacción con la aplicación. Con el desarrollo de este módulo se logra obtener información importante sobre los estudiantes que puede ser utilizada por el profesor para una mejor estructuración del proceso de aprendizaje. Se brinda la posibilidad de acceder a las evaluaciones obtenidas por los estudiantes en su interacción con la aplicación, así como, mostrar la evolución de los mismos en las asignaturas y contenidos especificados en el sistema.

El módulo Resultados posibilita llevar un control detallado de las actividades realizadas por los estudiantes a través de cuatro reportes:

Trayectoria del estudiante:

Presenta información consolidada de la interacción del estudiante en una sesión de trabajo con la colección.

Análisis de los contenidos específicos:

Facilita la consulta de las calificaciones que el estudiante va obteniendo de forma paulatina, con los cuestionarios interactivos del módulo Ejercicios, en función de los contenidos.

Historial del estudiante:

Se analiza a nivel general los resultados de la interacción con los ejercicios de todos los productos de la colección en función de las asignaturas.

Análisis integral del estudiante:

Presenta un análisis de los resultados de la interacción de los estudiantes con los contenidos, tomando en cuenta sus resultados en función de las fechas en que los estudiantes interactuaron con los ejercicios.

A pesar de todas las funcionalidades y reportes significativos brindados por el módulo Resultados, este no es capaz de realizar un proceso de extracción de conocimiento a partir de los datos generados en la interacción con la aplicación.

Surge la necesidad del uso de técnicas de minería de datos con fin de agrupar, identificar o descubrir ciertos patrones y características que puedan ser relevantes para los profesores y así perfeccionar el proceso de enseñanza y aprendizaje en función de los esquemas detectados. Además con la incorporación de estas técnicas se pueden predecir comportamientos o resultados futuros de los estudiantes, lo cual sirve de apoyo para la preparación de los profesores en el proceso docente educativo y elevar la calidad del mismo.

1.3 Definición de minería de datos

La minería de datos es un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos (2). Se puede definir además la minería de datos como un proceso completo de aplicar métodos basados en la informática para descubrir conocimiento oculto en grandes cantidades de información (3).

Se adopta como concepto de la investigación el expresado por Gregory Piatetsky-Shapiro, cuando se refieren a la minería de datos como:

“... un conjunto de técnicas y herramientas aplicadas al proceso no trivial de extraer y presentar conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a

partir de grandes conjuntos de datos, con objeto de predecir, de forma automatizada, tendencias o comportamientos y descubrir modelos previamente desconocidos” (4).

1.4 Características de la minería de datos

La minería de datos es un proceso para extraer conocimiento útil y comprensible que se encuentre oculto entre grandes volúmenes de información. Dentro de las características más significativas se encuentran:

- Investigar la información en las profundidades de las bases de datos, que muchas veces contienen información almacenada durante varios años y esta información nunca ha sido procesada ni analizada.
- Los datos pueden ser encontrados en almacenes de datos, mercados de datos, así como, en servidores de *internet* e *intranet* (5).
- El entorno de la minería de datos suele tener una arquitectura cliente-servidor.
- Las herramientas para desarrollar los procesos de la minería de datos ayudan a extraer la esencia de la información enterrada en archivos corporativos, registros públicos o archivados en cualquier tipo de base de datos.
- El minero es, muchas veces un usuario final con poca o ninguna habilidad de programación, facultado por barrenadoras de datos y otras poderosas herramientas indagatorias para efectuar preguntas y obtener rápidamente respuestas (5).
- Hurgar y sacudir a menudo implica el descubrimiento de resultados valiosos e inesperados (5).
- Las herramientas de la minería de datos constituyen una poderosa arma capaz de combinarse fácilmente, analizarse y procesarse rápidamente.
- Generalmente debido a la gran cantidad de datos que se procesan se hace necesario usar procesamiento en paralelo para la minería de datos.
- La minería de datos produce cinco tipos de información (5):
 - Asociaciones
 - Secuencias
 - Clasificaciones
 - Agrupamientos

- Pronósticos

1.5 Fases del proceso de minería de datos

Los procesos de minería de datos están guiados por un conjunto de fases que miden su cumplimiento y brindan una mayor organización y estructuración con el fin de agilizar y facilitar el proceso. Las fases mencionadas son (5):

- Filtrado de datos
- Selección de variables
- Extracción de conocimiento
- Interpretación y evaluación

1.5.1 Filtrado de datos

La información que se encuentra almacenada en los almacenes de datos o en simples bases de datos, generalmente no posee un formato estándar, idóneo, por lo que se hace casi imposible la aplicación de un algoritmo de minería sobre los datos en su formato inicial.

En esta fase se realiza un pre-procesamiento de los datos para así eliminar los datos incorrectos, no válidos, desconocidos, (5) teniendo en cuenta siempre el algoritmo a utilizar para que los datos recolectados se ajusten al mismo. Además se pueden obtener muestras de los datos con el fin de mejorar los tiempos de respuestas de los algoritmos, así como, su velocidad de procesamiento. También se reduce el número de valores posibles para los atributos de análisis.

1.5.2 Selección de variables

Aún después de haber realizado un pre-procesamiento de los datos, generalmente estos almacenan un gran número de variables o atributos. En esta fase la selección de características disminuye considerablemente el tamaño de los datos sin que se comprometa la calidad del modelo de conocimiento obtenido, eligiendo las variables más importantes e influyentes en el problema.

1.5.3 Algoritmos de Extracción de Conocimiento

La extracción de conocimiento es la fase más importante dentro del proceso de minería de datos, en la cual, mediante la aplicación de una técnica, se obtiene un modelo de conocimiento, que no es más que

una aproximación del comportamiento de los valores de las variables del problema y de la relación existente entre estas (5).

En esta fase se centra el proceso de investigación, pues es en la que se aplican las técnicas para extraer el conocimiento de grandes volúmenes de información que sirvan para la toma de decisiones acertadas por parte de los usuarios finales.

1.5.4 Interpretación y evaluación

Después de obtener un modelo de datos se procede a la fase de interpretación y evaluación de los resultados. Primeramente se procede a la validación de los datos, comprobar que los datos son correctos y que las conclusiones obtenidas son válidas y suficientemente satisfactorias (5).

En el caso que se hayan utilizado diversas técnicas para la obtención de varios modelos de datos, se procede a realizar una comparación entre todos los modelos y se selecciona el que más se ajuste a las características del problema.

En el caso que ninguno de los modelos obtenidos se ajuste a los resultados esperados se deben alterar una de las fases anteriores para la obtención de resultados satisfactorios (5).

1.6 Tareas realizadas en el proceso de minería de datos

En los procesos de minería de datos las tareas pueden considerarse como un tipo de problema a ser resuelto por un determinado algoritmo de minería, es decir, cada tarea posee sus propios requerimientos y los resultados obtenidos en la aplicación de cada una de estas tareas puede ser muy diferente con respecto al resultado de las demás. Estas tareas se pueden clasificar en predictivas o descriptivas, entre las predictivas se encuentran la clasificación y la regresión y entre las descriptivas el agrupamiento, las reglas de asociación, las reglas de asociación secuenciales y las correlaciones (6).

Las tareas dentro de un proceso de minería de datos serán cumplidas o no en dependencia de la naturaleza del problema, o sea, no es vital que todos los proyectos cumplan con todas las tareas, sino, solo aquellas que son necesarias para dar solución al problema en cuestión. Dentro de las tareas más importantes de los procesos de minería de datos se encuentran:

Clasificación: La clasificación es una de las tareas más utilizadas en los procesos de minería de datos. En ella, cada instancia pertenece a una clase, la cual se indica mediante el valor de un atributo que se denomina la clase de la instancia (6). Agrupa todas las herramientas que permiten asignar a un elemento la pertenencia a un determinado grupo o clase (7). Más concretamente, el objetivo del algoritmo es maximizar la razón de precisión de la clasificación de las nuevas instancias, la cual se calcula como el cociente entre las predicciones correctas y el número total de predicciones (correctas e incorrectas). Se usan árboles de decisión, sistemas de reglas o análisis de discriminantes.

Regresión o estimación: Pertenece al grupo de las tareas predictivas y consiste en el aprendizaje de una función real que asigna a cada instancia un valor real de tipo numérico (6). Esta constituye la principal diferencia con respecto a la clasificación, el valor a predecir es numérico. Tiene como objetivo inducir un modelo para poder predecir el valor de la clase dados los valores de los atributos, o sea, minimizar el error entre el valor predicho y el valor real. Se usan árboles de regresión, redes neuronales artificiales, regresión lineal (8).

Agrupamiento o clustering: Es considerada como la tarea descriptiva por excelencia y consiste en la búsqueda de la identificación de tipologías o grupos a partir de los datos en los cuales los elementos tienen gran similitud entre sí y se diferencian de los otros grupos, lo cual permite el tratamiento de forma particular de cada uno de estos grupos (7). Los datos son agrupados según el principio de maximizar la similitud entre los elementos de un grupo minimizando la similitud entre grupos distintos. Este tipo de tareas son utilizadas en una gran variedad de ámbitos para la descripción y clasificación de información.

Correlaciones: Son del tipo de las tareas descriptivas que se utilizan para examinar el grado de similitud de los valores de dos variables numéricas (6). Una fórmula estándar para medir la correlación lineal es a través del coeficiente de correlación r , el cual es un valor real comprendido entre 1 y -1. Si r es igual a 1 entonces las variables están perfectamente correlacionadas, si r es -1 entonces están perfectamente correlacionadas negativamente, en el caso que sea 0 entonces las variables no están correlacionadas. Esto se traduce a que cuando el valor de correlación es positivo las variables tiene un comportamiento similar, o sea, ambas crecen o decrecen al mismo tiempo. En el caso contrario una de las variables crece y la otra decrece.

Reglas de asociación: Es una de las tareas reinas dentro del proceso de minería de datos y pertenece al grupo de las tareas descriptivas. Son muy similares a las correlaciones y tienen como

objetivo identificar relaciones no explícitas entre atributos categóricos (6). Se establecen las posibles relaciones entre acciones o sucesos aparentemente independientes, es decir, mediante esta tarea se puede determinar cuándo un suceso puede inducir a la aparición de otro. Con este fin se identifican relaciones no explícitas entre atributos categóricos o nominales. Estos patrones pueden servir para conocer el comportamiento general del problema que genera los datos a tratar (9). Es importante señalar que estas reglas no implican relaciones de causa efecto ya que puede no existir una causa para que los datos estén asociados.

Reglas de asociación secuenciales: Se consideran un caso especial de las reglas de asociación, pues tienen un concepto similar en el que influye además el factor tiempo, es decir, que permite reconocer el tiempo que transcurre entre el suceso inductor y los sucesos inducidos (6).

1.7 Análisis de soluciones similares existentes

En la actualidad el desarrollo de aplicaciones informáticas para guiar los procesos de enseñanza y aprendizaje constituye uno de los grandes aportes tecnológicos a la esfera de la educación a nivel mundial. La relación existente entre las tecnologías informáticas y la educación surge desde los primeros momentos en la etapa evolutiva del *software*, especialmente como apoyo a los procesos de la educación a distancia, de forma tal que el *software* educativo evolucione a la par de las tecnologías informáticas, posibilitando una integración más adecuada y potente que posibilite un enriquecimiento enorme de los procesos instruccionales.

1.7.1 Minería de datos aplicada a entornos educativos

Durante las últimas décadas, las innovaciones más importantes en los sistemas educativos están relacionadas con la introducción de nuevas tecnologías. Los sistemas educativos basados en la web constituyen una forma de instrucción asistida por computadora prácticamente independiente de una ubicación específica y plataforma de *hardware*. La minería de datos aplicada a estos entornos educativos ha resultado ser una herramienta muy útil para la toma de decisiones, permitiendo el análisis automático y profundo de la información generada por los estudiantes al interactuar con los sistemas educativos (10). Este proceso es un ciclo iterativo en el cual se establecen hipótesis, se prueban y se refinan tal como se muestra en la figura 1.

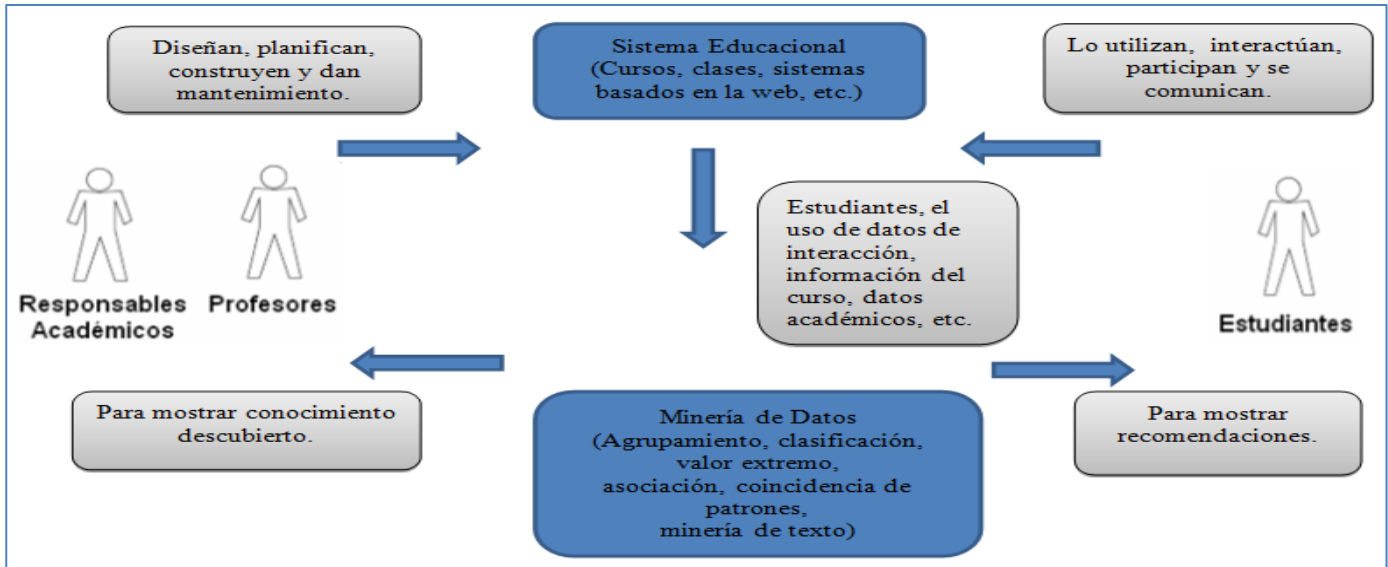


Ilustración 1: Ciclo interactivo del PEA

La aplicación de técnicas de minería de datos en la educación se puede ver desde dos puntos de vista distintos (10) (11):

Orientado hacia los autores: Tiene como objetivo ayudar a los profesores y/o autores de los sistemas de *e-learning* para que puedan mejorar el funcionamiento o rendimiento de estos sistemas a partir de la información de utilización de los alumnos. Sus principales aplicaciones son: obtener una mayor retroalimentación de la enseñanza, conocer más sobre cómo los estudiantes aprenden en la web, evaluar a los estudiantes por sus patrones de navegación, así como, reestructurar los contenidos del sitio web para personalizar los cursos, clasificar a los estudiantes en grupos, entre otros. (10) (11)

Orientado hacia los estudiantes: Con el objetivo de ayudar o realizar recomendaciones a los alumnos durante su interacción con el sistema de *e-learning* para poder mejorar su aprendizaje. Sus principales aplicaciones son: sugerir buenas experiencias de aprendizaje a los estudiantes, adaptación del curso según el progreso del aprendiz, ayudar a los estudiantes dando sugerencias y atajos, recomendar caminos más cortos y personalizados, etc. (10) (11)

Las principales aplicaciones de las técnicas de minería de datos en educación, son sistemas de personalización, sistemas recomendadores, sistemas de modificación, sistemas de detección de irregularidades, sistemas predictivos, entre otras. Dichas aplicaciones se basan en sus capacidades para: el descubrimiento de patrones de navegación regulares e irregulares, realización de clasificaciones de

alumnos y de los contenidos, construcción adaptativa de planes de enseñanza, descubrimiento de relaciones entre actividades, diagnóstico incremental de los estudiantes.

Las tareas 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 (10) (12).

1.7.2 Minería de datos relacionada a Entornos Educativos en el mundo

En el mundo existen disímiles sistemas que hacen uso de técnicas de minería de datos. En un estudio realizado por Sebastián Ventura en el año 2008 (13) sobre la minería de datos en sistemas educativos se destacan el uso de las tareas de clasificación, regresión, *clustering* y asociación. Las tareas de clasificación y agrupamiento o *clustering* aplicadas a sistemas de *e-learning* permiten agrupar a los usuarios por su comportamiento de navegación, agrupar a las páginas por su contenido, tipo o acceso y agrupar los comportamientos de navegación similares. Las reglas de asociación descubren relaciones entre atributos de un conjunto de datos que superan unos determinados umbrales. Su aplicación más típica ha sido en los sistemas de comercio electrónico para informar sobre las preferencias de compra de los clientes. Su aplicación a sistemas de *e-learning* permite descubrir relaciones o asociaciones entre distintas páginas Web visitadas.

En el trabajo “Minería de Datos aplicada a entornos educativos: estudio e implementación de técnicas libre de parámetros” (14) se emplean las tareas de clasificación y reglas de asociación con el objetivo de ofrecer a los profesores la posibilidad de extraer información y modelos de minería de datos, a partir de los datos de actividades de cursos virtuales registrados en plataformas e-learning, sin que estos tengan conocimientos de dichas tareas y las técnicas que comprenden. En este trabajo se hace uso de los algoritmos Apriori y Predictive Apriori presentes en la herramienta Weka para la extracción de reglas de asociación.

Otro ejemplo lo constituye la utilización conjunta de *clustering* con otras técnicas como secuenciación, analizando el comportamiento de navegación de los usuarios para la personalización de sistemas *e-learning* (15). Existen otros trabajos que analizan los ficheros *log* de entornos de aprendizaje web utilizando técnicas de minería de reglas de asociación y filtrado, con el fin de descubrir patrones de navegación útiles y proponer un modelo de navegación (11). El trabajo realizado por Porcel Dapozo (16) en el que se hizo uso de técnicas de minería de datos que les permitió, a través de un clasificador, determinar el rendimiento académico de los alumnos de la carrera de Licenciatura en Sistemas de

Información, para cumplir con esto llevaron a cabo un estudio comparando los diferentes algoritmos clasificadores disponibles en Weka, analizando los resultados que se obtuvieron de la aplicación de cada uno de ellos. Utilizaron la base de datos que alojaba toda la información sistematizada de los alumnos de la Facultad de Ciencias Exactas de la UNNE tomando particularmente en cuenta los siguientes datos:

- La cantidad de exámenes finales rendidos (número de intentos).
- La cantidad de exámenes finales aprobados.
- La cantidad de exámenes finales reprobados.

En base a esos valores, ellos generaron tres categorías que sirvieron para identificar a los alumnos:

- En el año de ingreso no rindieron ninguna materia.
- Los que sí rindieron pero no aprobaron ninguna.
- Los que rindieron y aprobaron por lo menos una materia.

1.7.3 Minería de datos relacionada a Entornos Educativos en Cuba

En Cuba existen pocos trabajos de minería de datos relacionadas a entornos educativos. Uno de ellos es el realizado en la UCI (17), cuyo principal resultado es el uso de técnicas de agrupamiento y asociación guiado por la metodología Crisp-DM, para determinar patrones entre la procedencia del origen social y los resultados académicos en los estudiantes de la UCI.

Existe otro trabajo que fue desarrollado en el Instituto Superior Politécnico José Antonio Echeverría CUJAE (18), con el empleo de técnicas de clasificación para predecir el resultado de los estudiantes en su primer año, en dependencia de sus características de ingreso y la especialidad que cursaban. Otro trabajo realizado en la CUJAE (19) fue desarrollado por el Ing. Raycos Brito Sarasa. Su principal objetivo es descubrir conocimiento en las bases de datos de la universidad y ayudar a la toma de decisiones de la Vicerrectoría Docente a través del uso de técnicas de minería de datos, auxiliándose en la metodología Crisp-DM y la herramienta Weka. En dicho trabajo se emplean tareas de asociación, agrupamiento y clasificación con árboles de decisión, para agrupar a los estudiantes atendiendo a sus características, determinar cuáles son las características de los estudiantes que pueden influir en sus resultados docentes de cada año de estudio y determinar cómo influyen las características de los estudiantes en el promedio de cada año de estudio.

En un trabajo realizado en la CUJAE por varios autores (20), se describen las actividades realizadas siguiendo el modelo propuesto por la metodología Crisp-DM y enfocados en la herramienta de análisis

Weka para la extracción de conocimientos de sus bases de datos. Para la realización del proceso de KDD hicieron uso de las tareas y algoritmos que se presentan a continuación:

Tareas	Algoritmos	Objetivos
Asociación	Apriori (Apriori en Weka)	Obtener reglas que describan las relaciones entre las características de ingreso de los estudiantes.
Agrupamiento	K-medias (Simple Kmeans en Weka)	Obtener grupos de estudiantes con características de ingreso similares.
Clasificación	Árboles de Decisión C4.5 (J48 en Weka)	Obtener, luego de la selección de atributos, como estos influyen en el promedio de un año de los estudiantes.

Tabla 1: Ejemplo de minería de datos en los entornos educativos

Resultados del estudio

Una vez concluido el estudio de las soluciones similares existentes tanto en el ámbito nacional como internacional, se concluye que no se encontró ningún trabajo que se adapte a las características del sistema a implementar. Los ejemplos analizados mostraron a Weka, como una de las herramientas de apoyo a los procesos de extracción de conocimientos más utilizadas y a Crisp-DM como metodología para guiar las distintas fases del proceso, presente en la mayoría de los trabajos analizados. Se identificaron las tareas de agrupamiento, clasificación y reglas de asociación como las más comunes en los sistemas educativos.

1.8 Metodología para el desarrollo de proyectos de minería de datos

Con el propósito de organizar el trabajo en el desarrollo de proyectos de minería de datos, han sido desarrolladas diversas metodologías y estrategias de trabajo. El empleo de una metodología bien estructurada y organizada presenta las siguientes ventajas (21):

- Facilita la planificación y dirección del proyecto.
- Permite realizar un mejor seguimiento del proyecto.
- Facilita el desarrollo de nuevos proyectos de minería de datos con características similares.

Crisp-DM

Es una metodología para el desarrollo de proyectos de minería de datos desarrollada en 1996, entre sus principales características de destacan:

- Concibe el proyecto de minería de datos de forma global y estrechamente relacionado al negocio en cuestión.
- Fue diseñada de forma neutra a la herramienta que se utilice para el desarrollo del proyecto.
- Es de distribución libre y se encuentra en constante perfeccionamiento por parte de la comunidad internacional.
- Presenta una precisa y sólida distribución de tareas de carácter general con sus resultados, así como una guía para su desarrollo.
- Muchas de las metodologías que se pueden encontrar en la actualidad se basan en este estándar.
- Es la que cuenta con mayor aceptación por parte de los desarrolladores de procesos de extracción de conocimientos a partir de datos.

La metodología Crisp-DM estructura el ciclo de vida de un proyecto de *Data Mining* en seis fases, que interactúan entre ellas de forma iterativa (no lineal) durante el desarrollo del proyecto. A nivel más general, el proceso está organizado en seis fases, estando cada fase a su vez estructurada en varias tareas generales de segundo nivel. Las tareas generales se proyectan a tareas específicas, donde se describen las acciones que deben ser desarrolladas para situaciones específicas (**Ver anexo 1**)

1. Comprensión del negocio

Fase inicial enfocada a entender los objetivos y requerimientos desde una perspectiva de negocio; para luego definirlos en términos de un problema de minería de datos y diseñar un plan para satisfacerlos.

2. Comprensión de los datos

En esta fase se hace una recolección y exploración inicial de los datos para familiarizarse con ellos e identificar problemas de calidad. Además, se trata de descubrir o estimar las relaciones más evidentes para formular las primeras hipótesis sobre información oculta en ellos.

3. Preparación de los datos

Esta fase cubre todas las actividades necesarias para construir la colección de datos que finalmente será minada a partir del grupo inicial. Incluye la colección, exploración, limpieza, transformación y construcción de datos.

4. Modelado

Durante esta fase se aplican varias técnicas de modelado. Comúnmente existen diversas técnicas para resolver un problema de minería de datos del mismo tipo. Incluye la evaluación desde el punto de vista de precisión de los modelos.

5. Evaluación

Al llegar a esta fase se tendrán los modelos de mayor calidad desde la perspectiva de la precisión. Se impone una evaluación de los modelos y de los pasos que se siguieron para su construcción, a fin de determinar si responden apropiadamente a los objetivos de negocio que se determinaron en la primera fase.

6. Despliegue

En dependencia de los requerimientos y objetivos, la fase de despliegue puede ser tan simple como generar un reporte, o tan compleja como emprender un proceso de KDD de mayor envergadura. Consiste en explotar las potencialidades de los modelos, integrarlos en los procesos de tomas de decisiones de las organizaciones, difundir informes sobre el conocimiento extraído, entre otros.

1.9 Metodología para el desarrollo de *software*

Los procesos de desarrollo de *software* son guiados a nivel internacional por un conjunto de métodos, procedimientos y herramientas denominadas metodologías de desarrollo. Surgen con el fin de organizar a la par de agilizar y optimizar los procesos para así desarrollar un producto de calidad y en un tiempo óptimo garantizando la satisfacción de los clientes. Define las etapas para el desarrollo de un *software*, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto (22). Se selecciona como metodología para el desarrollo a RUP pues es la que está definida por el equipo de trabajo del proyecto El Navegante para el desarrollo de la plataforma. Un cambio de metodología implicaría grandes transformaciones, desde el punto de vista arquitectónico en el desarrollo de la colección. El sistema a

desarrollar forma parte de la colección por lo que se debe mantener la metodología de desarrollo, con el objetivo de que los artefactos generados puedan ser incorporados al documento de arquitectura de la colección.

Proceso Unificado de Desarrollo (RUP)

Es un marco de proceso integral que ofrece probadas prácticas de la industria para el *software* y la entrega e implementación de sistemas y para la gestión eficaz de los proyectos (23). Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo.

Características fundamentales:

Las principales consideraciones a tener en cuenta para la selección de esta metodología son:

- Apto para el desarrollo de proyectos de gran tamaño y elevada complejidad.
- Es considerado como una metodología de desarrollo tradicional o pesada debido a que hace un mayor énfasis en la documentación a diferencia de las metodologías ágiles.
- El cliente debe interactuar con el equipo de desarrollo pero no tiene que formar parte de este.
- Asociado al desarrollo se genera una gran cantidad de documentación que pueden ayudar a entender y mantener el sistema.

1.10 Tecnologías y herramientas a utilizar

En este epígrafe se mostrarán las principales características de las tecnologías que se usarán durante el desarrollo del sistema teniendo en cuenta los requerimientos de estas para el desarrollo de la aplicación. También se aludirá a las principales herramientas a utilizar basadas en sus características principales.

1.10.1 Herramientas usadas en el proceso de minería de datos

Las herramientas de minería de datos constituyen para los mineros unos potentes instrumentos para facilitar los procesos de minería y a su vez agilizan y facilitan estos procesos.

Weka 3.6.8

Weka (Waikato Environment for Knowledge Analysis) es un conjunto de librerías Java para la extracción de conocimientos desde bases de datos. Es un *software* desarrollado bajo la licencia GPL, código abierto e incluye interfaz gráfica compuesta por diversos entornos, desarrollada por un grupo de investigadores de la Universidad de Waikato en Nueva Zelanda en 1993. Se puede acceder a todas las funcionalidades de minería de datos a través de una interfaz de líneas de comandos (CLI), de manera que las aplicaciones puedan sacar el máximo partido de las funciones disponibles. Constituye un entorno de experimentación de análisis de información, formado por una serie de paquetes de código abierto con diferentes técnicas de pre-procesado, clasificación, asociación, y visualización de datos (24). Se encuentra pública en los repositorios de los sistemas operativos Unix y por tanto se tiene acceso a su código fuente.

RapidMiner 5.2

RapidMiner (antes YALE) es una herramienta para el análisis y minería de datos la cual permite a través de un entorno gráfico el desarrollo de procesos de análisis de datos mediante encadenamiento de operadores y se usa en investigación y en aplicaciones empresariales. El mismo es desarrollado en Java, multiplataforma y está disponible como una aplicación independiente para el análisis de datos y como un motor de minería de datos para la integración con otros productos. Miles de aplicaciones de RapidMiner son utilizadas en más de 40 países. Se encuentra disponible de forma gratuita y de código abierto. Funciona en las principales plataformas y sistemas operativos y presenta un proceso muy intuitivo y con buen diseño. Puede ser accesible a través de una interfaz gráfica de usuario, modo servidor (líneas de comando) o accedido por medio de las API de Java. Posee un mecanismo de extensión simple, brinda soluciones completas. Posee un diseño de proceso gráfico para tareas estándar y un lenguaje de *scripts* para operaciones arbitrarias. Permite utilizar los algoritmos incluidos en Weka y el acceso a fuentes de datos, como Excel, Access, Oracle, IBM DB2, Microsoft SQL, Sybase, Ingres, MySQL, Postgres, SPSS, dBase, archivos de texto y más (25).

Orange 2.7

Es una herramienta para el análisis predictivo, la cual consta de una serie de componentes que implementan algoritmos de minería de datos, así como operaciones de preprocesamiento y representación gráfica de datos. Dichos componentes pueden ser manipulados desde programas

desarrollados en Python o a través de un entorno gráfico bastante cómodo. Los algoritmos están programados en C++ y están interconectados con Python (26).

KNIME 2.7

KNIME (o Konstanz Information Miner) es una plataforma de minería de datos que permite el desarrollo de modelos en un entorno visual. Está construido bajo la plataforma Eclipse en lenguaje Java (27). La versión base de KNIME ya incorpora más de 100 nodos de entrada y salida de datos, de pre-proceso y limpieza, de modelado, de análisis y minería de datos, así como diversas vistas interactivas, tales como diagramas de dispersión, coordenadas paralelas y otros.

Integra todos los módulos de análisis del conocido entorno de minería de datos Weka y otros *plugins*¹ adicionales que permiten ejecutar *scripts*, que ofrece acceso a una vasta biblioteca de rutinas estadísticas (28).

KNIME puede ser descargado en el escritorio y utilizado de forma gratuita. Los productos de KNIME incluyen funcionalidades adicionales, tales como repositorios compartidos, la autenticación, la ejecución remota, programación y una interfaz de usuario de la web, así como soporte de clase mundial.

En la presente investigación, después de un análisis de las principales herramientas se selecciona a Weka como herramienta de apoyo en el KDD. Esta selección se fundamenta debido a que es una herramienta reconocida a nivel mundial y se caracteriza por ser estable y contar con una gran comunidad de desarrollo, así como ser una de las más utilizadas a nivel internacional (29) (30). Es una herramienta que se encuentra muy bien documentada por parte de sus desarrolladores y los mismos brindan soporte a las investigaciones a través de foros y correos electrónicos. Brinda acceso a sus algoritmos a través de comandos que pueden ser ejecutados desde el servidor web, garantizando así una fácil integración con la colección. Otra de las características distintivas es que se encuentra pública en los repositorios de los sistemas Unix lo cual garantiza la instalación de la herramienta en el despliegue del producto sin necesidad de ser incluida en el instalador del mismo, garantizando así que siempre se cuente con la última versión de la herramienta.

¹ Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

1.10.2 Frameworks de desarrollo

Un *framework* de desarrollo no es más que una estructura *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta (31).

Características fundamentales:

- Acelerar el proceso de desarrollo.
- Reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.
- Simplifican el desarrollo de las aplicaciones mediante la automatización de las tareas comunes.
- Proporcionan estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener.

Framework del lado del cliente:

Jquery 1.5

Es una biblioteca de JavaScript que facilita el manejo de eventos, animaciones así como el tratamiento de los diferentes elementos HTML presentes en las páginas web. Permite además la creación de interfaces de usuario, efectos dinámicos y aplicaciones que hacen uso de AJAX.

Características fundamentales:

- Es una biblioteca basada en *software* libre y de distribución gratuita.
- Gran facilidad en la manipulación de las animaciones, eventos y en el tratamiento de las reglas CSS.
- Es fácilmente interpretado por los navegadores más importantes ahorrando problemas de compatibilidad.
- Posee una comunidad de desarrollo, muchos *plugins* y buen soporte.

Framework del lado del servidor:

Symfony 1.4.3

Es un completo *framework* diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web (32). Se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Características fundamentales (32):

- Separa la lógica de negocio, la lógica del 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.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

1.10.3 Lenguaje de modelado

UML 1.4: *Unified Modeling Language* o Lenguaje Unificado de Modelado es un lenguaje gráfico destinado al modelado de sistemas y procesos. Está basado en la orientación a objetos que condujo, en primer lugar, a la creación de lenguajes de programación como Java y C++ (33).

UML brinda una aproximación gráfica del sistema que se analiza posibilitando una mejor comprensión de sus componentes y facilitando un mejor entendimiento de todos los procesos del sistema (33).

Se puede destacar que UML se puede aplicar en el desarrollo de *software* facilitando gran diversidad de formas para dar soporte a metodologías de desarrollo de *software* como RUP.

1.10.4 Herramienta Case

Visual Paradigm 8.0: Es una herramienta Case que provee el modelado de procesos de negocios, creada para ayudar en el proceso de desarrollo de *software*. Soporta los principales estándares de la industria, tales como UML, SysML, BPMN y XMI (34).

Esta herramienta tiene unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que sigue el estándar de UML. Además posee soporte para aplicaciones web, así

como, se encuentra en varios idiomas y es capaz de generar código Java y exportación como HTML. Posee compatibilidad entre las ediciones del producto y es relativamente fácil de instalar y actualizar.

1.10.5 Lenguajes de programación web:

En el desarrollo de aplicaciones web se identifican dos tipos o ramas fundamentales de lenguajes de programación:

1. Lenguajes del lado del servidor:

Los programas son ejecutados por el servidor y lo que se envía al cliente es la respuesta o resultado de dicha ejecución en un formato comprensible para él (35).

2. Lenguajes del lado del cliente:

Son los lenguajes ejecutados en el cliente web, es decir, se ejecutan en el navegador web del usuario.

1.10.5.1 Lenguajes del lado del cliente:

HTML 5:

HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados (36).

Ventajas fundamentales (37):

1. La vida de los documentos. Las páginas que se elaboren de acuerdo con los últimos estándares de la web, tendrán código válido por muchos años.
2. Compatibilidad mayor con los navegadores.
3. Mejor legibilidad del código, sobre todo desde el punto de vista de las actualizaciones de las páginas.
4. Una mejor indexación. El contenido, separado de la presentación, puede ser identificado más rápido y eficazmente por los motores de búsqueda.

CSS 3:

Cascading Style Sheets (Hojas de Estilo en Cascada) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS

es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (38).

Ventajas fundamentales (37):

1. Las hojas de estilos gestionan todo lo concerniente a la apariencia, dejando al HTML la función de estructura y de codificar la información bruta.
2. Contribuyen a despertar la creatividad de los usuarios aportando nuevas posibilidades de presentación ausentes en el código HTML.
3. Simplificación del código: permiten aligerar considerablemente el código fuente volviéndolo más legible y accesible.
4. Permiten garantizar un mantenimiento al sitio más fácil y con mayor rapidez.

JavaScript 1.5:

JavaScript 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 (39).

Ventajas fundamentales (40):

1. Brinda las páginas HTML una apariencia más agradable y ofrece varias funcionalidades suplementarias.
2. Contribuye a mejorar la facilidad de uso del sitio e incrementa además la fidelidad del usuario.
3. Contribuye a un mejoramiento de la profesionalidad de un sitio ya que permite a los usuarios desde sus páginas HTML acceder a otras funcionalidades y servicios.
4. Tiene la propiedad de ser dinámico y responde a los eventos en tiempo real.
5. Es un lenguaje de programación relativamente sencillo y no son necesarios grandes conocimientos de programación para usarlo.

1.10.5.2 Lenguajes del lado del servidor:

PHP 5.3:

PHP, acrónimo de *Hypertext Pre-processor*, es un lenguaje de programación con una sintaxis similar a los lenguajes Perl y C que se interpreta por un servidor web y genera código HTML dinámicamente, es

decir, permite crear un programa que se pueda ejecutar en el servidor desde un programa visualizador de páginas web y dar respuestas en función de los datos que se introduzcan por el usuario (41).

Ventajas fundamentales (41):

1. Se ejecuta a través de una interfaz que le resulta familiar al usuario: el cliente web.
2. El código se encuentra protegido tanto de la manipulación de los usuarios como de la presencia de virus.
3. Se caracteriza por su facilidad de aprendizaje en breve tiempo.
4. Es multiplataforma y no requiere de recursos desmesurados para que funcione.

1.10.6 Servidor web

Un servidor web es un simple servidor de archivos. Los clientes se dirigen hacia él mediante el protocolo HTTP o HTTPS para obtener un recurso. Cuando el servidor web recibe una petición HTTP, extrae simplemente de la petición el nombre del recurso solicitado, lo busca en el disco y lo devuelve dentro de una respuesta HTTP para trasmitirlo al cliente (42).

Apache 2.0: Apache es un servidor de páginas web de código abierto multiplataforma y modular, cuyo nombre se debe a que en sus orígenes era sólo en un conjunto de parches a aplicar en un servidor web. En inglés era “*a patchy server*”, es decir, un servidor parcheado (43).

Características fundamentales:

- Basado en la tecnología de *Software* libre y a la vez uno de los más usados a nivel mundial.
- Debido a su flexible sistema modular, permite cargar y descargar módulos sin necesidad de tocar el *kernel*.
- Los módulos se cargan en memoria cuando los necesita y se descargan automáticamente cuando dejan de utilizarse.
- Soporta la autenticación HTTP.
- Posibilita el chequeo avanzado de sintaxis de la URL, incluyendo caracteres falsos que son fuentes frecuentes de ataques como *SQL Injection*, *Command Injection* y *XSS*.

1.10.7 Sistemas de gestión de bases de datos

Un sistema gestor de bases de datos (SGBD) es un *software* o conjunto de programas que permiten crear y mantener una base de datos. El SGBD actúa como interfaz entre los programas de aplicación (usuarios) y el sistema operativo. El objetivo fundamental de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos (44).

PostgreSQL 8.4

PostgreSQL es un potente sistema de gestión de bases de datos, multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y está liberado bajo la licencia *Berkeley Software Distribution* (BSD) (45). Tiene interfaces de programación con diferentes lenguajes como PHP, C, C++, Java, .NET, Perl, Python, entre otros. Además brinda una amplia documentación lo que hace más fácil trabajar con el mismo.

Características fundamentales (45):

- Presenta claves ajenas también denominadas llaves ajenas o claves foráneas en inglés (*foreign keys*).
- Tiene integridad transaccional
- Posee herencia de tablas
- Cuenta con tipos de datos y operaciones geométricas
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores. Soporta el uso de índices, reglas y vistas.

1.10.8 Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado o *Integrated Development Environment* (IDE), es un programa que proporciona herramientas muy útiles para los programadores, tales como depuradores de código, ayuda en línea y un constructor de interfaz gráfica para facilitar el desarrollo de otras aplicaciones.

NetBeans 7.2

Es un IDE que permite el desarrollo de forma rápida y fácil de aplicaciones de escritorio Java, móviles y aplicaciones web, mientras que también proporciona una gran herramienta para PHP. Para el desarrollo

de la solución se selecciona como IDE de desarrollo NetBeans en su versión 7.2.

Características fundamentales (46):

- Tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS, Ajax, Ruby y Ruby onRails, Groovy y Grails, y C / C + +.
- Incluye resaltado de sintaxis, es fácil y eficiente en la gestión de proyectos.
- A partir de su versión 6.8 tienen soporte nativo a los *frameworks* Symfony y JQuery lo cual es muy importante para el desarrollo del proyecto.
- Debido a que el editor es extensible, se le pueden agregar otros *plugin* para el soporte de otros lenguajes.
- Ofrece plantillas de código, información sobre herramientas, parámetros, consejos y soluciones rápidas (aplicación de métodos abstractos), autocompletado de código inteligente, formateo de código y marcado de los sucesos y los puntos de salida.
- Se integra con los controladores de versión Subversion, Git, y otros.
- Es multiplataforma.
- Es gratuito, de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo.

Análisis de tecnologías y herramientas

El set de herramientas y tecnologías seleccionadas para el desarrollo de la solución, fue escogido por políticas del proyecto, para garantizar que el desarrollo de la colección sea realizado utilizando las mismas herramientas en todos sus componentes. La solución formará parte del módulo Resultados, por lo que se trata de continuar con la utilización de las mismas herramientas para garantizar una uniformidad dentro del módulo. El equipo de desarrollo ha venido trabajando en la implementación de la colección por lo que ya posee experiencia en el trabajo con estas herramientas y el cambio de las mismas no brindaría grandes beneficios al desarrollo de la solución. Fueron actualizadas las herramientas que no constituían cambios significativos en el documento de arquitectura y no afecten el correcto funcionamiento del sistema. Para garantizar un desarrollo exitoso de la aplicación se utilizará como sistema de control de versiones subversion apoyado en el cliente RapidSVN.

1.11 Conclusiones del capítulo

El estudio bibliográfico asociado al objeto de estudio evidenció la importancia del uso de la minería de datos en el análisis de información, así como, el impacto significativo en los entornos educativos y la

Capítulo 1

necesidad de que el módulo Resultados de la colección El Navegante permita la realización de un KDD. Luego de realizar un estudio de las soluciones similares, se identificaron las tareas agrupamiento y extracción de reglas de asociación, apoyado en la herramienta Weka y guiado por la metodología Crisp-DM para el desarrollo de la solución. Este desarrollo estará sustentado en el uso de tecnologías libres, para ello se hizo un estudio de las principales tendencias en lenguajes, metodologías y demás herramientas para el desarrollo de aplicaciones web. De ellos fueron seleccionados los que estaban en concordancia con la arquitectura previamente definida en el proyecto al que pertenece la solución.

CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA

2.1 Introducción

Con el objetivo de realizar el diseño de un sistema informático eficiente, es necesario dejar claro todos los conceptos y sus relaciones utilizando un lenguaje perceptible por todos los involucrados. Para lograr una comprensión del sistema en el presente capítulo se describe el Modelo del Dominio y sus principales conceptos. Se exponen además, los requisitos funcionales y no funcionales requeridos en el desarrollo de la aplicación, así como, los actores que interactúan con el sistema, el Diagrama de Caso de Uso del sistema y la descripción de cada uno de ellos. Se describe el desarrollo del proceso de minado siguiendo el flujo propuesto por la metodología Crisp-DM.

2.2 Modelo de dominio

El modelo de dominio define un modelo de clases común para todos los involucrados en el sistema, tanto los analistas como los clientes (47). Es una aproximación clara a la realidad objetiva de cómo funciona el sistema representando elementos del mundo real, no componentes de *software*, y así, describir los objetos más importantes del dominio del sistema.

2.2.1 Análisis de los conceptos del dominio

El Navegante: Colección de *software* formada por 10 productos basados en hiperentornos de aprendizaje destinada a los estudiantes de secundaria básica.

Producto: Cada uno de los *software* de la colección en los que se combinan los distintos tipos de medias. Se trata un tema específico de una o varias áreas de conocimientos relacionados con los temas del plan de estudio de los estudiantes de secundaria básica.

Módulo: Agrupa los elementos fundamentales del producto constituyendo su principal estructura.

Contenido: Módulo encargado de agrupar los contenidos más importantes desde el punto de vista del aprendizaje de los estudiantes. Recopila todo el sistema de conocimientos relacionados con el tema del producto.

Ejercicios: Módulo encargado de comprobar el proceso de aprendizaje de los estudiantes a través de disímiles tipologías de ejercicios y cuestionarios.

Mediateca: Módulo encargado de gestionar los recursos multimedia (Imágenes, videos, textos) con los que el estudiante interactúa en el sistema.

Juegos: Módulo que constituye el componente lúdico del hiperentorno con la finalidad de que los estudiantes pongan en práctica los elementos de los conocimientos adquiridos.

Maestro: Módulo que permite a los profesores administrar la información sobre sus estudiantes, grupos académicos, noticias, escuelas y demás maestros en la colección.

Resultados: Módulo que se encarga de controlar los resultados obtenidos por los estudiantes durante su interacción con la aplicación y detallar todas las actividades realizadas.

Opción: Conjunto de reportes que brinda el módulo Resultados de las actividades realizadas por los estudiantes durante su interacción con la aplicación.

Trayectoria del estudiante: Brinda el reporte relacionado con información de la interacción del estudiante en una sesión de trabajo con la colección.

Análisis integral: Muestra los resultados de la interacción de los estudiantes con los contenidos, teniendo en cuenta sus resultados en función de las fechas en que los estudiantes interactuaron con los ejercicios.

Análisis de contenido: Brinda el reporte de las calificaciones que el estudiante va obteniendo de forma paulatina, con los cuestionarios interactivos del módulo ejercicios, en función de los contenidos.

Historial del estudiante: Muestra los resultados de la interacción con los ejercicios de todos los productos de la colección en función de las asignaturas.

Procesamiento inteligente: Genera un reporte basado en los procesos de extracción de conocimientos que muestre los elementos más significativos de los datos almacenados en el módulo Resultados.

2.2.2 Diagrama del modelo de dominio

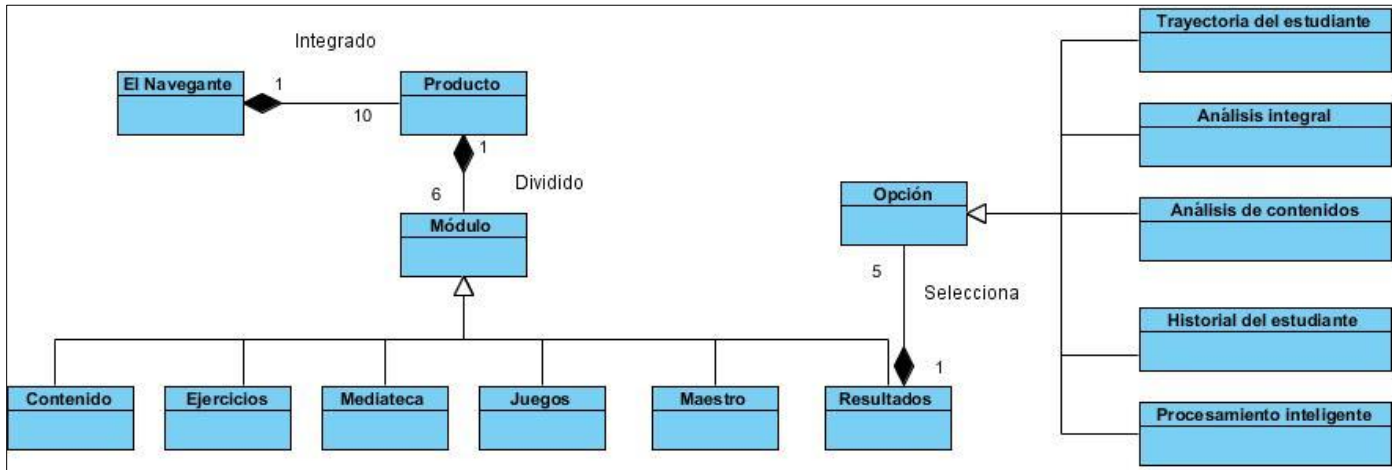


Ilustración 2: Modelo de dominio

2.3 Especificación de requisitos

Un requisito (o requerimiento) es una cualidad o capacidad especificada por el cliente y que debe cumplir el sistema. Estos se pueden clasificar en:

Requerimientos funcionales:

Definen las funcionalidades que el sistema tiene que realizar, para lo cual se identifican las funcionalidades requeridas.

Requerimientos no funcionales:

Son propiedades o cualidades que el producto debe tener. Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad, entre otras.

2.3.1 Requisitos funcionales

RF 1: Mostrar pantalla principal del procesamiento inteligente.

RF 2: Mostrar los posibles algoritmos a utilizar.

RF 2.1 Mostrar opción del algoritmo Kmeans.

RF 2.2 Mostrar opción del algoritmo DBScan.

RF 2.3 Mostrar opción del algoritmo Apriori.

RF 2.4 Mostrar opción del algoritmo PredictiveApriori.

RF 3: Aplicar algoritmos de minería de datos.

RF 3.1 Aplicar algoritmo Kmeans.

RF 3.2 Aplicar algoritmo DBScan.

RF 3.3 Aplicar algoritmo Apriori.

RF 3.4 Aplicar algoritmo PredictiveApriori.

RF 4: Mostrar resultados.

RF 4.1 Mostrar resultados de Kmeans.

RF 4.1.1 Mostrar los grupos en forma de gráfico.

RF 4.1.2 Mostrar características de un grupo seleccionado.

RF 4.2 Mostrar resultados de DBScan.

RF 4.2.1 Mostrar los grupos en forma de gráfico.

RF 4.2.2 Mostrar características de un grupo seleccionado.

RF 4.3 Mostrar resultados de Apriori.

RF 4.4 Mostrar resultados de PredictiveApriori.

RF 5: Generar reporte de resultados.

RF5.1: Genera un reporte en formato pdf de los resultados del procesamiento con los algoritmos Apriori y PredictiveApriori.

2.3.2 Requisitos no funcionales

RNF1 Usabilidad

El análisis se podrá realizar desde cualquier máquina cliente, haciendo las solicitudes necesarias al servidor. La funcionalidad podrá ser accedida por los profesores, sin la necesidad de que estos tengan profundos conocimientos sobre el KDD. Los resultados son mostrados de forma comprensible para el usuario final, además de contar con comentarios sugerentes al interactuar con el sistema.

RNF2 Software

La aplicación dependerá de un navegador Web (Mozilla Firefox versión 3.6 o superior). El servidor web debe contar con la instalación de Weka (versión 3.6.6 o cualquier otra versión compatible con este) y la máquina virtual de Java (JVM) en su versión 1.4 o superior.

RNF3 Hardware

Servidor de aplicaciones y de base de datos:

- Procesador Pentium 2.60 GHz.
- 512 MB de RAM (o mayor).
- 15 GB de espacio en disco duro.
- Dispositivo de red de al menos 10 Mbits.

Estaciones clientes:

- Procesador Pentium 2.33 GHz (recomendado 2.60 MHz o mayor).
- 128 MB de RAM (recomendado 256 MB de RAM o mayor).
- 1 GB de espacio en disco duro.
- Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- Dispositivo de red de al menos 10 Mbits.

RNF4 Restricciones del diseño y la implementación.

En la implementación de la aplicación se deben tener en cuenta los siguientes aspectos:

- Lenguaje de programación del lado del servidor:
 - PHP en su versión 5.3.
- Lenguaje de programación del lado del cliente:
 - Lenguaje JavaScript en su versión 1.5.
 - Lenguaje HTML en su versión 5.
 - Lenguaje CSS en su versión 3.
- Entorno integrado de desarrollo (IDE) NetBeans 7.0.
- *Framework* del lado del servidor Symphony 1.4.3.
- *Framework* del lado del cliente JQuery 1.5.

Los aspectos anteriormente mencionados están en concordancia con la necesidad del uso de tecnologías libres, que permitan el desarrollo de aplicaciones web robustas que puedan cumplir con requerimientos especificados.

RNF5 Requisitos de Licencia

En el desarrollo del *software* no podrán utilizarse componentes que no hayan sido liberados bajo licencias de *software* libre. La aplicación debe ser liberada bajo alguna de las licencias de *software* libre.

RNF6 Requisitos de soporte

Una vez desplegado el producto se desarrollarán procesos de capacitación a los usuarios finales para garantizar mejores resultados en el trabajo con la herramienta.

RNF7 Seguridad

Para contribuir con la seguridad de la información que se maneja en el sistema solo podrán acceder al procesamiento los profesores.

RNF8 Portabilidad

El sistema será multiplataforma, razón por la cual podrá ser utilizado en sistemas operativos Windows y Linux.

2.4 Modelo de casos de uso del sistema

Un caso de uso (CU) es una manera de utilizar el sistema o interactuar con él. Proporcionan una definición de las necesidades a cubrir por un proyecto desde el punto de vista del usuario y a su vez una técnica utilizada para ayudar al cliente a determinar sus necesidades y requisitos (48).

2.4.1 Actores del sistema

Un actor es un conjunto coherente de roles que desempeñan los usuarios de los casos de uso cuando interactúan con el sistema (49).

Actor	Descripción
Usuario	Representa a una persona que va interactuar con el sistema, solo tendrán acceso los profesores.

Tabla 2: Actores del sistema

2.4.2 Diagrama de casos de uso

Los diagramas de casos de uso sirven para mostrar las funciones de un sistema de *software* desde el punto de vista de sus interacciones con el exterior y sin entrar ni en la descripción detallada ni en la implementación de estas funciones (50).

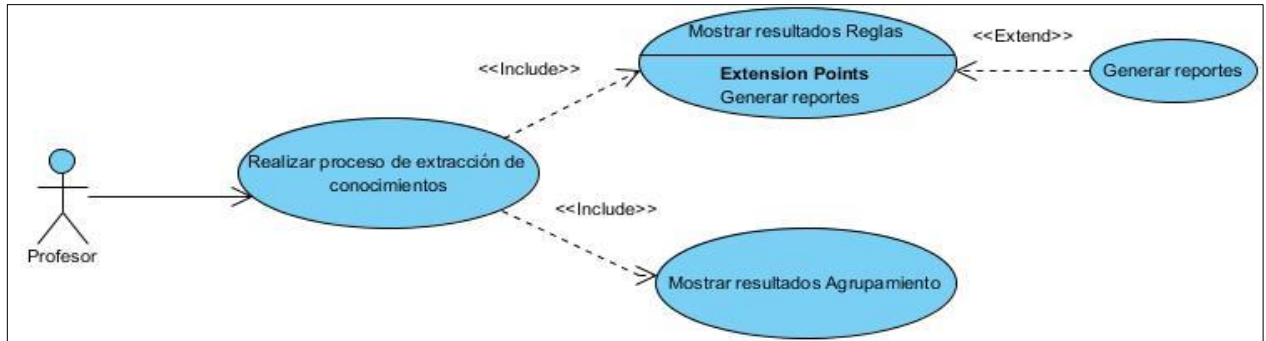


Ilustración 3: Diagrama de casos de uso del sistema

2.5 Descripción de los casos de uso del sistema

A continuación se muestra un resumen de las descripciones de los casos de uso identificados en el modelado del sistema, para ver las descripciones de forma ampliada (Ver anexo 2).

CU-1	Realizar proceso de extracción de conocimiento
Actor	Profesor
Propósito	Mostrar al usuario la opción de realizar un proceso de extracción de conocimiento, permitiéndole configurar los algoritmos a utilizar en el proceso.
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción Resultados. El sistema le facilita al usuario la realización de un análisis de los registros de la interacción de los estudiantes con la aplicación utilizando minería de datos, permitiendo seleccionar los distintos tipos de algoritmos en función de los objetivos. Permite además la especificación de parámetros de los distintos algoritmos por parte del usuario.

Tabla 3: CU Realizar proceso de extracción de conocimiento

CU-2 Mostrar resultados Agrupamiento	
Actor	Profesor
Propósito	Mostrar los resultados del proceso de agrupamiento al usuario final.
Resumen	El caso de uso inicia cuando el usuario aplica un algoritmo de agrupamiento. El sistema realiza una interpretación de los resultados obtenidos y los muestra garantizando una fácil comprensión.

Tabla 4: CU Mostrar resultados Agrupamiento

CU-3 Mostrar resultados Reglas	
Actor	Profesor
Propósito	Mostrar los resultados del proceso de extracción de reglas de asociación al usuario final.
Resumen	El caso de uso inicia cuando el usuario aplica un algoritmo de reglas de asociación. El sistema realiza una interpretación de los resultados obtenidos y los muestra garantizando una fácil comprensión.

Tabla 5: CU Mostrar resultados Reglas

CU-4 Generar reporte	
Actor	Profesor
Propósito	Generar un reporte con los resultados del proceso de extracción de regla de asociación al usuario final.
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción generar reporte. El sistema genera un reporte en formato PDF con las reglas encontradas y lo muestra al usuario.

Tabla 6: CU Generar reportes

2.6 Fundamentación de la propuesta de solución

En este epígrafe se realiza una descripción de las tres primeras fases de la metodología Crisp-DM (Análisis del problema, Análisis de los datos y Preparación de los datos) así como las principales actividades con las que se le dio cumplimiento a cada una de estas fases como parte del proceso de extracción de conocimientos.

2.6.1 Comprensión del negocio

La primera fase se centra en entender los objetivos y requerimientos del proyecto desde una perspectiva de negocio, plasmando todo esto en una definición del problema de minería de datos (6).

- **Establecimiento de los objetivos de negocio**

Se pretende incluir entre los reportes generados por el módulo Resultados de la colección El Navegante, uno que sea capaz de generar información importante para los profesores tomando como punto de partida los datos almacenados en dicho módulo y como base la aplicación de algoritmos de minería de datos. El reporte generado debe ser expuesto de forma clara y de fácil comprensión para cualquier tipo de usuario.

- **Evaluación de la situación**

Inventario de recursos y requerimientos: Para el desarrollo del sistema se selecciona la herramienta Weka como base para la aplicación de los algoritmos de minería de datos. Dicha herramienta está desarrollada en Java por lo que se hace necesaria la instalación de la máquina virtual. Ambas tecnologías son de libre uso y distribución por lo que no existen restricciones en cuanto a las propiedades de las herramientas.

- **Establecimiento de los objetivos de la minería de datos**

Se pretende con la aplicación de técnicas y algoritmos de minería de datos la obtención de reglas y patrones que puedan ser utilizados por los profesores para el perfeccionamiento del proceso de aprendizaje.

2.6.2 Comprensión de los datos

Esta fase se centra en recopilación y familiarización con los datos, identificar los problemas de calidad y ver las primeras potencialidades o subconjuntos de datos que puedan ser interesantes analizar (6).

- **Recopilación y descripción de los datos**

Los datos que serán analizados se encuentran almacenados en el módulo Resultados de la colección, que son muestra de la interacción de los estudiantes con el sistema. Se analiza la información de las diferentes tablas de la base de datos, seleccionando la que más aspectos significativos posea y que sean más interesantes para analizar.

La colección El Navegante cuenta con cinco módulos que pueden ser accedidos por los estudiantes, por lo que resulta interesante analizar la interacción de estos con cada uno de los módulos, con el fin de encontrar patrones de navegación y otras informaciones importantes a tener en cuenta por los profesores a la hora de estructurar los elementos que integran la colección. A su vez, el módulo Contenido es el eje central del hiperentorno pues tiene la aspiración de cubrir el currículo de las asignaturas de la colección. Es de gran interés para los profesores controlar el acceso de los estudiantes a este módulo ya que contiene la información necesaria para adquirir los conocimientos para poder interactuar con los restantes módulos, es decir, controlar cuáles son los temas más visitados y así estructurar los conocimientos con el fin de perfeccionar el proceso de aprendizaje de los estudiantes.

2.6.3 Preparación de los datos

Esta fase cubre todas las actividades necesarias para construir la colección de datos que finalmente será minada a partir del grupo inicial. La herramienta Weka posee varias formas para definir el origen de los datos; especificándole la dirección de un fichero con extensión .ARFF localizado en el propio ordenador, o especificándole la dirección de este fichero a través de una dirección URL, o finalmente definiéndole la dirección de una base de datos existente y detallando los parámetros para la conexión.

Fichero ARFF

Los datos que se definen para el trabajo con la herramienta Weka se incluirán en un archivo de extensión .ARFF que se construirá siguiendo la estructura definida por el programa Weka y con los datos más importantes para la aplicación de los algoritmos de minería de datos.

Estructura del fichero ARFF

El fichero ARFF está compuesto fundamentalmente por tres partes: cabecera, declaración de atributos y sección de los datos.

En la cabecera se define el nombre de la relación siguiendo como estructura:

```
@relation <nombre_relación>.
```

En la sección de declaración de los atributos se declaran todos los atributos que contendrá el archivo, unido al tipo de datos del atributo. Se definirán los atributos de la siguiente forma:

```
@attribute <nombre_atributo> <Tipo_atributo>
```

Los tipos de atributos que define Weka son:

STRING: Para expresar cadenas de texto.

NOMINAL: Para expresar los posibles valores que puede alcanzar una variable. Se enuncian entre llaves y separados por coma.

NUMERIC: Para expresar los números reales.

INTEGER: Para expresar los datos que representan números enteros.

DATE: Expresa fechas, para ello debe ir precedido de una etiqueta de formato entrecomillada la cual está compuesta por caracteres separadores y unidades de tiempo: dd Día, MM Mes, yyyy Año, HH Horas, mm Minutos, ss Segundos.

La última sección es donde se declaran las instancias que forman la relación. El inicio de la sección viene dado de la forma @data. Los atributos que forman las instancias se separan por coma y las instancias por saltos de líneas.

En la figura se muestra un ejemplo de un fichero ARFF que cumple con la estructura propuesta por Weka.

```
@relation Estudiantes_navegante

@attribute nombre STRING
@attribute apellido STRING
@attribute edad INTEGER
@attribute fecha_nacimiento date "dd-MM-yyyy"
@attribute evaluaciones nominal {B,R,M}

@data
Gilber,Benitez,23,8-5-1989,B
Yunior, Baez, 25,11-12-1989,B
Lili,Gautier,22,20-5-2004,M
Yadira,Diaz,7,20-1-2012,R
Yelena,Rodriguez,15,22-8-2007,M
```

Ilustración 4: Ejemplo de fichero ARFF

2.7 Resultados y evaluación

En el siguiente epígrafe se describen las tres últimas fases del proceso de extracción de conocimientos que propone la metodología Crisp-DM (modelado, evaluación y despliegue). Se analizan y

desarrollan las actividades propias de cada una de estas fases; además de realizar una evaluación de los resultados alcanzados con su aplicación.

2.7.1 Modelado

En esta fase se aplican las técnicas de modelado o de minería de datos seleccionadas teniendo en cuenta las características de los datos o vista minable previamente seleccionada. También se incluye una evaluación desde el punto de vista de precisión de los modelos generados.

Selección de las técnicas de modelado

Es uno de los aspectos más importantes dentro del proceso debido a que constituye la base sobre la cual se van a sustentar las posteriores actividades. Se selecciona la técnica de minería que más se ajusta a las características de los datos con los que se trabajan y además teniendo en cuenta el tipo de problema a resolver y los resultados que se esperan. Como tareas del proceso de minería de datos se seleccionaron agrupamiento (*clustering*) y reglas de asociación.

Agrupamiento o *clustering*:

Algoritmo Simple k-medias

Es un algoritmo basado en particiones. Los algoritmos de *clustering* basados en particiones intentan buscar una división del conjunto de datos en subconjuntos con intersección vacía. Todos ellos siguen un patrón común que consiste en realizar una asignación de los objetos a los diferentes *clusters* en función de la proximidad de dichos objetos a un representante elegido para cada *cluster* (51). Se caracteriza además por ser no jerárquico, lo cual permite realizar una partición de los datos en k grupos; el investigador debe fijar el número de grupos que deben formarse. Dado sus características es uno de los más utilizados en la rama de la minería de datos. El objetivo del algoritmo es dividir el conjunto X de los objetos en un cierto número (K) de subconjuntos naturales y homogéneos, donde los elementos de cada conjunto son tan similares como sea posible entre ellos y que, al mismo tiempo, sean lo más distinto posible a los demás integrantes de X (52).

Parámetros de configuración

displayStdDevs: Muestra las desviaciones de los atributos numéricos y cuenta los atributos nominales. Este parámetro es activado para mostrar al usuario dicha desviación.

distanceFunction: Este atributo selecciona la función de distancia, por lo que se registrará la semejanza/desemejanza de los objetos, se mantiene el valor que se carga por defecto EuclideanDistance.

dontReplaceMissingValues: Sustituir los valores faltantes por la media. Este atributo se coloca en *false*, ya que los datos omisos son tratados en la etapa de pre-procesamiento de datos.

numClusters: Cantidad de clúster a generar, se le brindará esta opción al usuario para que pruebe la formación de distintos grupos y observar el comportamiento del mismo.

maxIterations: Número máximo de iteraciones, se pondrá en 1000 para aumentar la fiabilidad del método.

preserveIntanceOrder: Preservar el orden de las instancias, se mantiene el valor que trae por defecto.

Seed: La semilla del número aleatorio que se utiliza, se mantiene en 10, que es el valor que trae por defecto.

Algoritmo DBScan:

Se caracteriza por ser un algoritmo basado en densidad. Estos algoritmos se basan en el concepto de densidad de un punto, que mide el número de puntos que son alcanzables desde él considerando un determinado radio (51) y 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 (53).

Parámetros de configuración:

Eps: Radio máximo de la vecindad. Este valor se mantiene con su valor por defecto (0.9).

MinPts: Número mínimo de puntos en un vecindario de ese punto, considerando el radio Eps. Este valor es definido por el usuario.

Técnicas basadas en casos, en densidad o distancia

Los algoritmos Kmeans y DBScan basan su funcionamiento en técnicas basadas en casos, densidad o distancia. Son métodos que se basan en distancias al resto de elementos, ya sea directamente, como los vecinos más próximos (los casos más similares), de una manera más sofisticada, mediante la estimación de funciones de densidad. Además de los vecinos más próximos, algunos algoritmos muy conocidos son los jerárquicos, como Two-step o COBWEB.

Reglas de asociación:

Algoritmo Apriori

Este algoritmo se basa en la búsqueda de los conjuntos de ítems con determinada cobertura. Para ello, en primer lugar se construyen simplemente los conjuntos formados por sólo un ítem que supera la cobertura mínima. Este conjunto de conjuntos se utiliza para construir un conjunto de dos ítems, y así sucesivamente hasta que se llegue a un tamaño en el cual no existan conjuntos de ítems con la cobertura requerida (6).

La propiedad plantea que todo subconjunto no vacío de un conjunto de ítems frecuentes tiene que ser frecuente también. Si un conjunto de ítems X no satisface el soporte mínimo entonces no es frecuente y si a este conjunto X se le adiciona un ítem entonces el conjunto resultante ocurrirá con menor o igual frecuencia que X (53).

Características fundamentales:

- Es uno de los algoritmos de reglas de asociación más utilizados.
- Solo es capaz de buscar reglas de asociación a través de atributos nominales por lo que se hace necesario convertir los atributos numéricos a nominales.
- Mejora la eficiencia en la búsqueda de conjuntos de ítems frecuentes eliminando los atributos que tengan sus valores desconocidos en todos los resultados.

Parámetros de configuración:

Car: Brinda la posibilidad de seleccionar si se quiere hacer las reglas de asociación de forma general tomando todos los atributos como clase o tomando un solo atributo como clase. Para el desarrollo de la solución se escogieron todos los atributos para garantizar la obtención de un mayor número de reglas.

ClassIndex: En caso de seleccionar un solo atributo como clase en el parámetro anterior en este se especifica que atributo se desea tomar. Por defecto trae -1 que significa que se va a tomar como clase el último atributo.

Soporte Mínimo: Se establece el soporte mínimo que la regla debe cumplir. Un soporte mínimo muy alto no arrojaría ningún resultado por lo que se definió 0,1 (mínimo).

Delta: Iterativamente se decrece el valor del soporte máximo hasta llegar al soporte mínimo por este parámetro para el cual se definió 0,05.

Tipo de métrica: Esta opción brinda la posibilidad de seleccionar el tipo de métrica que se quiere utilizar, en este caso se indica la confianza para poder determinar el porcentaje de la calidad de las reglas.

Mínimo de la métrica: Mediante este parámetro se introduce la confianza mínima que las reglas a obtener tengan que cumplir. Este valor debe ser pequeño para poder obtener reglas que no se descubran a simple vista y así obtener información interesante este, se definió en 0,4.

Número de reglas: Indica el número máximo de reglas a obtener, es decir, se utiliza como criterio de parada para detener la ejecución si se llega a este número de reglas cumpliendo las restricciones anteriores. Este valor de configuración, por ser simple, el usuario tendrá la opción de escoger la cantidad máxima de reglas a mostrar.

Soporte máximo: Se establece un soporte máximo a partir del cual se va a empezar a decrecer hasta llegar al soporte mínimo establecido, para el cual se definió 1,0 (máximo).

Algoritmo PredictiveApriori

Apriori Predictivo es una mejora al algoritmo Apriori y resuelve el problema de manera similar. Se trata de encontrar las N mejores reglas de asociación, donde N es un número fijo. Se busca entonces un balance adecuado entre el soporte y la confianza de forma que maximice la probabilidad de hacer una predicción correcta sobre el conjunto de datos.

Técnicas basadas en conteos de frecuencias y tablas de contingencia

Los algoritmos Apriori y PredictiveApriori basan su funcionamiento en técnicas basadas en conteos de frecuencias y tablas de contingencia. Estas técnicas se basan en contar la frecuencia en la que dos o más sucesos se presentan conjuntamente. Cuando el conjunto de sucesos posibles es muy grande, existen algoritmos que van comenzando por pares de sucesos e incrementando los conjuntos sólo en aquellos casos que las frecuencias conjuntas superen un cierto umbral.

Los algoritmos fueron seleccionados basándose fundamentalmente en el reconocimiento y la aceptación con la que cuentan a nivel internacional, así como, las características de los mismos y su forma de visualizar los resultados en la aplicación de los algoritmos a los datos seleccionados.

2.7.2 Evaluación

Al llegar a esta fase se analizarán los modelos seleccionados para medir su precisión y rendimiento para decidir si cumplen apropiadamente con los objetivos del negocio propuestos en la primera fase. Para lograr esto la evaluación se apoya de dos subfases: la evaluación de los resultados (evaluación de los resultados de la aplicación de la minería de datos, modelos aprobados) y la revisión del proceso (6).

Evaluación de los resultados

Una vez aplicados los modelos se obtuvieron resultados que aportan información interesante a los profesores acerca del acceso de los estudiantes a los distintos temas del módulo Contenidos y sobre la interacción de los mismos con los restantes módulos de la colección. A través de estas informaciones los profesores pueden determinar cuáles son los temas más visitados y cuáles son los menos visitados, lo que brinda la posibilidad de incorporar a los temas que más visitan los alumnos aquellos conocimientos que por su importancia resultan relevantes y necesarios a tener en cuenta en el proceso de aprendizaje. Además resulta interesante analizar el acceso a los distintos módulos con el fin de estructurar el funcionamiento de la colección en general.

Revisión del proceso

Con la fase de la comprensión del negocio, que constituye su etapa inicial, centrada en entender los objetivos y metas fundamentales del negocio se establece como objetivo fundamental del negocio, generar un reporte que sea capaz de exponer de forma clara y sencilla elementos importantes y de relevancia almacenados en el módulo Resultados de la colección El Navegante.

Se dio paso a una de las fases más importantes del proceso, la selección de los datos, que por su relevancia se deben tener en cuenta a la hora de realizar el proceso; el acceso de los estudiantes al módulo Contenidos y los restantes módulos de la colección.

Una vez seleccionados los datos se procede a la preparación de estos para poder ser interpretados correctamente por la herramienta de minería de datos, transformándolos siguiendo la estructura del fichero de extensión ARFF que propone Weka.

En la siguiente fase se aplican las tareas y algoritmos de minería de datos seleccionados teniendo en cuenta las características de los datos que se eligieron. Se decide la realización de tareas basadas en agrupamientos y reglas de asociación. Se seleccionaron como algoritmos para la aplicación de estas tareas, relacionados con el agrupamiento a K-medias y DBScan, y por parte de las reglas de asociación los algoritmos Apriori y PredictiveApriori.

2.7.3 Despliegue

En esta fase es donde se trata de explotar las potencialidades y garantías obtenidas de la aplicación de los modelos y la forma más idónea de integrar los conocimientos descubiertos a los sistemas de las organizaciones interesadas. Se utilizan los datos generados por los modelos para mostrar información más importante relacionada con los estudiantes.

Se muestra los resultados de las tareas de agrupamiento a través de gráficos que reflejan los distintos grupos que resultaron de la aplicación de los algoritmos y también las reglas generadas por las tareas de reglas de asociación, todo esto de la forma más representativa posible.

2.8 Conclusiones del capítulo

La realización del modelo de dominio y la descripción de los conceptos asociados al mismo, fueron actividades que contribuyeron a una correcta captura de requisitos, lográndose al mismo tiempo un entendimiento pleno entre desarrolladores y usuarios. Como parte de la descripción del sistema se identificaron y modelaron los casos de uso que guiarán el proceso de desarrollo del *software*; los cuales se describieron detalladamente de acuerdo a lo que plantea la metodología en uso. Partiendo de la investigación realizada y siguiendo el flujo de trabajo propuesto por la metodología Crisp-DM se definieron los algoritmos y variables que mejor satisfacen las necesidades del problema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se realiza el análisis y diseño del sistema a desarrollar. Se muestran los principales artefactos generados, siguiendo el flujo de trabajo análisis y diseño propuesto por la metodología RUP, lo que será de gran ayuda para la comprensión de la realización del sistema. Se describen los diagramas de clases del análisis y el diseño asociados a los CU que fueron definidos previamente. Se incluyen además los diagramas de colaboración asociados a los CU del sistema. Se realiza un análisis de los patrones arquitectónicos y de diseño del sistema y se describe igualmente el diagrama Entidad-Relación asociado al sistema.

3.2 Análisis del sistema

El análisis abarca un conjunto de tareas de trabajo (por ejemplo, la investigación, elaboración, negociación, especificación y validación de requisitos) que conducen a la creación del modelo de análisis (o a la especificación de requisitos) (22).

3.2.1 Modelo de análisis

El modelo de análisis tiene como objetivo comprender y generar una arquitectura de objetos para el sistema con base en lo especificado en el modelo de requisitos. Es una representación conceptual, correspondiente al problema y al modelo de requisitos, en términos de clases de objetos. Cada una de estas clases contribuye de manera especial para lograr la arquitectura deseada (47).

3.2.2 Diagrama de clases del análisis

Los diagramas de clases del análisis representan una abstracción de los conceptos más significativos del proceso de negocio en el problema identificado, en el que se muestran los usuarios, las entidades y las relaciones entre ellos. A continuación se expone el diagrama de clases del análisis asociado al CU Realizar proceso de extracción de conocimientos. Los diagramas asociados a los restantes casos de uso se reflejan en los anexos del documento **(Ver Anexo 3)**

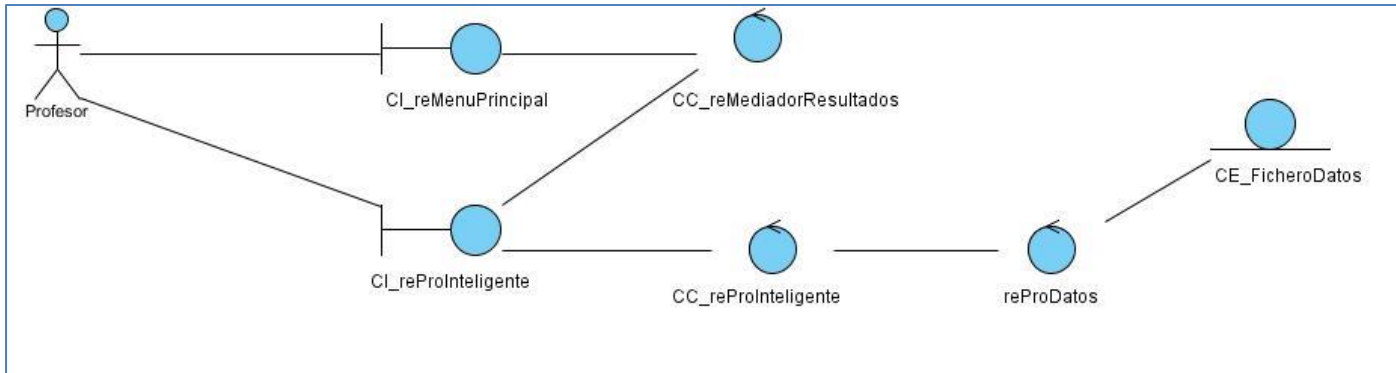


Ilustración 5: CU1: Realizar proceso de extracción de conocimientos

3.3 Diagrama de colaboración

Los diagramas de colaboración se caracterizan por describir el comportamiento de los objetos y se resalta la organización estructural y el intercambio de mensajes entre estos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto a los demás (54). A continuación se muestra el diagrama de colaboración correspondiente al CU Realizar proceso de extracción de conocimientos. Los restantes diagramas de colaboración se exponen en los anexos del documento **(Ver Anexo 4)**

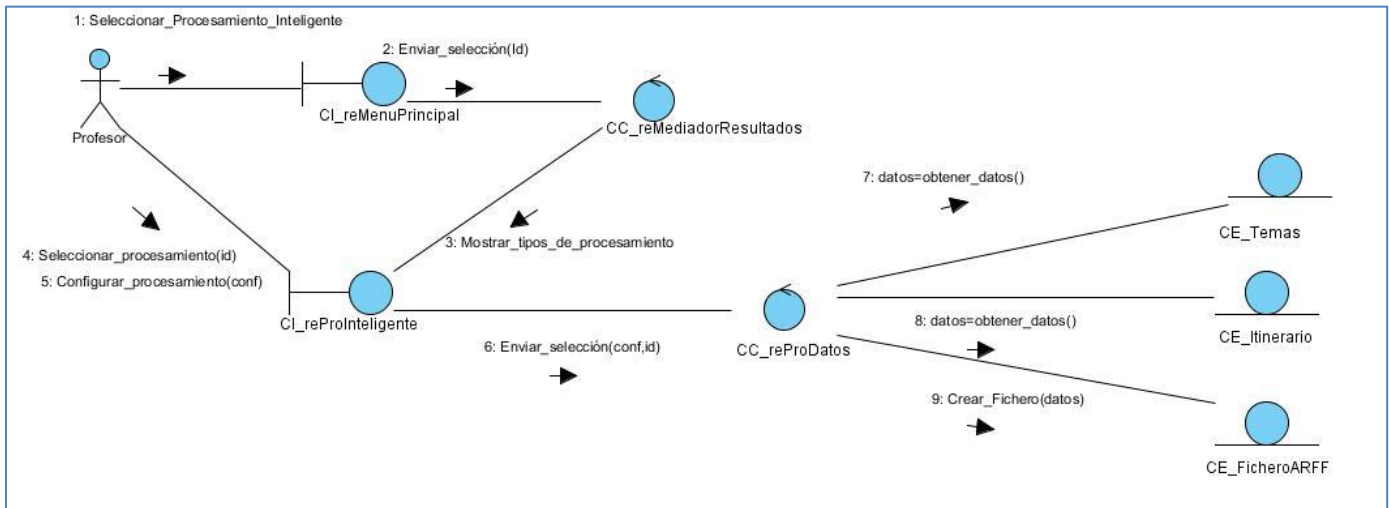


Ilustración 6: Diagrama de colaboración CU1: Realizar proceso de extracción de conocimiento

3.4 Arquitectura del diseño

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares (55).

3.4.1 Patrones arquitectónicos

Estos patrones definen la estructura general del *software*, indican las relaciones entre los subsistemas y los componentes del *software* y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura (22).

Para el desarrollo del sistema se utiliza el patrón arquitectónico Modelo-Vista-Controlador (MVC) en el cual está basado el *framework* Symfony. MVC es un patrón arquitectónico que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el SGBD y la lógica de negocio, el controlador es el responsable de recibir los eventos de entrada desde la vista y realizar todas las operaciones necesarias para garantizar la comunicación entre los componentes del sistema. (56) Por tanto, el Modelo, las Vistas y los Controladores se tratan como entidades independientes donde (32):

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

3.4.2 Patrones de diseño

Estos patrones se aplican a un elemento específico del diseño como un agregado de componentes para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente (22).

El *framework* Symfony utiliza en su implementación un conjunto de patrones de diseño para dar solución a problemas específicos del diseño orientado a objetos durante el flujo de ejecución de una petición.

Patrón Controlador frontal

El patrón de diseño Controlador frontal se basa en utilizar un controlador como punto inicial para la gestión de las peticiones. El controlador gestiona estas peticiones, y realiza algunas funciones como: comprobación de restricciones de seguridad, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario.

Las aplicaciones Symfony utilizan el patrón de diseño Controlador frontal, con el objetivo de garantizar un único punto de entrada a cada aplicación. De esta manera todas las peticiones que realiza el usuario se controlan mediante un único *script* de PHP, que se encarga de redirigir las peticiones a cada módulo y acciones correspondientes en cada caso (32).

Patrón Mediador

Un Mediador es un patrón que define un objeto y encapsula cómo interactúan un conjunto de objetos. El Mediador estimula la pérdida de acoplamiento, ocultando las referencias explícitas entre los objetos, permitiendo variar su interacción de forma independiente. Este patrón de diseño está considerado como un patrón de comportamiento debido al hecho de que puede alterar el comportamiento del programa en ejecución (56).

3.5 Modelo de diseño

Es un modelo físico y concreto que constituye una aproximación del diseño del sistema centrándose en la realización de los requisitos funcionales y no funcionales, así como, la realización de los casos de uso del sistema, teniendo en cuenta los requerimientos arquitectónicos.

3.5.1 Diagrama de clases del diseño

Los diagramas de clases del diseño constituyen un modelo que representa la estructura física de la realización de los casos de uso del sistema, así como, las relaciones existentes entre los distintos elementos. Muestra además las operaciones o métodos de los componentes del diagrama. Los elementos fundamentales del diagrama se representan en la siguiente tabla:




Clases	Estereotipos	Función
Server Page [SP] (en español Página Servidora)		Representa la página web que tiene código que se ejecuta en el servidor. Su principal función es construir la Página cliente.
Client Page [CP] (en español Página Cliente)		Representa una página Web, con formato HTML. Son interpretadas por el navegador, su función es visualizar, interactuar y mostrar lo que el usuario necesita.
Form (en español Formulario)		Colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario. Su principal función es enviar los datos entrados por el usuario a la Página servidora.

Tabla 7: Clases del diseño

A continuación se representan los diagramas de clases de diseño asociados a los CU que fueron definidos previamente:

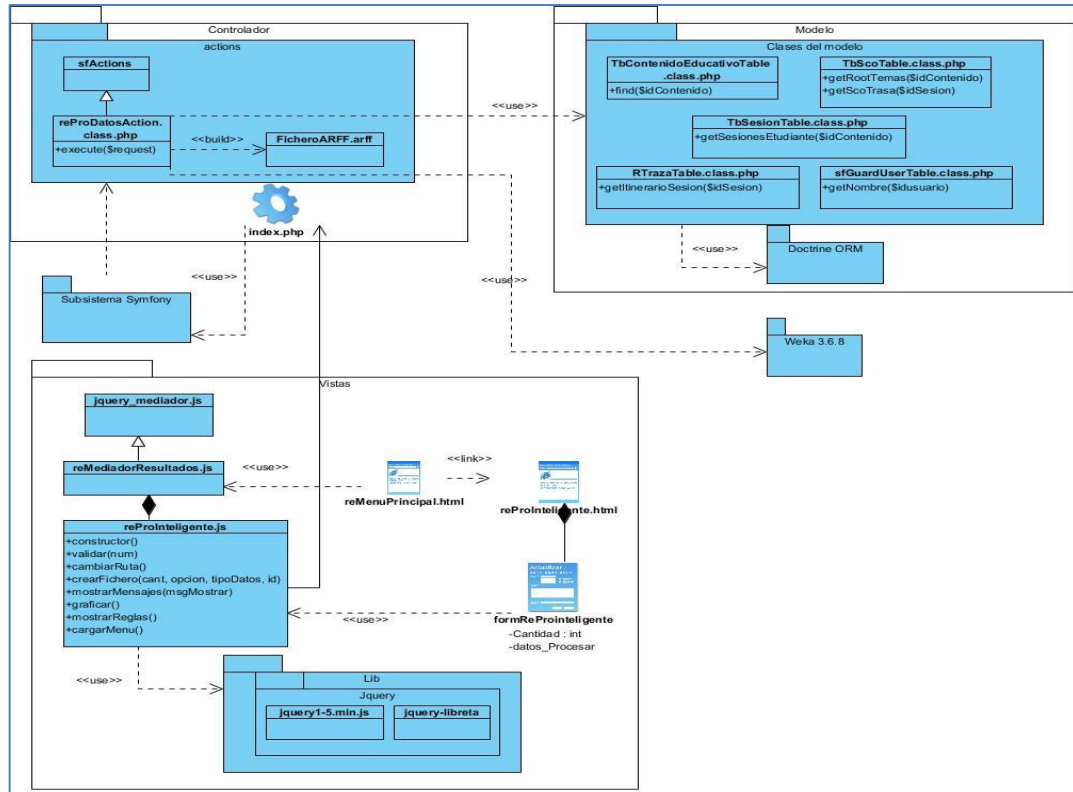


Ilustración 7: Diagrama de clases del diseño: CU Realizar proceso de extracción de conocimiento

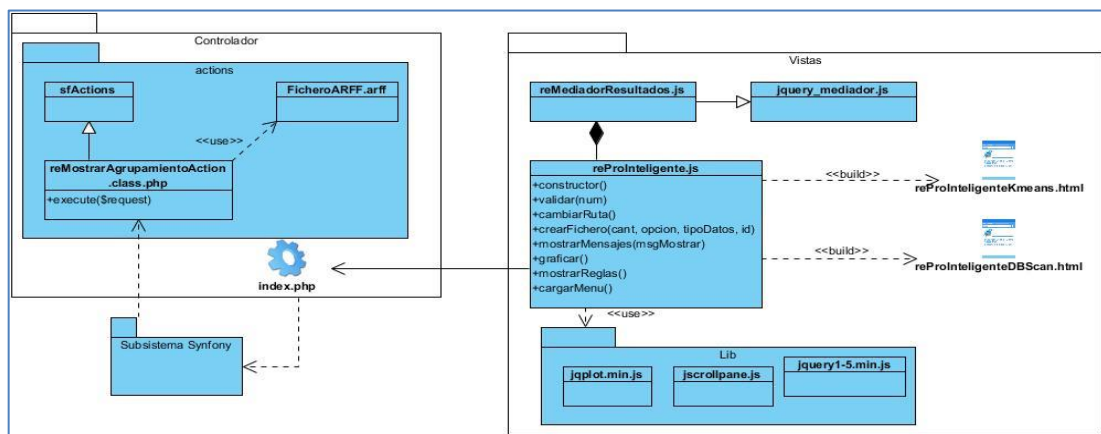


Ilustración 8: Diagrama de clases del diseño: CU Mostrar resultados Agrupamiento

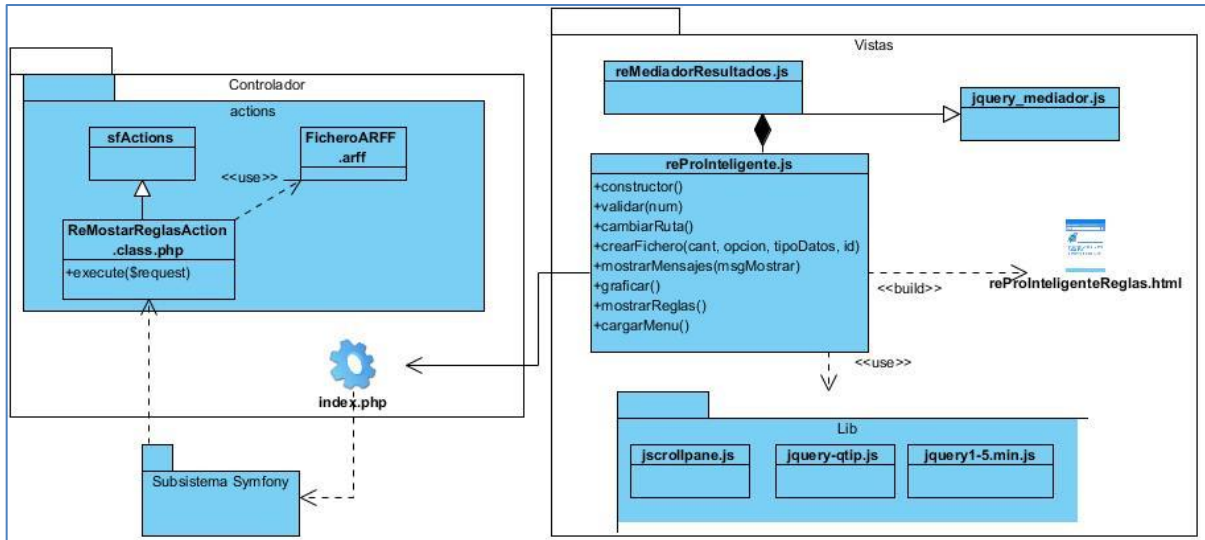


Ilustración 9: Diagrama de clases del diseño: CU Mostrar resultados Reglas

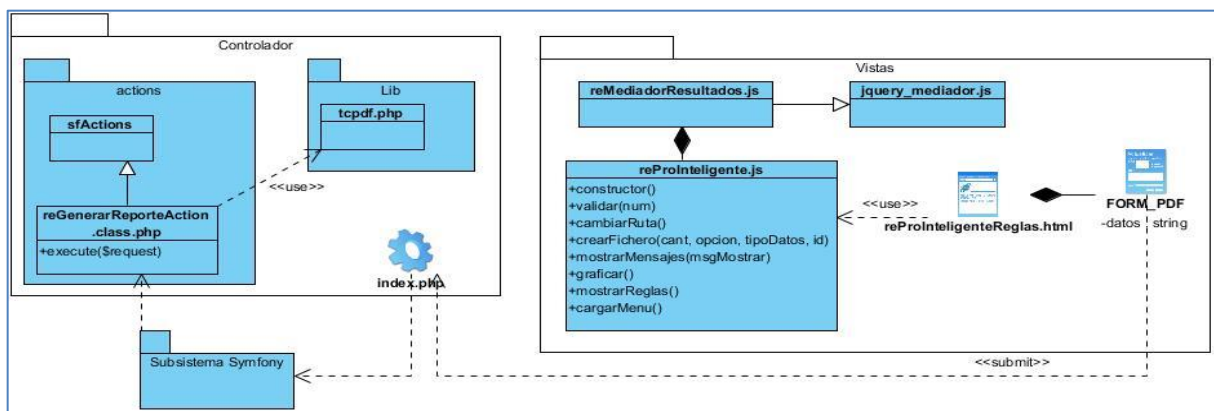


Ilustración 10: Diagrama de clases del diseño: CU Generar reporte

3.6 Diagrama de secuencia

Los diagramas de secuencia, también conocidos como diagramas de interacción o de eventos, los cuales describen los diferentes casos de uso según la interacción o eventos enviados entre los objetos de la arquitectura del modelo de análisis. Estos diagramas describen aspectos dinámicos del sistema, a diferencia de los diagramas de clases que utilizan clases como elementos básicos (47).

A continuación se muestran los diagramas de secuencias asociados a los casos de uso identificados:

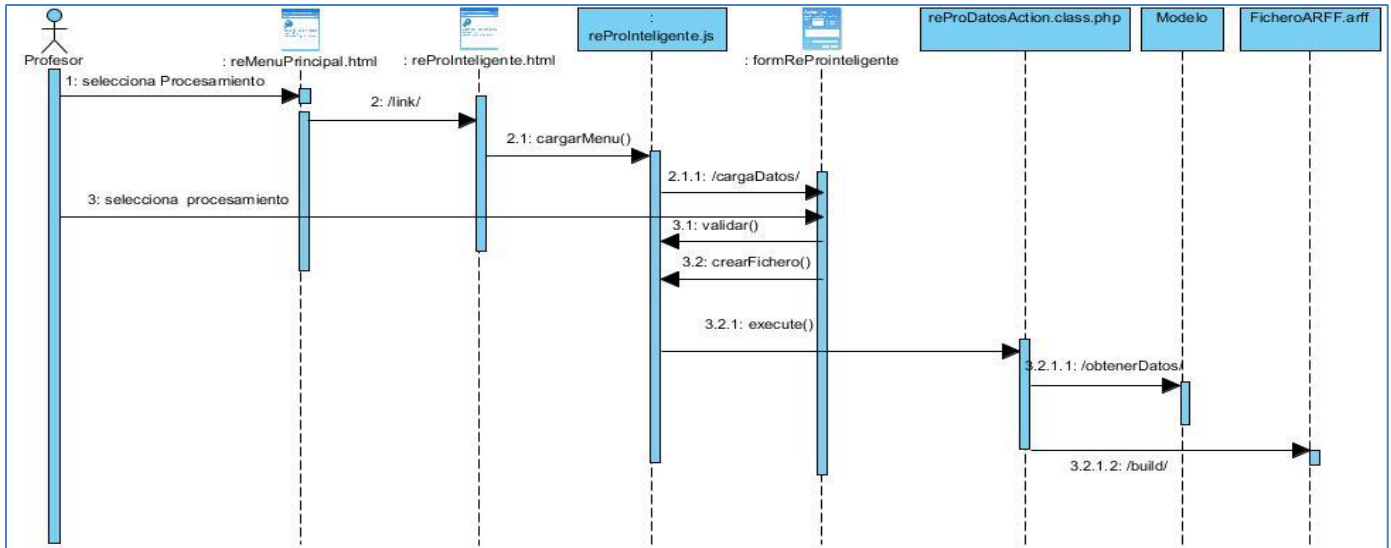


Ilustración 11: Diagrama de secuencia CU Realizar proceso de extracción de conocimiento

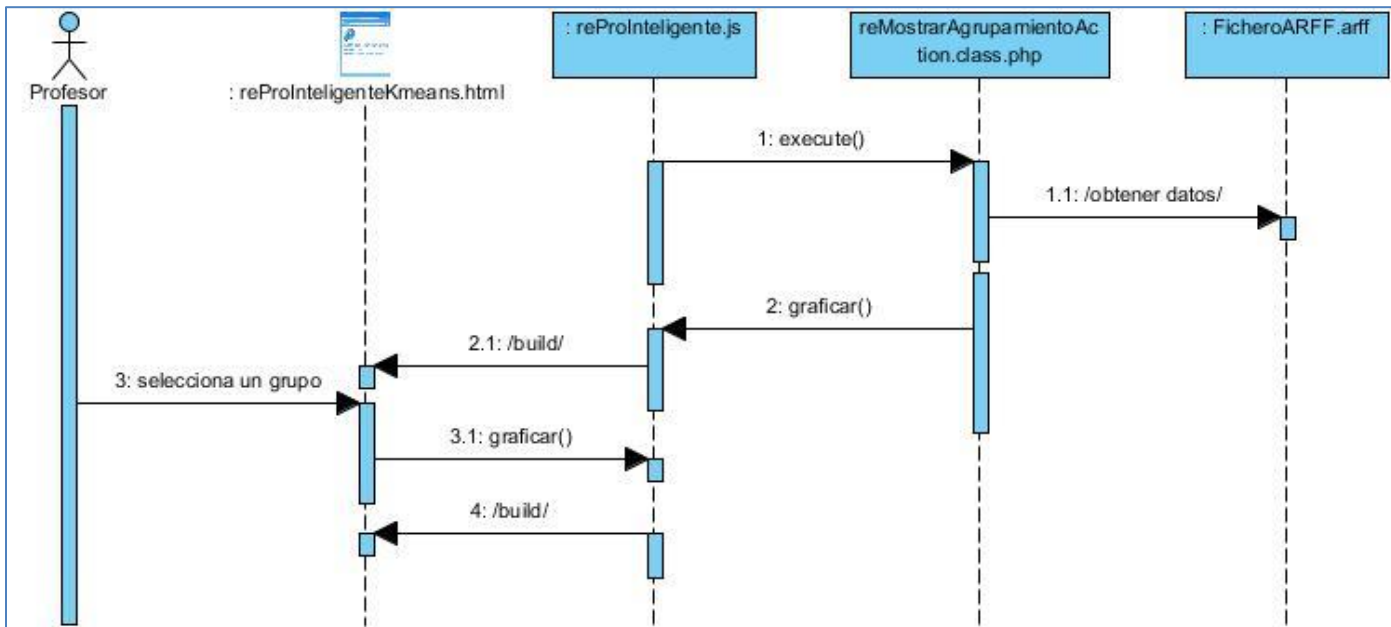


Ilustración 12: Diagrama de secuencia CU Mostrar Resultados Agrupamiento: Escenario 1

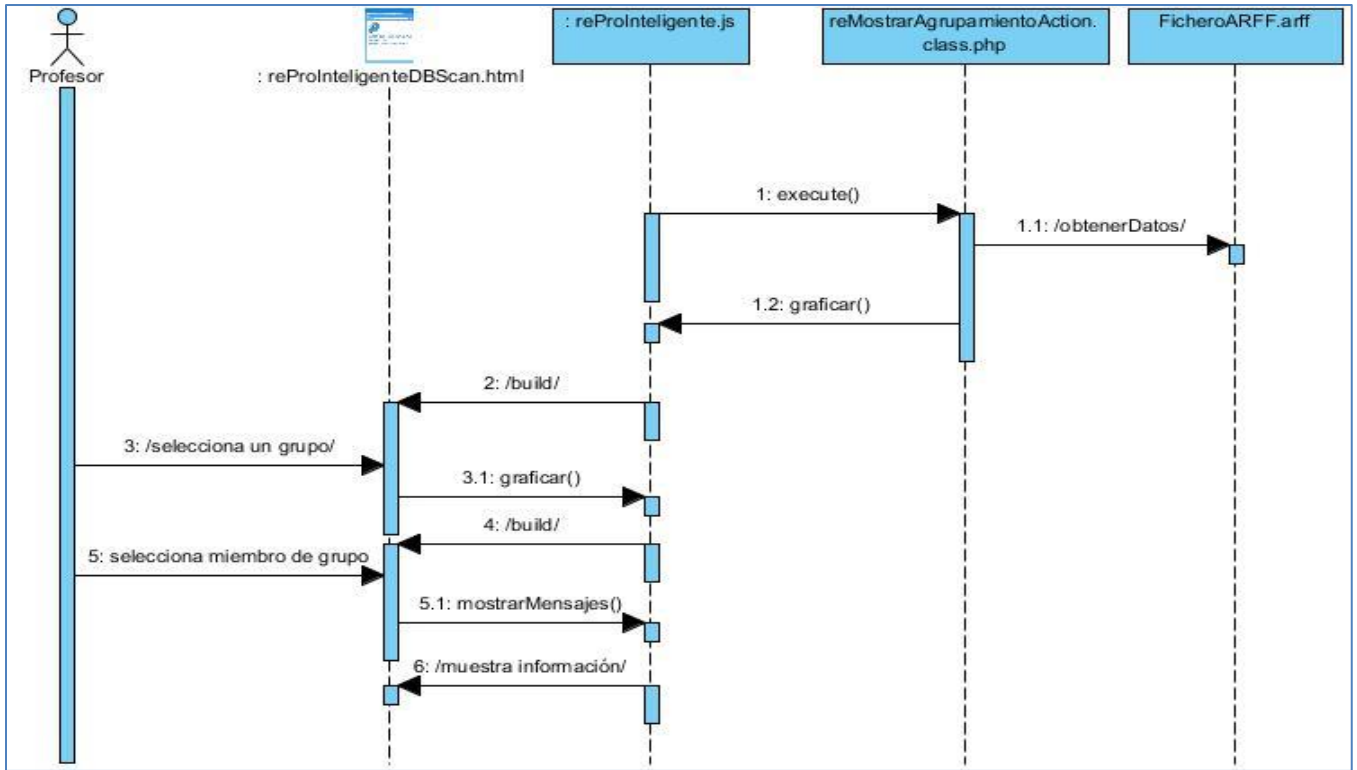


Ilustración 13: Diagrama de secuencia CU Mostrar Resultados Agrupamiento: Escenario 2

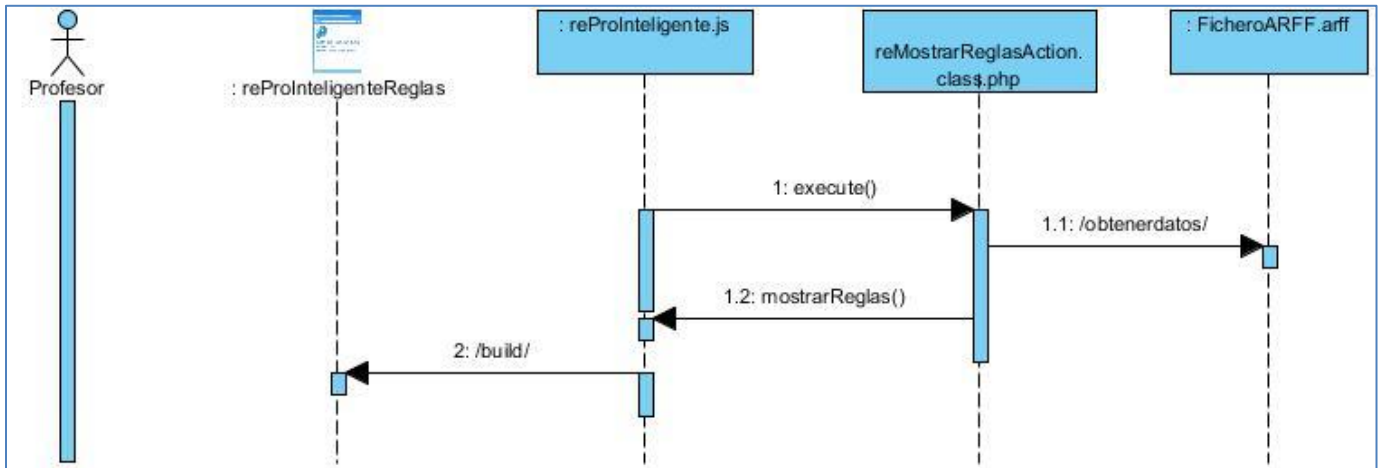


Ilustración 14: Diagrama de secuencia CU Mostrar Resultados Reglas

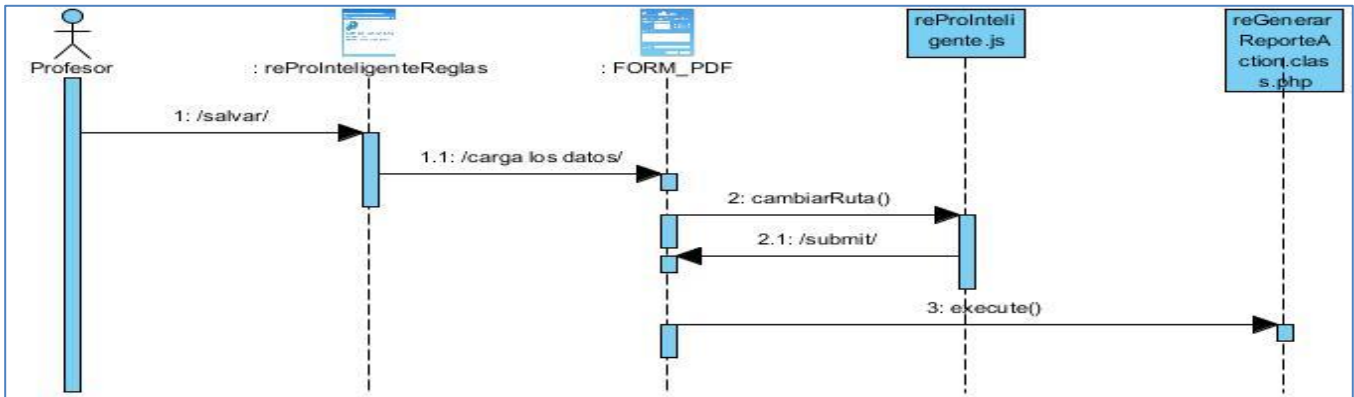


Ilustración 15: Diagrama de secuencia CU Generar reporte

3.7 Diseño de la base de datos

Un correcto diseño y organización de la base de datos es uno de los elementos más importantes dentro de la estructura de un sistema informático. Dentro de sus funciones principales se encuentran la limpieza y perfeccionamiento de los datos evitando que exista redundancia en la información, así como, la existencia de datos innecesarios almacenados. Este procedimiento se realiza con el objetivo de simplificar la complejidad y aumentar la funcionalidad de las bases de datos.

3.7.1 Modelo Entidad-Relación

El Modelo Entidad-Relación (MER) constituye una representación gráfica de las entidades y sus relaciones en una estructura de base de datos. Es considerado además una herramienta de modelado de datos a nivel conceptual y complementa los conceptos de modelo de datos relacional, con los que se establecieron las bases para un ambiente de diseño de bases de datos bien estructurado que garantice el diseño apropiado de las bases de datos relacionales (57).

La figura 22 muestra una porción del MER asociado a la colección El Navegante en la cual se encuentran representados los datos utilizados en el proceso de extracción de conocimientos. En la tabla *tb_sco* se generalizan las diferentes clasificaciones de objetos educativos que se manejan en el sistema, como son los elementos de la Mediateca (imágenes, videos, audios, diaporamas, animaciones, palabras del glosario y personalidades) y los artículos del módulo Contenidos. En la tabla *tb_sesion* se registran los datos asociados a las distintas sesiones de trabajo (hora de inicio y fin, usuario y el contenido educativo).

La tabla r_traza muestra la relación existente entre los distintos objetos educativos y las sesiones de trabajo.

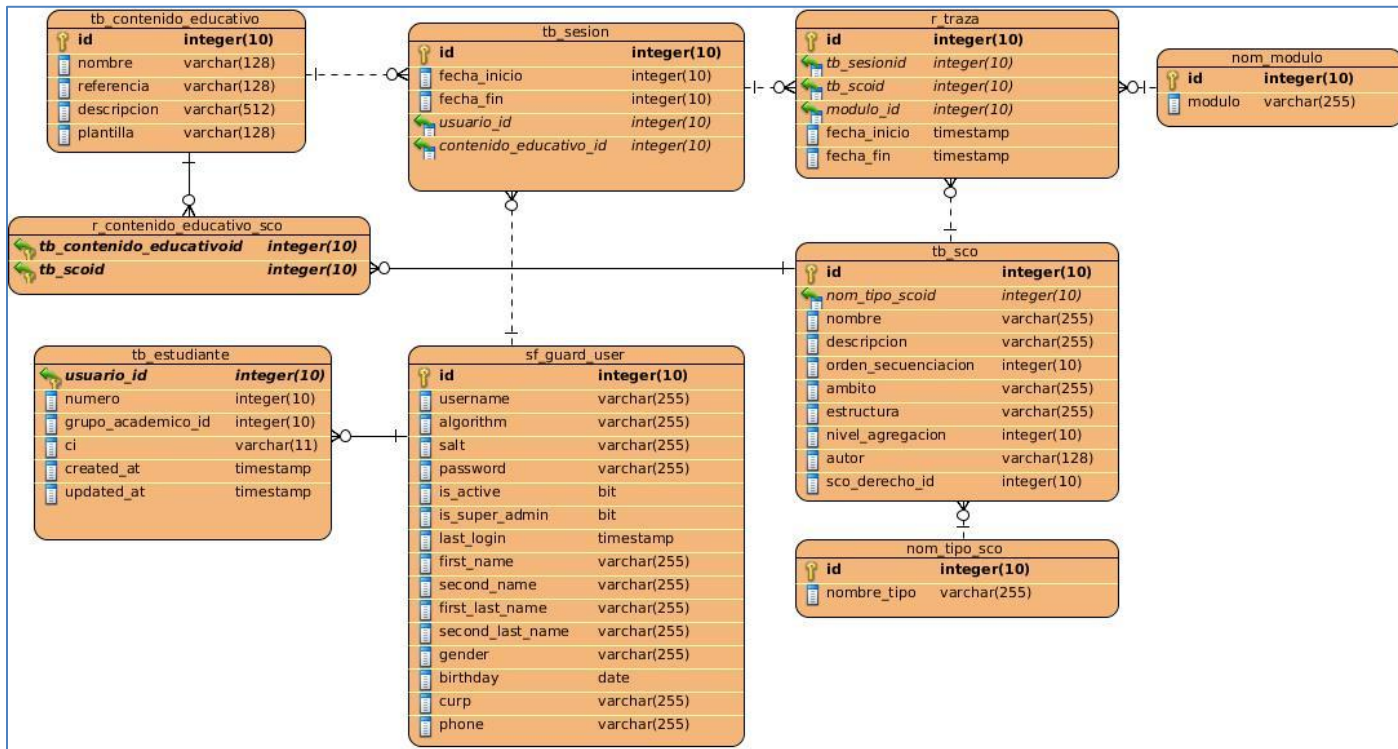


Ilustración 16: Diagrama Entidad-Relación

3.8 Conclusiones del capítulo

La obtención de los artefactos generados como resultados del flujo de trabajo de análisis y diseño propuesto por la metodología RUP brinda una aproximación del funcionamiento del sistema como fase previa al desarrollo de la aplicación, fundamentada en un entendimiento pleno entre los desarrolladores y clientes. El establecimiento de patrones arquitectónicos y de diseño posibilita la estructuración de forma general del sistema, así como, las relaciones entre sus componentes garantizando el desarrollo de forma organizada.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

4.1 Introducción

Los artefactos generados en el flujo de trabajo de análisis y diseño propuesto por la metodología RUP, constituyen una base para la implementación y posteriores pruebas del sistema. En el presente capítulo se procede a describir el modelo de implementación de la aplicación basándose en los diagramas de componentes y de despliegue del sistema. Se muestra además la fase de prueba de la aplicación fundamentada sobre las pruebas de caja negra, reflejada a través de los distintos casos de prueba.

4.2 Modelo de implementación

El modelo de implementación describe la forma en la que los elementos definidos en el modelo de diseño, principalmente las clases, se implementan en forma de componentes (archivos, ejecutables, librerías) y la dependencia existente entre estos componentes. Constituye además la base fundamental para las posteriores pruebas a la aplicación.

4.2.1 Diagrama de componentes

El diagrama de componentes representa un conjunto de los elementos del sistema en forma de componentes (clases, librerías, archivos, bases de datos), además de subsistemas de implementación y las relaciones existentes entre estos.

El diagrama de componentes de la solución propuesta se divide en paquetes de clases, mostrando la relación existente entre cada una de las clases identificadas y evidenciando legibilidad en la construcción de cada uno de los paquetes.

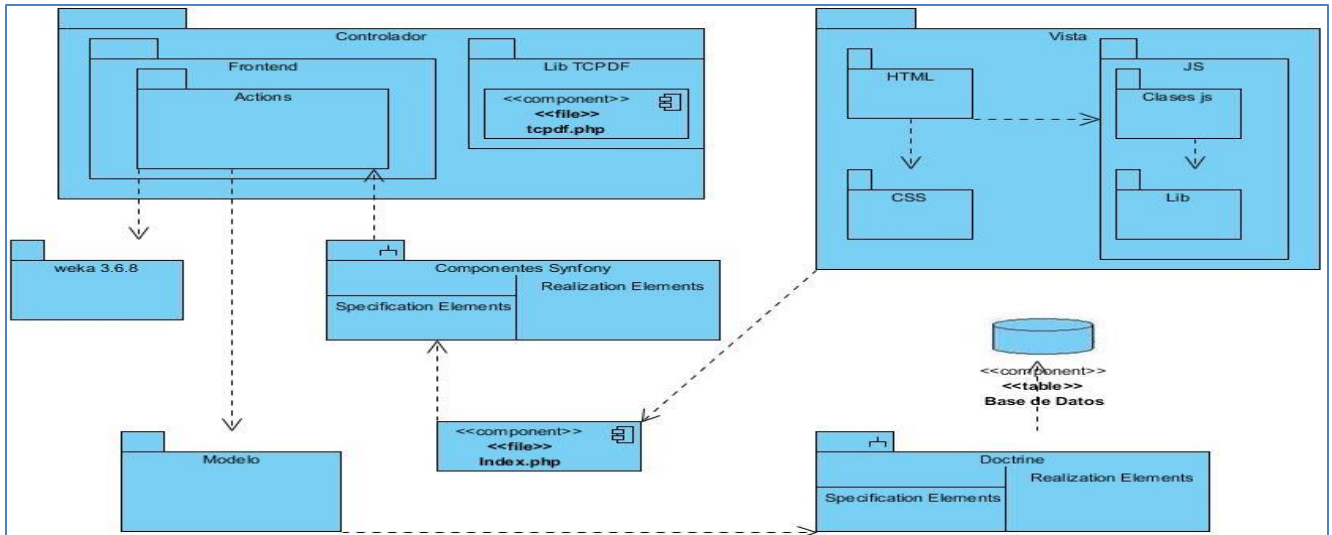


Ilustración 17: Diagrama de componentes del sistema

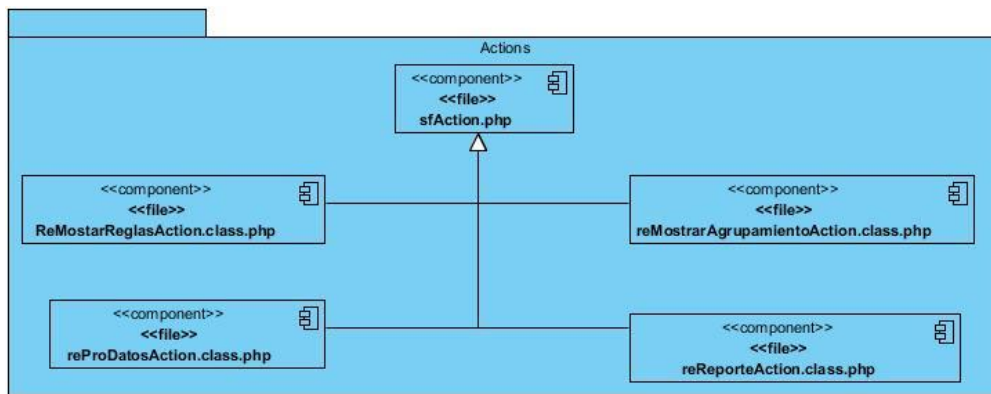


Ilustración 18: Diagrama de componentes del paquete actions

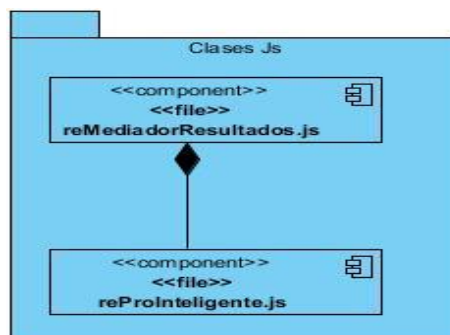


Ilustración 19: Diagrama de componentes del paquete Clases Js

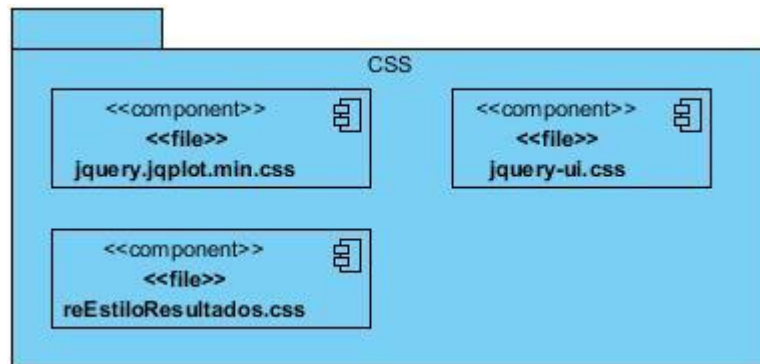


Ilustración 20: Diagrama de componentes del paquete CSS

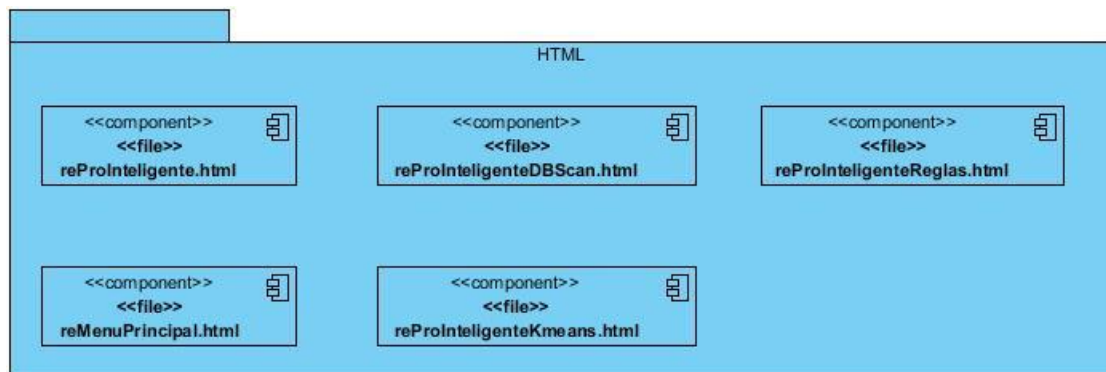


Ilustración 21: Diagrama de componentes del paquete HTML

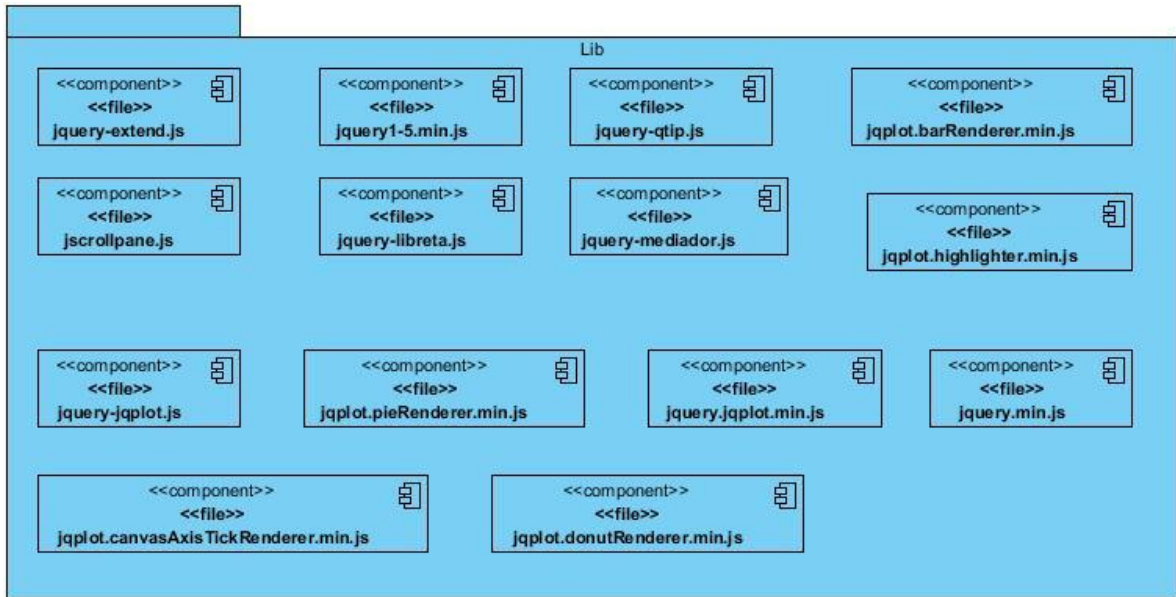


Ilustración 22: Diagrama de componentes del paquete Lib

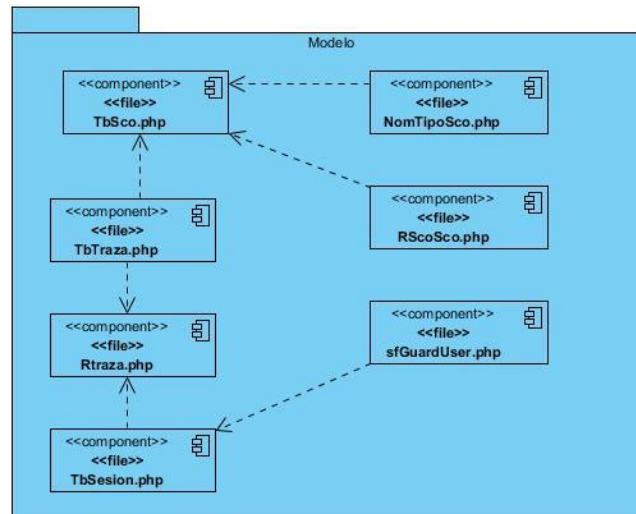


Figura 29: Diagrama de componentes del paquete Modelo

4.2.2 Diagrama de despliegue

El diagrama de despliegue constituye una representación de los elementos físicos que integran el sistema, en forma de componentes y las relaciones existentes entre ellos. Muestra una aproximación de la

distribución de los nodos (componentes del sistema), así como, los protocolos de comunicación a través de los cuales se relacionan.

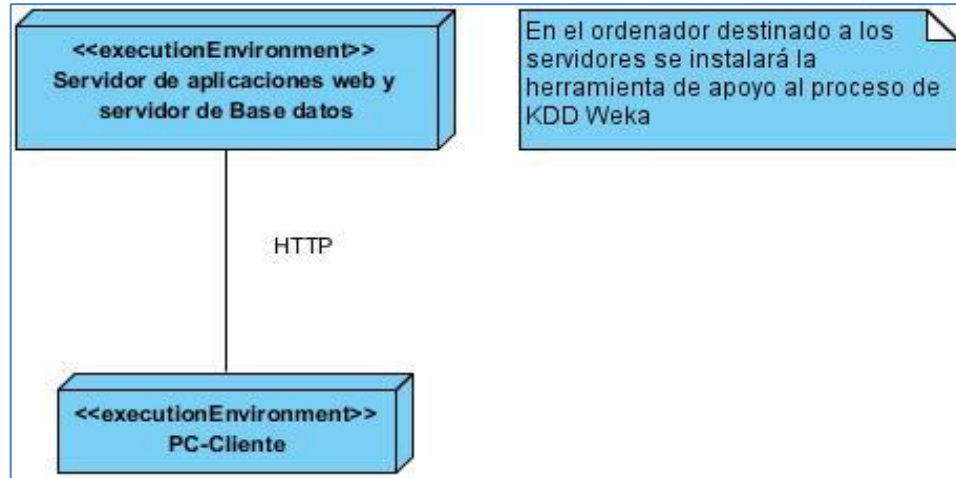


Ilustración 23: Diagrama de despliegue

4.3 Estándares de codificación

Los estándares de codificación son reglas que se utilizan para la escritura del código fuente. Estas permiten que todos los programadores del proyecto trabajen de forma coordinada y en un vocabulario común. Además aseguran la legibilidad del código y proporcionan una guía para el encargado del mantenimiento o actualización del sistema, con código claro y bien documentado.

Teniendo en cuenta que el desarrollo de la solución forma parte del módulo Resultados de la colección El Navegante se establecen las normas de codificación siguiendo la estructura del módulo garantizando así la uniformidad en el módulo.

4.3.1 Convenciones de nombres (Funciones y Clases)

Los nombres que se utilizan son cortos, y de forma abreviada, siempre que su contexto sea específicamente local y su lectura sea intuitiva. Para los nombres se establecen las siguientes reglas:

Funciones y métodos:

Se utiliza la notación lowCamelCase donde la primera letra de cada una de las palabras es mayúscula a excepción de la primera. Ejemplo: crearFichero.

Clases:

Los nombres de las clases comienzan con el prefijo del módulo, en este caso del módulo Resultados, seguidos de la palabra Pro y luego del formato CamelCase, ejemplo: reProDatos.php.

4.3.2 Indentación y espacios en blanco

Indentar con dos espacios, sin tabulador, para que cualquier editor de texto reconozca correctamente la indentación. Utilizar líneas entre 75-80 caracteres de longitud, de esta manera se maneja mejor algunas herramientas y terminales.

4.3.3 Condicionales if, for, while, switch

Siempre tendrán espacios y se extenderán a múltiples líneas para mejorar la legibilidad. Ejemplos:

```
if ( Condición )  
{  
    // código  
}
```

```
while (Condición)  
{  
    // código  
}
```

```
for ( var i = 0; i < 100; i++)  
{  
    // código  
}
```

4.3.4 Comentarios

- Se deberá comentar en la línea que está encima del código del que se está refiriendo.
- Son aconsejables los comentarios en múltiples líneas (bloques de comentarios).
- No se deberán emplear los comentarios en el fin de línea.

El uso de comentarios en línea se utiliza para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o para ayudar a la memoria del programador. Se recomienda utilizar el estilo (*/* */*).

En el caso de los bloques de comentarios tienen solo un asterisco al inicio de cada línea exceptuando la primera. Los comentarios cortos pueden aparecer en una sola línea alineados al mismo nivel del código que lo sigue, además debe estar precedido de una línea en blanco.

```
/**  
 *Ejemplo de un comentario  
 *de múltiples líneas  
 */
```

```
/* Comentario*/
```

4.4 Interfaces de usuario

La interfaz de usuario es quizás la parte más importante de la aplicación, dado que es el componente que define la percepción del comportamiento con el usuario final. Una interfaz pobre dejará al usuario la impresión de que el programa es difícil de manejar, o que no realiza el trabajo para el cual se diseñó. Por otro lado, una buena interfaz puede hacer que un sistema sofisticado sea fácil de manejar, dejando la impresión al usuario de que es una aplicación potente, bien diseñada y de gran valor.

Siguiendo las reglas de una interfaz amigable, sencilla y agradable para el usuario, fueron diseñadas las interfaces del sistema **(Ver Anexo 5)**

4.5 Pruebas de software

Una vez desarrollado el *software*, es necesario realizar pruebas con el objetivo de minimizar la ocurrencia de errores y garantizar la calidad del producto. Para determinar si el sistema cumple con las expectativas del usuario final se deben realizar pruebas de *software*, definiendo niveles, tipos de pruebas, métodos y técnicas, para verificar que todos los requisitos fuesen implementados correctamente. Las descripciones de las pruebas realizadas se encuentran documentadas en el Plan de prueba del proyecto Multisaber-Navegante describiendo el propósito, alcance y visión de los tipos de pruebas identificados para el desarrollo de la colección.

4.5.1 Niveles y tipos de prueba

Entre los niveles generales de las pruebas se realizaron pruebas de integración y sistema. Los casos de prueba resultantes **(Ver Anexo 6)** de esta estrategia están dirigidos a evidenciar la integración de cada componente y el funcionamiento del sistema como un todo.

A la aplicación le fueron realizadas pruebas de funcionalidad (función y seguridad) y usabilidad que permitieron validar la implementación de funciones y métodos. Además permitieron verificar la integridad y disponibilidad de la información y el acceso a ella solo por parte de los profesores, así como garantizar la comprensión de los resultados por el usuario final.

4.5.2 Métodos de pruebas

4.5.2.1 Pruebas de caja blanca

Las pruebas de caja blanca también conocidas como pruebas de cristal, se aplican cuando se conoce el funcionamiento interno del producto, para asegurarse de que todas las piezas encajan. Es decir, que las operaciones internas se realizan de acuerdo con las especificaciones.

4.5.2.2 Pruebas de caja negra

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del *software*. Es decir, permiten al ingeniero de *software* derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa (22).

Las pruebas de caja negra tratan de encontrar errores en las siguientes categorías:

- Funciones incorrectas o faltantes
- Errores de interfaz
- Errores en estructuras de datos o en acceso a bases de datos externas
- Errores de comportamiento o desempeño
- Errores de inicialización y término

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan elementos de entrada y salidas esperadas para probar el sistema. El objetivo es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos (58).

4.5.3 Técnicas de pruebas

Para la realización de las pruebas de *software* existen varias técnicas entre las que se encuentran:

Partición equivalente

Es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases que permiten validar el correcto funcionamiento de las funcionalidades.

Análisis de valores límite: Es mayor el número de errores que se encuentran en los límites del dominio de entrada que en el “centro”, por ello se ha desarrollado el análisis de valores límite (AVL) como técnica de prueba. El análisis de valores límite es una técnica de diseño de casos de prueba que

complementa la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la selección de casos de prueba en las “aristas” de la clase (22).

La siguiente tabla relaciona con más detalles la estrategia diseñada:

Niveles de prueba	Tipos de prueba	Métodos	Técnicas
Integración	Funcionalidad (función)	Caja negra	Partición de equivalencia y (AVL)
Sistema	Funcionalidad (seguridad)	Caja negra	Partición de equivalencia
Sistema	Usabilidad (usabilidad)	Caja negra	Partición de equivalencia

Tabla 8: Estrategia de pruebas

A partir del desarrollo de los casos de prueba se realizaron tres iteraciones, guiados por las técnicas expuestas anteriormente, se identificaron un total de 18 no conformidades (NC), las cuales fueron resueltas garantizando así el correcto funcionamiento del sistema. A continuación se muestra una tabla que resume las iteraciones realizadas y los tipos de no conformidades encontradas.

Categorías	Iteración 1	Iteración 2	Iteración 3
Funciones incorrectas o faltantes	4	2	
Errores de interfaz	3	2	1
Errores en estructuras de datos	1		
Errores de comportamiento	3	1	
Errores de inicialización y término	1		

Tabla 9: No conformidades encontradas

Concluida cada iteración de pruebas se analizaron, por parte del equipo de desarrollo, las (NC) encontradas para determinar cuáles realmente constituyeron defectos del sistema según la descripción de los casos de uso. Las (NC) generadas han sido solucionados en su totalidad, por tanto no afectan el desarrollo de la aplicación, por lo que se encuentra lista para ser desplegada.

4.6 Conclusiones del capítulo

Los artefactos que fueron generados en este capítulo facilitaron la implementación de un sistema que cumpliera con todos los requerimientos definidos por el usuario, representados por los diagramas de componentes y el diagrama de despliegue. Las pruebas realizadas, guiadas por la elaboración de los

distintos casos de prueba utilizando pruebas de caja negra, permitieron detectar y erradicar errores presentes en la aplicación, mejorando así el funcionamiento del sistema.

Conclusiones generales

Con el estudio de soluciones similares existentes en el ámbito internacional y nacional, se demostró la importancia del uso de la minería de datos como parte de un proceso de extracción de conocimientos, así como su impacto significativo en el desarrollo de soluciones informáticas en apoyo a los procesos de enseñanza-aprendizaje.

Luego de realizar un estudio de las soluciones similares, se identificaron las tareas agrupamiento y extracción de reglas de asociación, apoyado en la herramienta Weka y guiado por la metodología Crisp-DM para el minado de los datos en el desarrollo de la solución.

La utilización de la metodología Crisp-DM para guiar el proceso de minado y RUP para guiar el proceso de desarrollo de *software* garantizaron el desarrollo exitoso de la aplicación.

La funcionalidad desarrollada posibilita a los profesores realizar de forma sencilla un proceso de extracción de conocimiento que les permite identificar patrones de comportamiento de los estudiantes, posibilitando una mejor estructuración de los contenidos, con la finalidad de perfeccionar el proceso de aprendizaje de los estudiantes y mejorar sus resultados docentes.

La solución propuesta fue validada funcionalmente utilizando pruebas de caja negra, basadas en los casos de pruebas desarrollados, lo que permitió erradicar deficiencias encontradas.

Recomendaciones

Al término de la presente investigación, las recomendaciones que puedan ser tomadas en cuenta para dar continuidad a este proyecto son las siguientes:

Se recomienda a los desarrolladores realizar un proceso de extracción de conocimiento con nuevos algoritmos presentes en la herramienta Weka, con la finalidad de comparar los resultados de estos con los escogidos en el desarrollo de la solución para mejorar el rendimiento de la solución.

Se recomienda a la dirección del proyecto incorporar nuevos criterios para la realización del proceso KDD, con el fin de enriquecer los reportes obtenidos e incorporar una ayuda para una mejor comprensión del proceso.

Bibliografía

1. **Rizzo, Ms.C César Labañino.** *Maestría Ciencias de la Educación.* La Habana : s.n., 2006.
2. **López, César Pérez y Santín González, Daniel.** *Minería de Datos Técnicas y Herramientas.* s.l. : Paraninfo, 2008.
3. **Baena, Domingo Savio Rodríguez.** *Análisis de datos de Expresión Genética mediante técnicas de Biclustering .* Sevilla, España : s.n., 2006.
4. **Piatetski-Shapiro.** *Knowledge discovery in databases: an overview.* California : s.n., 1991.
5. *Minería de Datos.* **Vallejos, Sofia J.** Corrientes, Argentina : s.n., 2007.
6. **Orallo, José Hernández, Ramirez Quintana, Maria José y Ferri Ramírez, César .** *Introducción a la Minería de Datos.* Madrid : Pearson, 2004.
7. **González, Enrique López y Mendaña Cuervo, Cristina.** *Nuevas tecnologías para el marketing público: minería de datos y técnicas bio-inspiradas.* España : s.n.
8. **Navarro, Miguel Montero.** *Extracción de conocimiento en bases de datos astronómicas.* Sevilla : s.n., junio, 2009.
9. **Cáceres, Lesley Méndez.** *Un algoritmo basado en la Programación Genética para la extracción de reglas de asociación.* Habana, Cuba : UCI.
10. **Ventura, Sebastian y Romero, Cristobal.** *Educational data mining: a survey from 1995 to 2005.* *Expert Systems with Applications.* 2007. 33:1.
11. *Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web.* **Morales, Cristobal Romero, Ventura Soto, Sebastián y Hervás Martínez, Cesar.** Córdoba : s.n., 2005. 84-9732-449-8.
12. **Galindo, Álvaro Jiménez y Álvarez García, Hugo.** *Minería de Datos en la Educación.* Madrid, España : s.n., 2010.
13. **Soto, Sebastián Ventura.** *Minería de Datos en sistemas educativos.* Córdoba : s.n., 2008.
14. **Saiz, Diego García.** *Minería de Datos aplicada a entornos educativos: estudio e implementación de técnicas libre de parámetros.* Cantabria, España : s.n., 2011.
15. **Miguillón, Julia.** *E-learning Personalization based on Itineraries and Long-term Navigational Behavior.* Barcelona : s.n., 2005.

16. **Dapozo, G y Porce, E.** *Aplicación de minería de datos con una herramienta de software libre en la evaluación del rendimiento académico de los alumnos de la carrera de Sistemas de la FACENA-UNNE.* Argentina : s.n., 2010.
17. **Díaz, Ernesto González, y otros, y otros.** *Obtención de patrones y reglas en el proceso académico de la Universidad de las Ciencias Informáticas utilizando técnicas de minería de datos.* Habana, Cuba : s.n., 2009.
18. **Acosta, Rolando y Vázquez, Luis.** *Obtención de patrones y reglas en el Sistema Docente del Instituto Superior Politécnico José A. Echevarría (CUJAE) utilizando Minería de Datos.* Habana, Cuba : s.n., junio, 2007.
19. **Sarasa, Raycos Brito.** *Minería de Datos aplicada a la Gestión Docente.* Habana, Cuba : s.n., 2008.
20. **Suárez, Alejandro Rosete, Brito Sarasa, Raycos y Acosta Sánchez, Rolando.** *Desarrollo de un proceso de KDD en el ámbito docente: Preparación de los datos.* La Habana : s.n., 2008.
21. **Sánchez, Ing. Rolando Acosta, Rosete Suárez, Dr. Alejandro y Rodríguez Díaz, Lic. Alfredo .** *Empleo de Minería de Datos en la predicción de diabetes. Preprocesado de datos.* 2009.
22. **Pressman, Roger S.** *Ingeniería del Software, un enfoque práctico.* 2010. 9786071503145.
23. **(IBM), International Business Machines Corp.** IBM - Rational Unified Process (RUP). [En línea] [Citado el: 10 de 1 de 2013.] <http://www-01.ibm.com/software/awdtools/rup/>.
24. **Waikato, Department of Computer Science The University of.** [En línea] [Citado el: 13 de 1 de 2013.] <http://www.cs.waikato.ac.nz/>.
25. Rapid-I. [En línea] [Citado el: 21 de 1 de 2013.] <http://rapid-i.com>.
26. Orange. [En línea] [Citado el: 21 de 1 de 2013.] <http://orange.biolab.si>.
27. knime. [En línea] [Citado el: 21 de 1 de 2013.] www.knime.org.
28. datknosys. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.datknosys.com/productos/knime/>.
29. **Unit, Informatics Research and Development.** *Open Source Data Mining Software Evaluation.* USA : s.n., 2010.
30. KDnuggets. [En línea] [Citado el: 25 de 3 de 2013.] <http://www.kdnuggets.com/polls/2011/tools-analytics-data-mining.html>.
31. **Gutiérrez, Javier J.** *¿Qué es un framework web?* 2010.

32. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.* España : s.n., 2008.
33. **Debrauwer, Laurent.** *UML 2 - Iniciación, ejemplos y ejercicios corregidos.* s.l. : ENI, 2009. 9782746047419.
34. Visual Paradigm for UML. [En línea] [Citado el: 12 de 1 de 2012.] <http://www.visual-paradigm.com/>.
35. **Cobo, Ángel, y otros, y otros.** *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web.* España : Díaz de Santos, 2005.
36. **Musciano, Chuck.** *HTML, la Guía Completa.* México : s.n., 1999.
37. **Lancker, Luc Van.** *XHTML y CSS - Los nuevos estándares del código fuente.* Barcelona : ENI, 2009.
38. **Pérez, Javier Eguíluz.** *Introducción al CSS.* España : s.n., 2008.
39. —. *Introducción al JavaScript.* España : s.n., 2008.
40. **Gutierrez, Emmanuel.** *Recursos Informáticos JavaScript - Conceptos básicos y avanzados.* Barcelona : ENI, 2009.
41. **Rodríguez, Pedro Muñoz.** *Mantenimiento de portales de información.* Madrid : Visión Libros, 2009.
42. **Groussard, Thierry.** *Recursos Informáticos Java Enterprise Edition - Desarrollo de aplicaciones web con JEE 6.* Barcelona : ENI, 2010. 9782746058484.
43. ABCdatos. [En línea] [Citado el: 12 de 1 de 2012.] <http://www.abcdatos.com/webmasters/programa/servidor-apache-windows.html>.
44. **Cobo, Angel.** *Diseño y programación de bases de datos.* Madrid, España : Vision Libros, 2007.
45. PostgreSQL. [En línea] [Citado el: 1 de 12 de 2012.] <http://www.postgresql.org>.
46. Netbeans. [En línea] [Citado el: 2 de 12 de 2012.] <http://netbeans.org/>.
47. **Weitzenfeld, Alfredo.** *Ingeniería de software orientada a objetos con UML, Java e Internet.* s.l. : Cengage Learning Editores, 2005. 9789706861900.
48. **Areba, Jesús Barranco de.** *Metodología Del Análisis Estructurado de Sistemas.* s.l. : Univ Pontifica Comillas, 2002. 9788484680437.
49. **F. Alonso Amo, Loïc A. Martínez Normand, Francisco Javier Segovia Pérez.** *Introducción a la Ingeniería del software.* s.l. : Delta Publicaciones, 2005. 9788496477001.
50. **Falgueras, Benet Campderrich.** *Ingeniería de software.* s.l. : Editorial UOC, 2003. 9788484297932.

51. **Torallbo, Msc Juan Alfonso Lara.** *Marco de descubrimiento de conocimiento para datos estructuralmente complejos.* Madrid, España : s.n., 2010.
52. **Pérez, Iván y León, Elizabet.** *Logica difusa para principiantes.* Caracas : s.n., 2007. 9789802445240.
53. **Han, Jiawei y Kambe, Micheline.** *The Morgan Kaufmann Series in Data Management Systems.* 2001.
54. **Jacobson, Ivan, Booch, Grady y Rumbaugh, Jim.** *UML Lenguaje unificado de modelado.* 2010.
55. **Mühlrad, Daniel.** *Patrones de diseño.* 2008.
56. *Sistema informático para la gestión de los planes de ingreso a la Educación Media y Superior en Cuba.* **Ortíz, Luis Ernesto Acosta y Sánchez Delgado, Daylenis .** Habana : s.n., 2012. 1994-1536.
57. **Coronel, Carlos.** *Sistemas de bases de datos: diseño, implementación y administración.* s.l. : Cengage Learning Editores, 2003. 9789706862860.
58. **Sommerville, Ian.** *Ingeniería del software 7/e.* Madrid : s.n., 2005.
59. **Molina López, José Manuel y García Herrero, Jesús.** *Técnicas de análisis de datos.* Madrid, España : s.n., 2004.
60. *Trabajo Investigativo sobre Minería de Datos.* **García, Yaima García.** Santiago de Cuba : s.n., 2011.
61. **Catalina, Enrique Quero.** *Sistemas operativos y lenguajes de programación .* Madrid, España : Paraninfo, 2003.
62. *Herramientas de Minería de Datos.* **Suárez, Yuniel Rodríguez y Díaz Amador1, Anolandy .** Habana, Cuba : s.n., 2009.
63. **Groussard, Thierry.** *JAVA 6 Los fundamentos del lenguaje java.* s.l. : ediciones ENI, 2009.
64. **Pisón, Francisco Javier Martinez del.** *Minería de datos en Series temporales Para la Búsqueda de conocimiento oculto en históricos de procesos industriales.* España : s.n., 2005. 26004.

Glosario de términos

Cluster: Palabra propia del inglés que significa grupo, racimo, trozo.

Estructura Lingüística: Constituye el necesario soporte material que hace posible el pasaje de lenguaje interior a el lenguaje exterior.

Hiperentorno: Datos que se encuentran almacenados en una red de nodos conectados por ligaduras, cuyos nodos pueden contener texto, gráficos, audio, video y cualquier otra forma de datos.

Hipermedia: Hipertexto que no se limita a documentos escritos, sino también otros medios como imágenes, audio y video.

Log: Registro oficial de eventos durante un rango de tiempo. Registra datos o información sobre quién, qué, cuándo y dónde se realizó una acción.

Multiplataforma: Programa o aplicación que puede utilizarse sin inconvenientes en distintas plataformas de *hardware* y sistemas operativos.

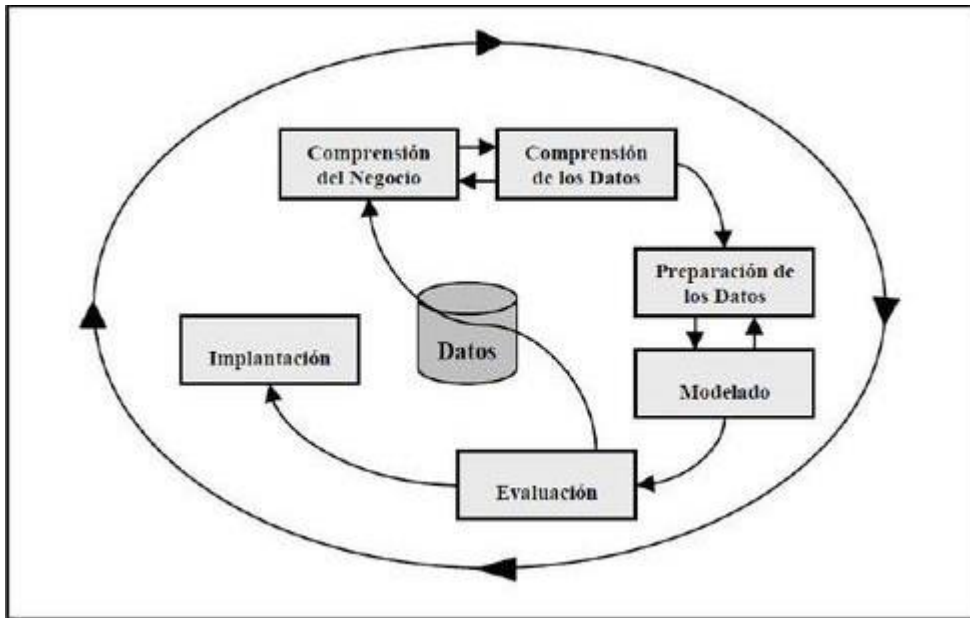
Script: Programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

Software educativo: Se refiere a programas informáticos creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza-aprendizaje.

Tipologías de software educativo: Con el desarrollo de la informática se fueron diversificando las formas de uso. Se definen algunas tipologías como tutoriales, *software* para la realización de ejercicios, adiestramiento, simulación, entre otras.

Anexos

Anexo 1: Ciclo de vida de un proyecto de minería de datos



Anexo 2: Descripción de los casos de uso del sistema

CU-1	Realizar proceso de extracción de conocimiento
Actor	Profesor
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción Resultados. El sistema le facilita al usuario la realización de un análisis de los registros de la interacción de los estudiantes con la aplicación utilizando minería de datos, permitiendo seleccionar los distintos tipos de algoritmos en función de los objetivos. Permitiendo además la especificación de parámetros de los distintos algoritmos por parte del usuario.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencia	RF1, RF2, RF3

Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 Selecciona la opción Resultados	
	<p>2 Brinda la posibilidad de seleccionar las distintas funcionalidades del módulo resultado.</p> <ul style="list-style-type: none"> • Trayectoria del estudiante • Análisis de contenidos • Historial del estudiante • Análisis integral • Análisis inteligente
3 Selecciona la opción Análisis inteligente	
	<p>4 Muestra los posibles algoritmos de minería de datos a utilizar en el proceso de extracción de conocimientos.</p> <ul style="list-style-type: none"> • Kmeans • DBScan Ver Sección 1: “Aplicar algoritmo DBScan” • Apriori Ver Sección 2: “Aplicar algoritmo Apriori” • PredictiveApriori Ver Sección 3: “Aplicar algoritmo PredictiveApriori”
5 Selecciona la opción Kmeans	
	<p>6 Muestra parámetros para la configuración del algoritmo a utilizar:</p> <ul style="list-style-type: none"> • Cantidad máxima de grupos a formar • Datos a procesar
7 Introduce la cantidad de grupos a formar y selecciona los datos a procesar. Selecciona la opción Aceptar.	

	8 Valida los datos
	9 Si los datos son correctos <ul style="list-style-type: none"> • Aplica el algoritmo Kmeans
	10 Termina el CU
Poscondiciones	Se aplicó un algoritmo de agrupamiento
Flujo alterno 7a	
7a Selecciona el botón Cancelar	
	8a Regresa a la pantalla de selección de algoritmos
Flujo alterno 9a	
	9.a Si los datos son incorrectos o los campos están vacíos <ul style="list-style-type: none"> • Muestra un mensaje de error.
Sección 1: “Aplicar algoritmo DBScan”	
1 Selecciona la opción DBScan	
	2 Muestra parámetros para la configuración del algoritmo a utilizar: <ul style="list-style-type: none"> • Datos a procesar • Cantidad mínima de elementos en un grupo
3 Introduce la cantidad de elementos y selecciona los datos a procesar. Selecciona la opción Aceptar.	
	4 Valida los datos
	5 Si los datos son correctos <ul style="list-style-type: none"> • Aplica el algoritmo DBScan
	6 Termina el CU
Poscondiciones	Se aplicó un algoritmo de agrupamiento
Flujo alterno 3a	
3a Selecciona el botón Cancelar	

	4a Regresa a la pantalla de selección de algoritmos
Flujo alterno 5a	
	5a Si los datos son incorrectos o los campos están vacíos <ul style="list-style-type: none"> • Muestra un mensaje de error.
Sección 2: “Aplicar algoritmo Apriori”	
1 Selecciona la opción Apriori	
	2 Muestra parámetros para la configuración del algoritmo a utilizar: <ul style="list-style-type: none"> • Cantidad máxima de reglas a formar • Datos a procesar
3 Introduce la cantidad máxima de reglas y selecciona los datos a procesar. Selecciona la opción de Aceptar.	
	4 Valida los datos
	5 Si los datos son correctos <ul style="list-style-type: none"> • Aplica el algoritmo Apriori
	6 Termina El CU
Poscondiciones	Se aplicó un algoritmo de reglas de asociación
Flujo alterno 3a	
3a Selecciona el botón Cancelar	
	4a Regresa a la pantalla de selección de algoritmos
Flujo alterno 5a	
	5a Si los datos son incorrectos o los campos están vacíos <ul style="list-style-type: none"> • Muestra un mensaje de error.
Sección 3: “Aplicar algoritmo PredictiveApriori”	

1 Selecciona la opción PredictiveApriori	
	2 Muestra parámetros para la configuración del algoritmo a utilizar: <ul style="list-style-type: none"> • Cantidad máxima de reglas a formar • Datos a procesar
3 Introduce la cantidad máxima de reglas y selecciona los datos a procesar. Selecciona la opción de Aceptar.	
	4 Valida los datos
	5 Si los datos son correctos <ul style="list-style-type: none"> • Aplica el algoritmo PredictiveApriori
	6 Termina el CU
Flujo alternativo 3a	
3a Selecciona el botón Cancelar	
	4a Regresa a la pantalla de selección de algoritmos
Flujo alternativo 5a	
	5a Si los datos son incorrectos o los campos están vacíos <ul style="list-style-type: none"> • Muestra un mensaje de error.
Poscondiciones	Se aplicó un algoritmo de reglas de asociación

Tabla 10: Descripción de CU 1 Realizar proceso de extracción de conocimiento

CU-2	Mostrar resultados Agrupamiento
Actor	Profesor
Resumen	El caso de uso inicia cuando el usuario aplica un algoritmo de agrupamiento. El sistema realiza una interpretación de los resultados obtenidos y los muestra garantizando una fácil comprensión.
Precondiciones	El usuario debe estar autenticado en el sistema y

	haber aplicado un algoritmo de agrupamiento.
Referencia	RF4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
sección 1 “Mostar resultados Kmeans”	
1. Se ejecuta el CU Realizar proceso de extracción de conocimiento.	
	2. Realiza una interpretación de los resultados de la aplicación del algoritmo de agrupamiento Kmeans. <ul style="list-style-type: none"> • Muestra un gráfico con los grupos formados
3. Selecciona uno de los grupos formados.	
	4. Muestra un nuevo gráfico con las características del grupo seleccionado.
	5. Termina el CU
sección 2 “Mostar resultados DBScan”	
1. Se ejecuta el CU Realizar proceso de extracción de conocimiento	
	2. Realiza una interpretación de los resultados de la aplicación del algoritmo de agrupamiento DBScan. <ul style="list-style-type: none"> • Muestra un gráfico con los grupos formados
3. Selecciona uno de los grupos formados.	
	4. Muestra los elementos que integran el grupo seleccionado por el usuario.
	5. Termina el CU.

Tabla 11: Descripción de CU 2 Mostrar resultados Agrupamiento

CU-3	Mostrar resultados Reglas
Actor	Profesor
Resumen	El caso de uso inicia cuando el usuario aplica un algoritmo de agrupamiento. El sistema realiza una interpretación de los resultados obtenidos y los muestra garantizando una fácil comprensión.
Precondiciones	El usuario debe estar autenticado en el sistema y haber aplicado un algoritmo de reglas de asociación.
Referencia	RF4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Se ejecuta el CU Realizar proceso de extracción de conocimiento	
	2. Realiza una interpretación de los resultados de la aplicación del algoritmo de reglas de asociación: <ul style="list-style-type: none"> • Muestra las reglas encontradas. • Brinda la opción de Generar reporte: Ver CU “Generar reporte”
	3. Termina el CU

Tabla 12: Descripción de CU 3 Mostrar resultados Reglas

CU-4	Generar reporte
Actor	Profesor
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción generar reporte, el sistema genera un reporte en formato PDF con las reglas encontradas y lo muestra al usuario.
Precondiciones	El usuario debe estar autenticado en el sistema y haber aplicado un algoritmo de reglas de asociación.

Referencia	RF5
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de generar reporte.	
	2. Realiza una interpretación de los resultados de la aplicación de los algoritmos de minería de datos y genera con los mismos un informe.
	3. Termina el CU.

Tabla 13: Descripción de CU 4 Generar reporte

Anexo 3: Diagrama de clases del análisis

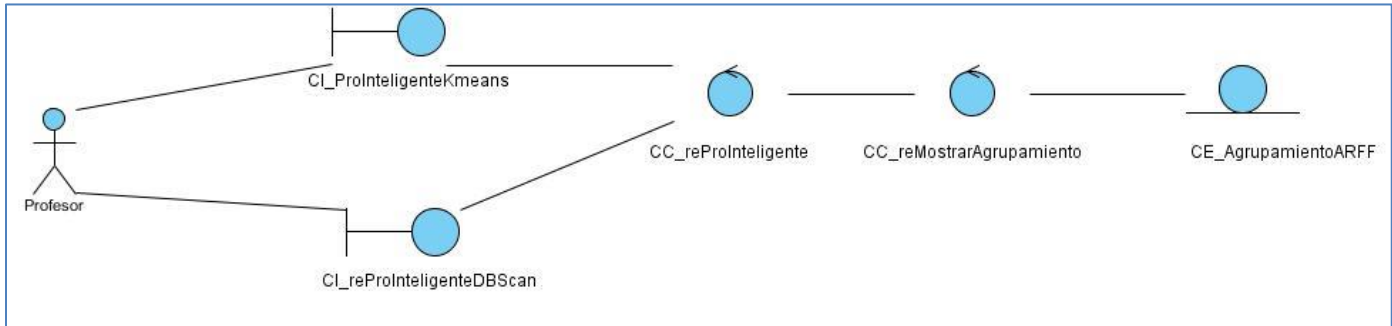


Ilustración 24: CU2: Mostrar resultados Agrupamiento

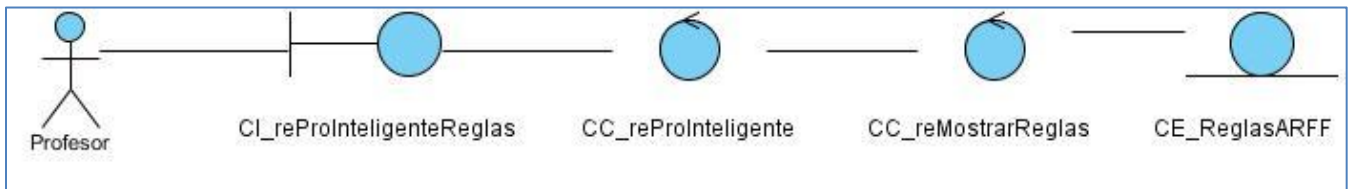


Ilustración 25: CU3: Mostrar resultados Reglas



Ilustración 26: CU4: Generar reporte

Anexo 4: Diagramas de colaboración:

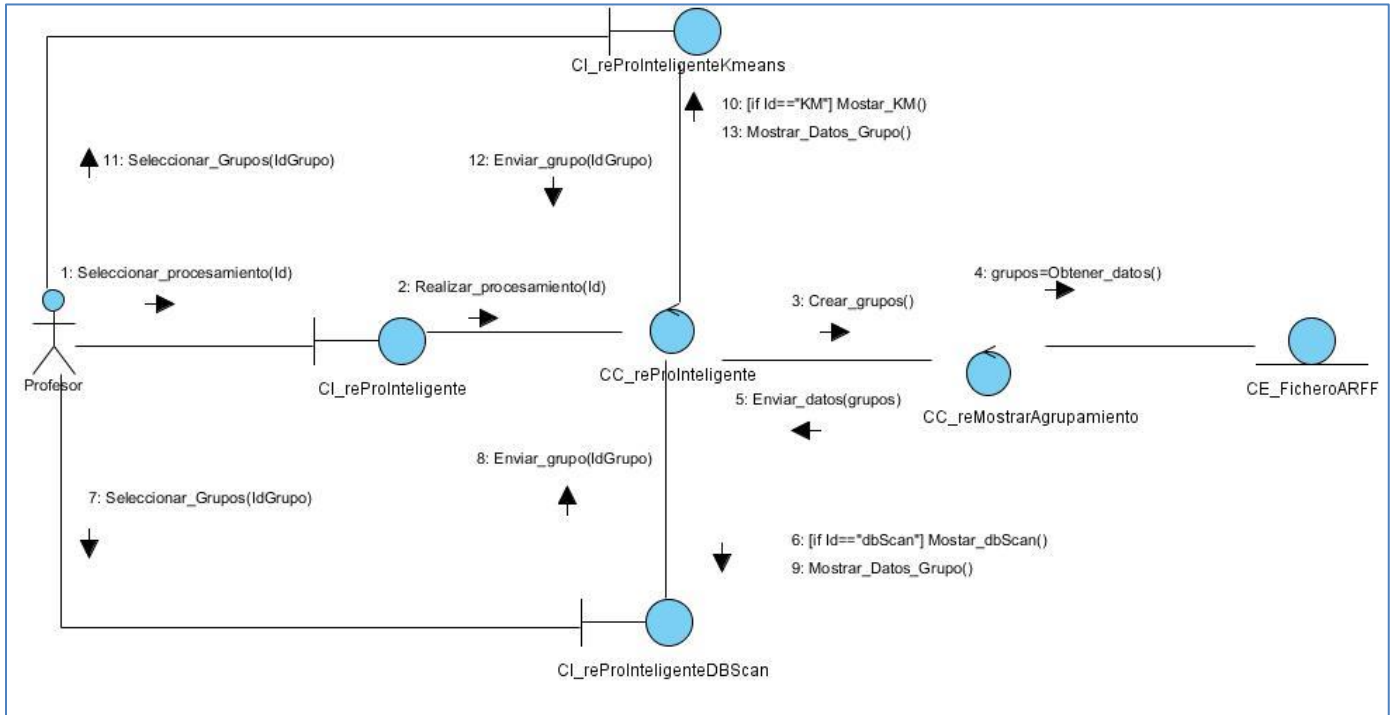


Ilustración 27: Diagrama de colaboración CU2: Mostrar resultados Agrupamiento

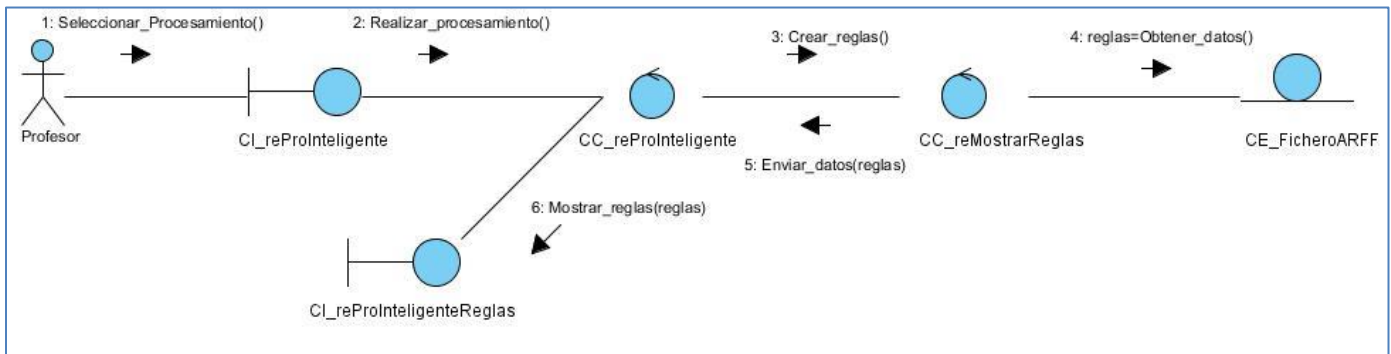


Ilustración 28: Diagrama de colaboración CU3: Mostrar resultados Reglas

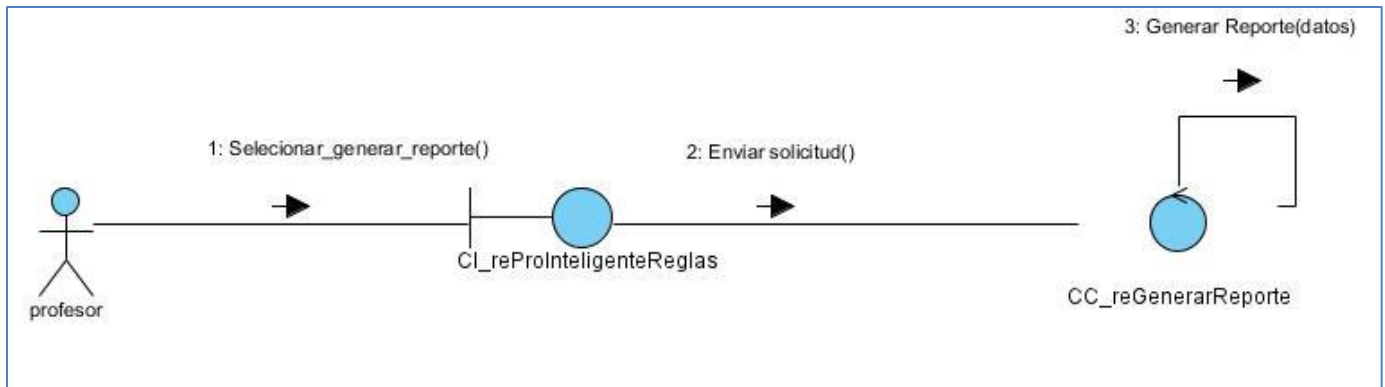


Ilustración 29: Diagrama de colaboración CU4: Generar reportes

Anexo 5: Interfaces de usuario

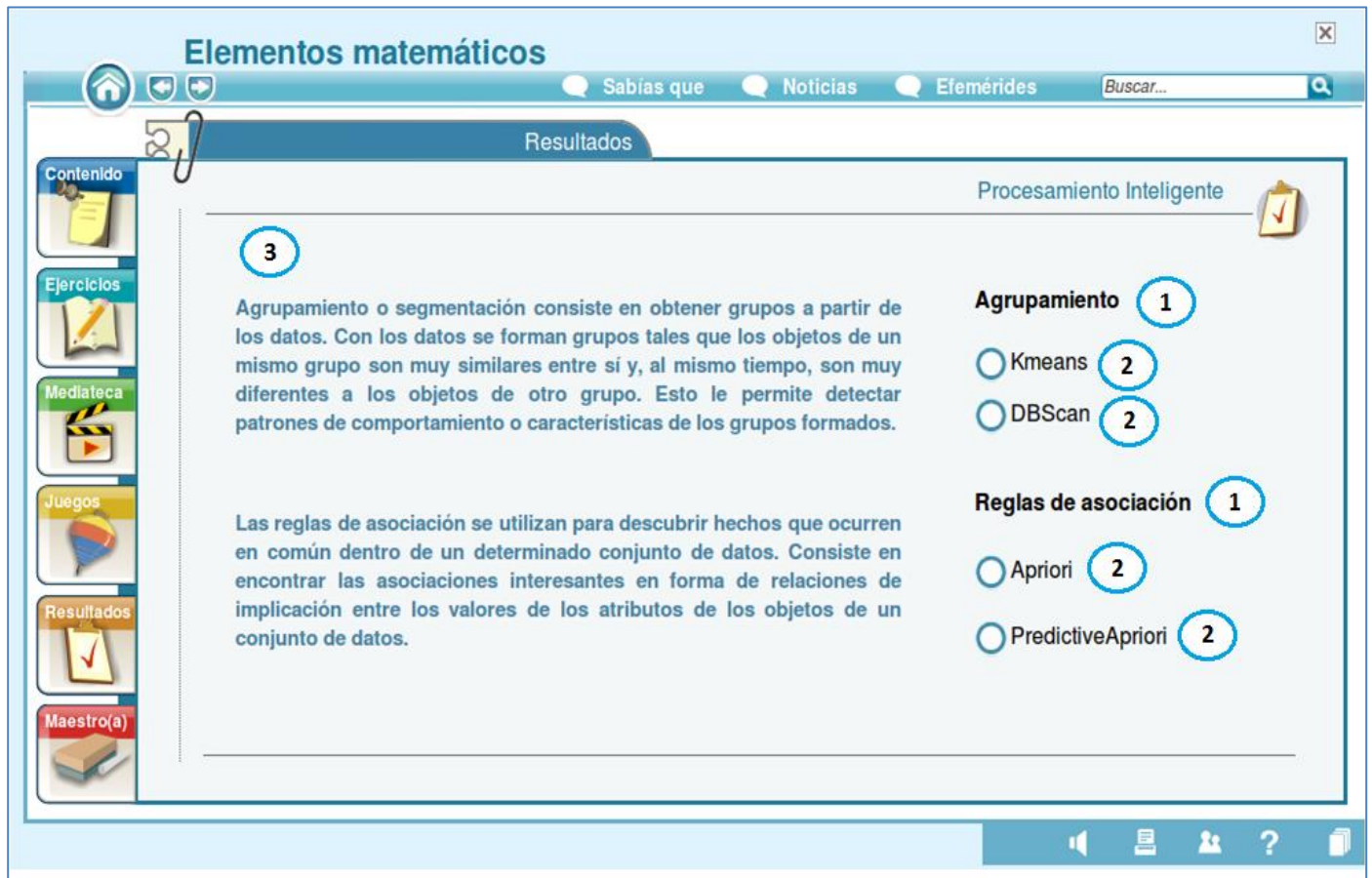


Ilustración 30: Interfaz de selección de los algoritmos

1. Posibles tareas a realizar en el proceso de extracción de conocimiento.
2. Posibles algoritmos a utilizar en las diferentes tareas.
3. Descripción de las tareas a realizar.



Ilustración 31: Interfaz del algoritmo Kmeans

1. Gráfica que muestra la cantidad de grupos formados.
2. Leyenda de los grupos formados.
3. Gráfica que muestra las características de cada uno de los grupos formados.
4. Leyenda de las características de los grupos formados.
5. Botón para mostrar la pantalla principal del procesamiento.

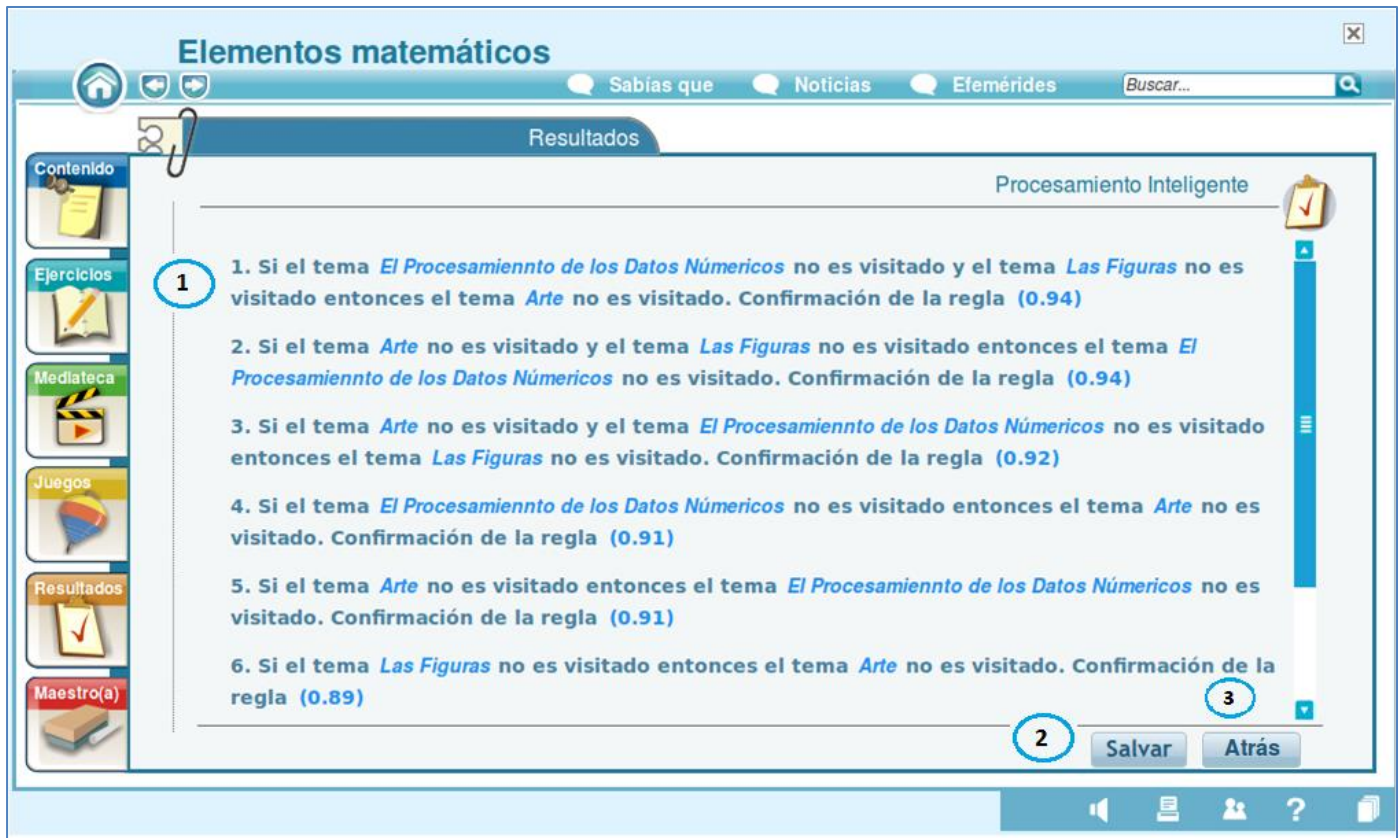


Ilustración 32: Interfaz del algoritmo Apriori

1. Reglas formadas como resultado de la aplicación del algoritmo.
2. Botón para generar un reporte en formato pdf de las reglas formadas.
3. Botón para mostrar la pantalla principal del procesamiento.

Anexo 6 Casos de prueba

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
V1	Datos	Lista desplegable	No	Secuencia de caracteres
V2	Cantidad	Campo de texto	No	Valor numérico entero mayor que cero.

Tabla 14: Descripción de variables

CU 1 Realizar proceso de extracción de conocimiento					
ID del Escenario	Escenario	Datos	Cantidad	Respuesta del sistema	Flujo central
SC 1	El caso de uso se inicia cuando el actor accede al módulo "Resultados".	NA	NA	Si el usuario autenticado en un profesor, el sistema muestra la opción de realizar procesamiento inteligente	Pantalla principal del Navegante / Resultados
SC 2	Selecciona la opción de "Procesamiento Inteligente"	NA	NA	El sistema muestra una descripción de las tareas a realizar y los posibles algoritmos a utilizar. Agrupamiento: <ul style="list-style-type: none"> • Kmeans • DBScan Reglas de asociación <ul style="list-style-type: none"> • Apriori • PredictiveApriori 	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente".
SC 3.1	Selecciona la opción "Kmeans"	NA	NA	Brinda la opción de definir la cantidad máxima de grupos a formar y seleccionar que datos se van a procesar.	Pantalla principal del Navegante / Clic en la imagen del

Anexos

				Cantidad de grupos: <ul style="list-style-type: none"> • Cantidad Datos a procesar: <ul style="list-style-type: none"> • Temas • Itinerario 	módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Kmeans".
SC 3.1	Introduce los datos y selecciona el botón Aceptar	V	V	Valida los datos y aplica el algoritmo Kmeans.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Kmeans" / clic en Aceptar
SC 3.1.1	Introduce datos incorrectos o incompletos y selecciona la opción Aceptar	V	I	Muestra el mensaje de error: "La cantidad de grupos es incorrecta"	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Kmeans" / clic en Aceptar
SC 3.1.2	Selecciona la opción Cancelar	NA	NA	Regresa a la pantalla de selección de algoritmos	Pantalla principal del Navegante /

Anexos

					Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Kmeans" / clic en Cancelar
--	--	--	--	--	---

Tabla 15: Caso de prueba CU1 Realizar proceso de extracción de conocimiento

CU 1 Realizar proceso de extracción de conocimiento					
ID del Escenario	Escenario	Datos	Cantidad	Respuesta del sistema	Flujo central
SC 3.2	Selecciona la opción "DBScan"	NA	NA	Brinda la opción de definir la cantidad mínima de elementos por grupos y seleccionar que datos se van a procesar. <ul style="list-style-type: none"> • Cantidad de elementos: • Cantidad Datos a procesar: <ul style="list-style-type: none"> • Temas • Itinerario 	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "DBScan".
SC 3.2	Introduce los datos y selecciona el botón Aceptar	V	V	Valida los datos y aplica el algoritmo DBScan.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en

					“Procesamiento Inteligente” / clic en “DBScan” / clic en Aceptar
SC 3.2.1	Introduce datos incorrectos o incompletos y selecciona la opción Aceptar	V	I	Muestra el mensaje de error: “La cantidad mínima de elementos para formar un grupo es incorrecta”	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “DBScan” / clic en Aceptar
SC 3.2.2	Selecciona la opción Cancelar	NA	NA	Regresa a la pantalla de selección de algoritmos	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “DBScan” / clic en Cancelar

Tabla 16: Caso de prueba CU1 Realizar proceso de extracción de conocimiento

CU 1 Realizar proceso de extracción de conocimiento					
ID del Escenario	Escenario	Datos	Cantidad	Respuesta del sistema	Flujo central

Anexos

SC 3.3	Selecciona la opción "Apriori"	NA	NA	Brinda la opción de definir la cantidad máxima de reglas a mostrar y seleccionar que datos se van a procesar. <ul style="list-style-type: none"> • Cantidad de reglas: • Cantidad Datos a procesar: <ul style="list-style-type: none"> • Temas • Itinerario 	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Apriori".
SC 3.3	Introduce los datos y selecciona el botón Aceptar	V	V	Valida los datos y aplica el algoritmo Apriori.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Apriori" / clic en Aceptar
SC 3.3.1	Introduce datos incorrectos o incompletos y selecciona la opción Aceptar	V	I	Muestra el mensaje de error: "La cantidad de reglas es incorrecta"	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Apriori" / clic en Aceptar
SC 3.3.2	Selecciona la opción	NA	NA	Regresa a la pantalla de	Pantalla principal

	Cancelar			selección de algoritmos	del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “Apriori” / clic en Cancelar
--	----------	--	--	-------------------------	--

Tabla 17: Caso de prueba CU1 Realizar proceso de extracción de conocimiento

CU 1 Realizar proceso de extracción de conocimiento					
ID del Escenario	Escenario	Datos	Cantidad	Respuesta del sistema	Flujo central
SC 3.4	Selecciona la opción “PredictiveApriori”	NA	NA	Brinda la opción de definir la cantidad máxima de reglas a mostrar y seleccionar que datos se van a procesar. Cantidad de reglas: <ul style="list-style-type: none"> • Cantidad Datos a procesar: <ul style="list-style-type: none"> • Temas • Itinerario 	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “PredictiveApriori”.
SC 3.4	Introduce los datos y selecciona el botón Aceptar	V	V	Valida los datos y aplica el algoritmo PredictiveApriori.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic

					en “Procesamiento Inteligente” / clic en “PredictiveApriori” / clic en Aceptar
SC 3.4.1	Introduce datos incorrectos o incompletos y selecciona la opción Aceptar	V	I	Muestra el mensaje de error: “La cantidad de reglas es incorrecta”	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “PredictiveApriori” / clic en Aceptar
SC 3.4.2	Selecciona la opción Cancelar	NA	NA	Regresa a la pantalla de selección de algoritmos	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “PredictiveApriori” / clic en Cancelar

Tabla 18: Caso de prueba CU1 Realizar proceso de extracción de conocimiento

CU 2 Mostrar resultados Agrupamiento			
ID del Escenario	Escenario	Respuesta del sistema	Flujo central
SC1	Una vez ejecutado el algoritmo Kmeans.	Muestra un gráfico de pastel con los grupos formados.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "Kmeans" / clic en aceptar.
SC1	El usuario selecciona uno de los grupos (particiones) del gráfico de pastel.	Muestra un gráfico de barras con el comportamiento de cada uno de los atributos procesados dentro del grupo.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados / Clic Procesamiento Inteligente / clic en "Kmeans" / clic en aceptar / clic en gráfico.
CU 2 Mostrar resultados Agrupamiento			
ID del Escenario	Escenario	Respuesta del sistema	Flujo central
SC2	Una vez ejecutado el algoritmo DBScan.	Muestra un gráfico de pastel con los grupos formados.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en "Procesamiento Inteligente" / clic en "DBScan" / clic en aceptar.
SC2	El usuario selecciona uno de los grupos	Muestra un listado de los estudiantes relacionados con el grupo seleccionado.	Pantalla principal del Navegante / Clic en la imagen del módulo

	(particiones) del gráfico de pastel.		Resultados/Clic en “Procesamiento Inteligente” / clic en “DBScan” / clic en aceptar / clic en gráfico.
SC2	El usuario selecciona uno de los estudiantes del listado.	Muestra el comportamiento del estudiante seleccionado en una sesión de trabajo.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “DBScan” / clic en aceptar / clic en porción / clic en estudiante.

Tabla 19: Caso de prueba CU 2 Mostrar resultados Agrupamiento

CU 3 Mostrar resultados Reglas			
ID del Escenario	Escenario	Respuesta del sistema	Flujo central
SC3	Una vez ejecutado el algoritmo Apriori.	Muestra un listado de las reglas de asociación encontradas.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en “Apriori” / clic en aceptar.
SC4	Una vez ejecutado el algoritmo PredictiveApriori.	Muestra un listado de las reglas de asociación encontradas.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento

			Inteligente” / clic en “PredictiveApriori” / clic en aceptar.
--	--	--	---

Tabla 20: Caso de prueba CU 3 Mostrar resultados Reglas

CU 3 Generar reporte			
ID del Escenario	Escenario	Respuesta del sistema	Flujo central
SC1	Exportar las reglas de asociación encontradas en la aplicación de los algoritmos Apriori y/o PredictiveApriori.	Genera un documento en formato pdf, con las reglas encontradas en la aplicación de los algoritmos.	Pantalla principal del Navegante / Clic en la imagen del módulo Resultados/Clic en “Procesamiento Inteligente” / clic en Apriori o PredictiveApriori / clic en aceptar / Clic en Salvar.

Tabla 21: Caso de prueba CU 4 Generar reporte