Universidad de las Ciencias Informáticas

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

Aplicación de la minería de datos para determinar reglas sobre estudiantes con problemas docentes

Autor: Itamys Arelis Hodelin Valiente

David Mourlot Matos

Tutor: Ing. Yeinier Ferrás Cecilio

Ciudad de la Habana, Cuba

Junio, 2013

“Año 55 de la Revolución”

"Instrucción no es lo mismo que educación: aquella se refiere al pensamiento, y esta principalmente a los sentimientos. Sin embargo, no hay buena educación sin instrucción. Las cualidades morales suben de precio cuando están realzadas por las cualidades inteligentes".

Datos de contacto

Ing. Yeinier Ferrás Cecilio.

Graduado de ingeniero en Ciencias Informáticas en el año 2007, profesor instructor. Se ha desempeñado como jefe de módulo de Anatomía Patológica del Departamento de Gestión Hospitalaria durante 3 años y medio, y participó en el desarrollo del mencionado módulo que pertenece al sistema alas-HIS. Ha impartido las asignaturas de pregrado: Física, Práctica Profesional y Arquitectura de Computadoras; y también impartió el posgrado: Técnicas avanzadas de desarrollo de aplicaciones empresariales en Java. Correo electrónico: **yferras@uci.cu**

Agradecimientos

Dedicatoria

Resumen

La Minería de Datos (MD) se ha convertido en una herramienta muy poderosa debido a su utilidad en la extracción de conocimiento de grandes volúmenes de datos. Sus técnicas y métodos han sido aplicados en múltiples áreas, incluyendo su uso en la educación superior para la comprensión, mejora y adaptación de los sistemas educacionales.

La presente investigación describe todo el proceso realizado para la obtención de un modelo que permita predecir dada su trayectoria actual, aquellos estudiantes con alta probabilidad de presentar problemas académicos con una asignatura en particular, apoyándose en las técnicas y algoritmos de la Minería de Datos Educacional. Para validar la aplicabilidad de este modelo al contexto de la Universidad de las Ciencias Informáticas, se implementó un sistema de alerta temprana. En la implementación de dicho sistema se utilizó el lenguaje de programación Java, la herramienta de desarrollo NetBeans y la librería Weka para el análisis de los datos, todo el proceso guiado por la metodología XP. Con el estudio de esta investigación, los profesores de la Universidad de las Ciencias Informáticas contarán con una nueva alternativa para predecir automáticamente estudiantes con problemas docentes.

Palabras clave: Minería de Datos, Minería de Datos Educacional, predecir, algoritmos, Weka, XP.

Índice de contenido

Índice de contenido

Índice de contenido

Índice de Figuras

Índice de Tablas

Índice de Tablas

Introducción

En las últimas décadas se ha experimentado un incremento en la disponibilidad e intercambio de grandes volúmenes de datos a través de Internet. El continuo crecimiento de la web global permitió que cada vez más negocios y organizaciones, recolectaran a gran escala los datos acerca de sus operaciones y oportunidades de mercado. Más allá del propósito inmediato de recoger, explicar y archivar las actividades de una organización, estos datos pueden, en ocasiones, constituir una verdadera mina de oro para el planeamiento estratégico (1).

En un esfuerzo por satisfacer las crecientes necesidades en el manejo de la información, los investigadores exploraron ideas y métodos desarrollados en los campos de la inteligencia artificial, el análisis estadístico de datos, la visualización de datos, el diseño de bases de datos, entre otros. Estos esfuerzos derivaron en el surgimiento de un área de investigación generalmente conocida como **minería de datos y descubrimiento de conocimiento** [en bases de datos] (1). En el presente trabajo se le llamará Minería de Datos (MD).

Las técnicas de la MD han sido aplicadas con éxito en muchas áreas, desde los negocios hasta la ciencia y los deportes. Se han utilizado en la astronomía, la biología molecular, la medicina, la detección de fraudes de hacienda, el monitoreo de lavado de dinero, entre otros campos.

La educación y sobre todo la educación superior, es otro de los terrenos donde las técnicas y métodos de la MD han sido aplicados buscando mejoras en la comprensión y adaptación de los sistemas educacionales. Los aportes realizados y el nuevo cuerpo de conocimientos generado, han dado origen a un área de investigación y desarrollo denominado Minería de Datos Educacional (del inglés, Educational Data Mining) o Analítica del Aprendizaje (del inglés, Learning Analitycs).

La tendencia de las universidades a incorporar el uso de la tecnología en sus distintos procesos, ha aumentado en los últimos años. Evidencia de ello es la implantación de varios sistemas computacionales como son: los Sistemas de Información del Estudiante (SIE), diseñados para colectar datos socio-demográficos de cada uno de los ingresados a una universidad; los Sistemas de Gestión Académica (SGA), encargados del manejo de las notas y trayectorias de los estudiantes en los

diferentes cursos; y los Sistemas de Gestión del Aprendizaje (LMS, por sus siglas en inglés) utilizados para la creación de entornos virtuales que sirvan de apoyo, o alternativa, a los procesos tradicionales de enseñanza.

El Plan Tecnológico para la Educación Nacional, del Departamento de Educación de los Estados Unidos de América (NETP, U.S. Department of Education) asegura que:

"[...] existen muchas oportunidades de explotar el poder de la tecnología para la evaluación formativa. La misma tecnología que da soporte a las actividades de aprendizaje acumula datos... que pueden ser utilizados para su evaluación..." (2).

Dicho departamento tiene la visión de un aprendizaje personalizado y de un sistema de retroalimentación interconectado, el cual serviría para asegurar que las decisiones vitales acerca del proceso de enseñanza y aprendizaje estén informadas por los datos; y que los datos sean agregados y accesibles a todo el sistema educacional, para su mejoramiento continuo (2).

El Reporte Horizontes, prestigiosa publicación que identifica tecnologías emergentes con alta probabilidad de impactar el sector de la educación en el lapso de un quinquenio, ha listado la Analítica del Aprendizaje como uno de los impactos que deben revolucionar la educación superior en los próximos dos o tres años. Su predicción se basa en una serie de iniciativas que están impulsando la utilización y desarrollo de esta tecnología. El reporte describe así la importancia y la potencialidad de la Analítica del Aprendizaje:

"La mayor promesa de la Analítica del Aprendizaje... consiste en que, cuando sea correctamente aplicada e interpretada, permitirá a la facultad entender con mayor precisión las necesidades de aprendizaje de los estudiantes y ajustar la instrucción... de una forma mucho más acertada y rápida de la que hoy es posible. Esto tiene implicaciones no solo en el desempeño individual del estudiante, sino en la forma en la que los educadores perciben los procesos de enseñanza, aprendizaje y evaluación" (3).

A pesar de todo el crédito y el apoyo alcanzado por la Minería de Datos Educacional (MDE) a nivel mundial, en Cuba los estudios o experimentos realizados sobre el tema son escasos. Una de las posibles explicaciones para esta situación, es la desconfianza que muestran algunos educadores y

administrativos hacia la toma de decisiones académicas basándose en los datos. Ésta es, en cierto modo, una reacción natural. Sin embargo, en el contexto actual de la educación, los educadores deben desarrollar una cultura de utilización de los datos en la toma de decisiones relativas a la instrucción. Necesitan la experiencia de tener datos del estudiante que les digan algo útil y aplicable acerca de la enseñanza y el aprendizaje (4).

La Universidad de las Ciencias Informáticas (UCI) cuenta con variantes específicas de los sistemas antes mencionados (SIE, SGA). En cada uno de ellos se ingresa y almacena información relativa a los estudiantes (nombre, apellidos, provincia, municipio, centro de procedencia, asistencia, evaluaciones parciales, cortes evaluativos, evaluaciones finales, entre otros). Cuenta además con un Entorno Virtual de Aprendizaje (EVA), en el cual los estudiantes realizan tareas y actividades orientadas por el profesor; el sistema guarda todas las acciones ejecutadas. Los Registros de Asistencia y Evaluaciones Frecuentes son otra valiosa fuente de información, pues contienen detalles de lo sucedido en el aula durante cada uno de los encuentros.

La disponibilidad de datos y el potencial de su utilización en la Universidad son grandes. No obstante, todo el cúmulo de información que se almacena actualmente solo se utiliza para mantener estadísticas actualizadas y para la generación de reportes.

Estas estadísticas, aunque importantes, no son de utilidad para guiar a los profesores menos avezados en el aspecto pedagógico de la instrucción. No reflejan ni los hábitos, ni los estilos de aprendizaje de los estudiantes; mucho menos dicen cuales alumnos necesitan una mayor atención. De igual modo, para los profesores encargados de diseñar las estrategias de enseñanza en la Universidad, estas estadísticas no proveen toda la retroalimentación necesaria para evaluar con precisión, y a tiempo, la efectividad de las decisiones tomadas.

Esto evidencia un pobre aprovechamiento de los datos generados por los procesos de administración y de enseñanza; así como también la carencia de investigaciones que aborden el uso de los mismos desde otras perspectivas y para otros fines, como por ejemplo: **predecir automáticamente estudiantes con problemas docentes.**

Introducción

Luego de analizar la situación antes expuesta queda definido el siguiente **problema a resolver:** ¿Cómo identificar estudiantes con alta probabilidad de desaprobar una asignatura en la Universidad de las Ciencias Informáticas?

Se definió como **objeto de estudio**: Las técnicas y métodos de la Minería de Datos; a raíz de lo cual el **campo de acción** se enmarca en las técnicas de predicción utilizadas en la Minería de Datos Educacional.

Para dar solución al problema existente se ha definido como **objetivo general de la investigación:** Obtener un modelo predictivo capaz de identificar aquellos estudiantes con alta probabilidad de desaprobar el examen final de una asignatura.

Para dar cumplimiento al objetivo general, se plantean las siguientes **tareas de investigación:**

1. Valorar las fuentes de datos que existen hoy en la Universidad y que almacenan diversa información de los estudiantes.
2. Realizar un estudio de las técnicas de la minería de datos educacional que se utilizan en los problemas de predicción.
3. Seleccionar los algoritmos que mejor se ajusten al dominio del problema.
4. Seleccionar el modelo predictivo que proporcione los mejores resultados.
5. Implementar un sistema de alerta temprana, basado en dicho modelo predictivo.
6. Validar la aplicabilidad del modelo predictivo al entorno de la Universidad.

Como **resultados esperados** una vez concluidas las tareas de la investigación se plantea la propuesta de un modelo predictivo para la identificación de estudiantes en riesgo de desaprobar el examen final de una asignatura, junto con un sistema de alerta temprana que implemente dicho modelo.

El contenido del presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1: Fundamentos teóricos. Se hace referencia a los elementos teóricos que constituyen la base de la investigación realizada. Se exponen los resultados del estudio del estado del arte y se describen las soluciones similares existentes. Se presentan las herramientas, y metodología a utilizar fundamentándose su selección en el análisis desarrollado.

Capítulo 2: Características del sistema. Se abordan temas relacionados con el proceso de Minería de Datos para obtener el modelo predictivo más adecuado, así como las características esenciales, funcionalidades y potencialidades del sistema de alerta que implementa este modelo. Además se desarrollan las fases de exploración y planificación definidas por la metodología XP, presentándose los requisitos no funcionales del sistema.

Capítulo 3. Diseño e implementación del sistema. En este capítulo se desarrollan las fases de diseño e implementación definidas por la metodología XP, con las cuales se finaliza todo el proceso de desarrollo del sistema, obteniéndose de esta manera la solución a la problemática planteada en la investigación.

Capítulo 1: Fundamentación teórica de la investigación

Fundamentación teórica de la investigación

Introducción

Este capítulo contiene la base teórica del presente trabajo. En él se resumen los principales conceptos relacionados con el objeto de estudio y campo de acción del problema. Se incluyen además las últimas tendencias en la aplicación de la Minería de Datos Educacional (MDE) para la predicción de resultados académicos, así como un análisis de las herramientas y técnicas que se han de utilizar para el desarrollo de la aplicación.

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

1.1 Conceptos importantes

A continuación se muestran algunos conceptos básicos relacionados con el dominio del problema para lograr una mejor comprensión del mismo.

1.1.1 **Inteligencia Artificial (IA)**

No existe una definición única y universalmente aceptada de la IA, mencionan Russel y Norving (5). A pesar de que la mayoría de los intentos para definir términos complejos como es el caso de la IA suelen ser inútiles, es positivo al menos trazar los límites aproximados en los que incluir el concepto de IA para poder proporcionar una perspectiva sobre la explicación que sigue a continuación. Para lograrlo se ha propuesto la siguiente definición, a pesar de no ser aceptada universalmente.

La Inteligencia Artificial es la parte de las ciencias de la computación que intenta aplicar rasgos del pensamiento y el comportamiento humanos a la solución automatizada de problemas. Para ello, ha tomado ideas de otras disciplinas tales como la Filosofía, las Matemáticas, la Neurociencia, la Psicología, la Ingeniería de Sistemas, la Cibernética y la Lingüística (5).

1.1.2 **Minería de Datos (MD)**

A partir del análisis de la literatura consultada se puede definir la MD de la siguiente forma:

La MD es el proceso que se enfoca en la búsqueda de información nueva, valiosa y no trivial dentro de grandes volúmenes de datos (6). Para ello utiliza métodos desarrollados en los campos de aprendizaje automatizado (del inglés, machine learning), reconocimiento de patrones, análisis estadístico de datos, visualización de datos, redes neuronales, entre otros (7).

Principales tareas de la MD

Aunque en la práctica, las dos metas principales de la MD tienden a ser la predicción y la descripción (8), la MD comprende también otras tareas dependiendo de las necesidades de la persona que analiza los datos. David Hand (9) clasifica dichas tareas del modo siguiente:

1. **Análisis exploratorio de los datos:**

El objetivo de esta tarea es explorar los datos sin ninguna idea clara de lo que se busca. Es decir, se examinan los datos en busca de estructuras que puedan indicar relaciones más profundas entre los casos o las variables. El análisis exploratorio puede describirse como la generación de hipótesis basadas en los datos. (10)

2. **Modelado descriptivo:**

El modelado descriptivo consiste en describir todos los datos (o los procesos que los generan), ejemplos de tales descripciones incluyen modelos de la distribución general de probabilidad en los datos (estimación de densidad), el particionamiento del espacio p-dimensional en grupos (agrupamiento o segmentación) y los modelos que describen la relación entre las variables (modelado de dependencias) (10)

3. **Modelado predictivo. Clasificación y Regresión:**

El objetivo de esta tarea es construir un modelo capaz de predecir el valor de una variable a partir de los valores conocidos de las demás variables. Por ejemplo, si se quiere determinar cuál caballo ganará una carrera o el grado de fragilidad de una soldadura. La diferencia clave entre la predicción y la descripción, es que la predicción tiene como objeto una sola variable; mientras que en los problemas descriptivos ninguna de las variables es primordial para el modelo. (10)

4. **Descubrimiento de patrones y reglas:**

Los tres tipos de tareas anteriores están relacionados con la construcción de modelos. Otras aplicaciones de la MD están enfocadas en la detección de patrones. Este problema [el de la detección de patrones] ha sido objeto de mucha atención en la MD y ha sido enfrentado utilizando técnicas algorítmicas basadas en reglas de asociación. (10)

5. **Recuperación de acuerdo al contenido:**

En esta tarea el usuario tiene un patrón de interés y desea encontrar patrones similares en el conjunto de datos. Esta tarea es frecuentemente utilizada para bases de datos que contienen textos o imágenes. (10)

Técnicas más utilizadas de la Minería de Datos

- ✓ **Agrupamiento (del inglés, Clustering)**

El proceso de agrupamiento es una técnica para el modelado descriptivo consiste en la división de los datos en grupos de objetos similares. El representar los datos en una serie de clusters implica la pérdida de detalles, pero consigue la simplificación de los mismos. El agrupamiento es una técnica más del aprendizaje automatizado en la que el aprendizaje es no supervisado (no se conocen a priori las etiquetas de las clases) (11) .

Desde un punto de vista práctico, el clustering juega un papel muy importante en aplicaciones de MD, tales como exploración de datos científicos, recuperación de la información, minería de texto, aplicaciones sobre bases de datos espaciales (datos procedentes de astronomía), aplicaciones web, marketing, diagnóstico médico, análisis de ADN en biología computacional y muchas otras (11).

- ✓ **Clasificación**

La clasificación tiene como idea colocar un objeto dentro de una clase o categoría, basándose para ello en sus otras características. La etiqueta de clase es un valor (simbólico) discreto y es conocido para cada objeto. Esto se conoce como aprendizaje supervisado (11).

Construir modelos de clasificación (a veces llamados clasificadores), permite que asignen la etiqueta de clase correcta a objetos aun no vistos y que carezcan de ellas. Los modelos de clasificación son utilizados sobre todo para el modelado predictivo (11).

- ✓ **Obtención de reglas de asociación**

La obtención de reglas de asociación es una de las técnicas principales de la minería de datos. Está relacionada al descubrimiento de patrones en los datos. Es decir, descubre relaciones entre los atributos de una base de datos; produciendo sentencias de tipo: si-entonces, relativas a los atributos y sus valores (11).

Las reglas de asociación no son diferentes de las reglas de clasificación, pero pueden predecir cualquier atributo, no sólo la clase y esto les brinda la libertad de predecir además, combinaciones de atributos (11).

1.1.3 **Minería de datos y educación**

Los esfuerzos de los especialistas para entender y mejorar los sistemas de enseñanza, unidos a la creciente asimilación de la tecnología en los centros de educación, abrieron el camino a una nueva y extensa área de investigaciones. Pronto los investigadores notaron que las herramientas computacionales y los sistemas de gestión en las universidades, por la gran cantidad de datos que generaban y almacenaban, constituían valiosas fuentes de información (12).

La toma de decisiones en los salones de clases incluye la observación del comportamiento del estudiante, analizar sus datos históricos y estimar la efectividad de las estrategias pedagógicas. Sin embargo, cuando los estudiantes trabajan en ambientes electrónicos, no es posible llevar a cabo este monitoreo informal; los educadores deben buscar otras formas de obtener esta información.

Minería de Datos Educacional (MDE)

La Minería de Datos Educacional desarrolla métodos y aplica técnicas tomadas de las estadísticas, el aprendizaje automatizado y la minería de datos; con el fin de analizar los datos recogidos durante los procesos de enseñanza y aprendizaje. La MDE pone a prueba las teorías de aprendizaje e informa la práctica educacional (13).

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

1. Predecir el comportamiento futuro de los estudiantes mediante la creación de modelos que incorporen información detallada, por ejemplo: el conocimiento, la motivación, la metacognición y las actitudes de los estudiantes.
2. Descubrir o mejorar los modelos de dominio que caractericen el contenido que ha de aprenderse y las secuencias de instrucción óptimas.
3. Estudiar los efectos de los distintos tipos de apoyo pedagógico que pueden proveer los softwares de aprendizaje.
4. Avanzar el conocimiento científico acerca del aprendizaje y los aprendices, construyendo modelos computacionales que incorporen modelos del estudiante, del dominio y de la pedagogía del software (13).

El reto de la clasificación en la MDE

Como se mencionó anteriormente, la clasificación es una de las técnicas más utilizadas en la MD y su uso en el área de la MDE no es menos importante. Los investigadores han utilizado los distintos enfoques de clasificación para predecir si el estudiante tendrá éxito o no en la educación superior, las notas que alcanzará, cuál tarea (la más adecuada) debe resolver a continuación, entre otros.

Principios fundamentales de la clasificación en la MDE

- ✓ **Clasificadores discriminativos y probabilísticos**

La forma básica de los clasificadores es conocida como discriminativa, porque determinan un valor de clase para cada instancia de los datos. Si *M* es un clasificador (modelo), *C= {c1,…, ci}* el conjunto de valores de clase y *t* una instancia de datos, entonces la predicción de clase es *M(t) = ci* solamente para una *i*.

Una alternativa es un clasificador probabilístico, el cual define la probabilidad de clases para cada uno de las instancias clasificadas. Ahora *M(t) = [P(C=c1|t),…,P(C= ci|t)]*, donde *P(C= ci|t)* es la probabilidad de que *t* pertenezca a la clase *ci.*

- ✓ **Precisión de la clasificación**

La precisión (certeza) de la clasificación sobre un conjunto de datos *r* se mide por el índice de clasificación, el cual define la proporción de instancias correctamente clasificadas en el conjunto *r*. Si la

clase predicha por el clasificador *M* para la instancia *t* es *M(t)* y la clase real es *C(t*), entonces la precisión es:

$$Cr = \frac{\#\text{ Instancias en r donde M(t)= C(t)}}{\#\text{ de instancias en r}}$$

Donde # instancias es una abreviación de número de instancias.

El error de clasificación en el conjunto de datos *r* es simplemente la proporción de instancias clasificadas erróneamente en *r*: *err = 1- cr.*

Cuando *r* es el subconjunto de entrenamiento, el error se llama error de entrenamiento. Si *r* tiene la misma distribución que la población completa (por ejemplo, todos los futuros estudiantes de un curso), entonces el error de entrenamiento brinda un buen estimado del error de generalización también. Desafortunadamente, ésto rara vez sucede en el dominio de la educación. Los subconjuntos de entrenamiento son tan pequeños que no pueden captar la distribución real y el clasificador obtenido está seriamente influido (14).

- ✓ **Sobrentrenamiento**

El sobrentrenamiento es un problema importante relativo a la precisión. Sobrentrenamiento significa que el modelo se ha adecuado tanto a los datos de entrenamiento que expresa incluso los errores y excepciones más raros que en ellos aparecen. El modelo resultante está tan especializado que es incapaz de generalizar para datos futuros.

El sobrentrenamiento ocurre cuando el modelo es demasiado complejo en relación con la cantidad de datos. Por el contrario. Si el modelo es demasiado simple, no puede captar ningún patrón esencial en los datos, por tanto no está bien entrenado (14).

- ✓ **Límites de clase lineal y no lineal**

El aspecto principal del poder de representación es la forma de límites de clase que puede representarse. Los clasificadores lineales pueden separar solamente dos clases, si son linealmente

separables; es decir, si existe un hiperplano (en el caso bi-dimensional solamente un línea recta) que separe los puntos en dos clases distintas. De lo contrario, las clases son linealmente inseparables (14).

- ✓ **Preprocesamiento de los datos**

El objetivo del preprocesamiento de los datos es el de mejorar la calidad de los mismos y producir buenos atributos para la clasificación. Las principales tareas son la limpieza de los datos (del inglés, data cleaning), la extracción de rasgos relevantes (del inglés, feature extraction) y la selección de los rasgos relevantes (del inglés, feature selection) (14).

En la limpieza de los datos se deben llenar los valores que falten y tratar de identificar y corregir los errores. En la extracción de rasgos relevantes, se producen nuevos atributos mediante la combinación y transformación de los atributos originales. En la selección de atributos relevantes, se selecciona un conjunto óptimo de atributos.

El dilema radica en que la exactitud de la extracción y selección de rasgos relevantes no puede ser comprobada antes de que el clasificador sea aprendido y probado. Si el número de atributos es grande, todas las posibilidades no pueden ser evaluadas.

Enfoques de clasificación. Su aplicación en la MDE

- ✓ **Árboles de decisión**

Un árbol de decisión representa un conjunto de reglas de clasificación en forma de árbol. Todo camino raíz-hoja corresponde a una regla de la forma$Ti1 \wedge \ldots \wedge \ldots \wedge Til \rightarrow (C = c)$, donde c es el valor de clase en la hoja y cada Tik es una condicional booleana sobre un atributo Aij.

La idea básica es la de particionar el espacio de atributos hasta que se alcance algún criterio de parada en cada hoja. Los árboles de decisión tienen muchas ventajas: son simples y fáciles de entender, pueden manejar variables mixtas (es decir, tanto numéricas como categóricas), pueden clasificar nuevos ejemplos con rapidez y son flexibles. Extensiones de los árboles de decisión pueden manejar con facilidad pequeñas cantidades de ruido y de valores faltantes. Los árboles de decisión tienen un alto poder de representación, ya que pueden aproximar límites de clases no lineales.

La mayor restricción de los árboles de decisión es que asumen que todos los puntos de datos en el dominio pueden ser clasificados determinísticamente con exactitud dentro de una clase. Como

resultado, todas las inconsistencias son interpretadas como errores y los árboles de decisión no son adecuados para dominios intrínsecamente no deterministas. Un ejemplo de ello son los datos del desempeño en un curso, donde una proporción significativa de las instancias pueden ser inconsistentes.

Otro problema de los árboles de decisión es que son muy sensibles al sobrentrenamiento, especialmente con los grupos de datos pequeños. En las aplicaciones educacionales, los datos futuros rara vez siguen la misma distribución que la del conjunto de entrenamiento y se puede necesitar un modelo más robusto (14).

- ✓ **Naive de Bayes**

Este método simple e intuitivo se basa en la regla de Bayes para la probabilidad condicional. La regla de Bayes establece que si se tiene una hipótesis H y evidencia E que la soporta, entonces

$$Pr[H|E] = \frac{Pr[E|H] * Pr[H]}{Pr[E]}$$

donde se entiende que Pr[A] es la probabilidad de un evento A y Pr[A|B] denota la probabilidad de A dado B (14).

Este método asume "ingenuamente" (naive) la independencia de los atributos -puesto que solo es válido multiplicar probabilidades cuando los eventos son independientes. Esta asunción es realmente simplista. No obstante, el Naive Bayes es muy efectivo cuando se le prueba en conjuntos de datos reales, en especial si se le combina con procedimientos de selección de atributos que eliminen la redundancia en los atributos (14).

El Naive Bayes podría fallar cuando un valor particular de un atributo no está asociado a ninguna clase en el conjunto de entrenamiento. En ese caso, la probabilidad de ese valor seria cero y, dado que las otras probabilidades son multiplicadas por esta, la probabilidad final de un valor de clase seria igual a cero. Existen mecanismos para contrarrestar esta problemática, el más utilizado es el estimador de Laplace. Este mecanismo asegura que un valor de atributo que no tenga ocurrencias reciba una probabilidad distinta de cero, si bien muy pequeña.

Una de las ventajas del Naive Bayes es su muy buena respuesta ante los atributos faltantes. Si un faltara un valor el cálculo simplemente omitiría este atributo. Si el valor faltara en el conjunto de entrenamiento, este no se incluiría en el conteo de frecuencias y los valores de probabilidad estarían basados en el número de valores que si aparecen, en vez de basarse en el total de instancias.

Los valores numéricos se calculan generalmente asumiendo que tienen una distribución de probabilidad "normal" o "gausiana". Para los valores numéricos se listan los valores que ocurren; luego se calculan la media y la desviación estándar para cada clase y para cada atributo. La función de densidad de probabilidad para una distribución normal con media μ y desviación estándar σ está dada por la formula:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

La función de densidad de probabilidad para un evento está muy ligada a la probabilidad de dicho evento. Sin embargo, no son la misma cosa. El verdadero significado de la función de densidad f(X) es que la probabilidad de que una cantidad se encuentre dentro de una pequeña región, digamos entre x − ε/2 y x + ε/2, es ε × f(x). El mismo valor ε aparecería en ambas probabilidades de clase (si o no, positivo o negativo), por lo que se seguirían y cancelarían cuando se calcularan las probabilidades (14).

- ✓ **Redes neuronales**

Las redes neuronales artificiales son muy populares en el reconocimiento de patrones y con toda justicia. Las Redes Neuronales de Alimentación hacia Adelante (RNAA) son el tipo de red neuronal más ampliamente utilizado (14). La arquitectura de una RNAA se divide en capas de nodos: uno para los nodos de entrada, otra para los nodos de salida y al menos una capa para nodos ocultos. En cada capa oculta, los nodos están conectados a las capas anterior y siguiente de nodos; y las aristas están asociadas con pesos individuales.

El algoritmo de aprendizaje es una parte esencial en el modelo de red neuronal. El aprendizaje resulta computacionalmente difícil y los resultados dependen de varios parámetros tales como el número de capas ocultas, número de nodos ocultos en cada capa, pesos iniciales y el criterio de parada. En especial la selección de la arquitectura (topología de la red) y del criterio de parada son tareas críticas, porque las redes neuronales son muy propensas al sobrentrenamiento.

Las RNAA tienen más de una característica atractiva. Pueden aprender con facilidad límites no lineales y en principio representar cualquier tipo de clasificador. Si las variables originales no son discriminatorias, las RNAA las transforman implícitamente. Además, las RNAA son resistentes al ruido y pueden ser actualizadas con nuevos datos (14).

La principal desventaja es que las RNAA necesitan muchos datos muchos más de los que contienen las bases de datos educacionales típicas. Los mismos deben ser numéricos y aquellos que sean categóricos tienen que ser cuantificados de algún modo antes de que puedan ser utilizados, esto incrementa la complejidad del modelo y los resultados dependen del método de cuantificación empleado (14).

- ✓ **Clasificadores K-nearest neighbors**

Los clasificadores K-nearest neighbors (K-vecinos más cercanos) no construyen ningún modelo global explícito, sino que lo aproximan sólo de forma local e implícita. La idea principal es la de clasificar un nuevo objeto, examinando para ello los valores de clases de los *K* puntos más similares a él. La clase seleccionada puede ser o la clase más común entre los vecinos o una distribución de clase dentro del vecindario.

La única tarea de aprendizaje en los clasificadores K-nearest neighbors (K-vecinos más cercanos) es la de seleccionar dos parámetros importantes: un número *K* de vecinos y la distancia métrica *d.*

Los clasificadores nearest neighborhoods (barrios cercanos) tienen varias ventajas: sólo hay que aprender (o seleccionar) dos parámetros, la precisión en la clasificación puede ser muy buena para algunos problemas y la clasificación es bastante robusta ante el ruido y los valores faltantes. Tienen un poder representativo muy alto, porque pueden trabajar con cualquier tipo de límite de clase, siempre que se les provean datos suficientes.

La mayor desventaja es la dificultad para seleccionar la función distancia *d.* Los datos educacionales a menudo están conformados por datos numéricos y categóricos; y los atributos numéricos pueden estar representados en distintas escalas. Esto significa que no sólo se necesita una función de distancia con ponderaciones, sino también una gran cantidad de datos para aprender las ponderaciones

adecuadamente. Los atributos irrelevantes son otra característica común en algunos bancos de datos educacionales y han de ser removidos con anterioridad (14).

- ✓ **Regresión lineal**

La regresión lineal no es realmente un método de clasificación, pero sirve bien a ese propósito cuando todos los atributos son numéricos. Por ejemplo, aprobar un curso depende de los puntos del estudiante y los puntos pueden predecirse mediante la regresión lineal.

En la regresión lineal, se asume que el atributo objetivo es una función lineal de otros atributos mutuamente independientes. Sin embargo, el modelo es muy flexible y puede trabajar bien incluso si la dependencia real es sólo aproximadamente lineal, o si lo otros atributos están débilmente correlacionados. La razón es que la regresión lineal produce modelos muy simples, los cuales no corren el riesgo de sobrentrenarse como los modelos más complejos. De cualquier modo, los datos no deben tener grandes brechas (áreas vacías) y el número de excepciones (del inglés, outliers) debe ser pequeño (14).

- ✓ **Comparación de los paradigmas (enfoques) mencionados**

La tarea de seleccionar el enfoque de clasificación más adecuado para la solución del problema es una de las más críticas; y a la vez una de las más difíciles. No existe una regla general para ello, escoger uno u otro enfoque dependerá y mucho de la naturaleza del problema que intentamos resolver.

Sin embargo, a modo de guía, se muestra una tabla confeccionada por Romero y Ventura (14), la cual compara los enfoques antes descritos atendiendo a ocho criterios generales: límites de clase no lineales, precisión con pocos datos, trabajo con datos incompletos, soporte a variables mixtas, interpretación natural, razonamiento eficiente, aprendizaje eficiente, actualización eficiente.

El primer criterio está relacionado con el tipo de límite de clase. Los árboles de decisión, las redes de Bayes generales, las RNAA y los clasificadores nearest neighbors (vecinos más cercanos), pueden representar límites con una no linealidad elevada. El modelo naive de Bayes, si utiliza datos nominales, puede representar sólo un subconjunto de límites lineales; pero con datos numéricos puede representar límites no lineales muy complejos. La regresión lineal se restringe sólo a los límites lineales, pero tolera pequeñas desviaciones de la linealidad.

El segundo criterio, precisión con pocos datos, es crucial para el dominio educacional. Un clasificador preciso no puede ser aprendido si no existen datos suficientes para ello. La cantidad suficiente de datos depende de la complejidad del modelo. En la práctica, se debe favorecer el uso de modelos simples, tales como los clasificadores naives Bayes o la regresión lineal; los árboles de decisión, las RNAA y los clasificadores nearest neighbors (vecinos más cercanos) requieren una cantidad de datos mucho mayor para poder trabajar con exactitud. La precisión de las redes de Bayes generales depende de la complejidad de la estructura seleccionada.

El tercer criterio se refiere a si el método puede o no, manejar los datos incompletos, es decir ruido (errores), excepciones (del inglés, outliers) y valores faltantes. Las redes naive y general de Bayes, las RNAA y los nearest neighbors (vecinos más cercanos) son especialmente robustos ante el ruido en los datos. Los clasificadores bayesianos, los nearest neighbors (vecinos más cercanos) y algunas extensiones de los árboles de decisión pueden manejar valores faltantes bastante bien. Sin embargo los árboles de decisión son generalmente muy sensibles a los pequeños cambios tales como el ruido en los datos. La regresión lineal no puede manejar los valores faltantes y la ocurrencia de muchas excepciones podría corromper todo el modelo.

El cuarto criterio dice si el método soporta el uso de variables mixtas, es decir, numéricas y categóricas. Todos los métodos son capaces de manejar los atributos numéricos, pero los categóricos resultan problemáticos para las RNAA y la regresión lineal.

	AD	NB	BG	RNAA	K-nn	RL
Límites no lineales	+	(+)	+	+	+	–
Precisión con pocos datos	–	+	+/-	–	–	+
Trabaja con datos incompletos	–	+	+	+	+	–
Soporta variables mixtas	+	+	+	–	+	–
Interpretación natural	+	+	+	–	(+)	+
Razonamiento efectivo	+	+	+	+	–	+
Aprendizaje efectivo	+/-	+	–	–	+/-	+
Actualización efectiva	–	+	+	+	+	+

Un signo + significa que el método soporta la propiedad, - significa que no.
Las abreviaciones son: AD, árbol de decisión; NB modelo naive de Bayes;
BG, modelo de Bayes general; RNAA, red neuronal de alimentación hacia adelante;
K-nn, clasificador K-nearest neighbor; RL, regresión lineal.

Figura 1: Comparación de los distintos paradigmas de clasificación.

La interpretación natural (sencilla) es otro criterio importante, ya que los modelos educacionales deben ser transparentes al usuario. Todos los paradigmas excepto las redes neuronales, ofrecen modelos más o menos entendibles. En especial los árboles de decisión y las redes de Bayes tienen una representación visual muy fácil de entender.

El último criterio está relacionado con la eficiencia computacional de la clasificación, del aprendizaje y de la actualización del modelo. Lo más importante es la clasificación eficiente, puesto que el sistema debe adaptarse inmediatamente a la situación actual del estudiante.

El clasificador nearest neighbors (vecinos más cercanos) el único que le falta esta propiedad. La eficiencia del aprendizaje no es tan crítica, porque no se realiza en tiempo real. En algunos métodos, los modelos pueden ser actualizados de manera eficiente dados los nuevos datos. Esto es una característica atractiva, porque a menudo se pueden recolectar nuevos datos cuando el modelo ya está en uso (14).

1.2 Estudios vinculados al campo de acción

Los estudios llevados a cabo por los investigadores en el área de la predicción académica, sirven para confirmar teorías, destruir mitos y propiciar cambios en la forma que tienen las instituciones de entregar los servicios educacionales. Sin embargo, dado el hecho de que cada centro de estudio, más aun, cada curso (asignatura), tiene sus peculiaridades, los resultados de dichos experimentos no pueden ser generalizados. Sirven a lo sumo, como marco referencial y teórico para otras investigaciones tal es el caso de esta que aquí se presenta.

El grupo de avanzada en el área de la Minería de Datos Educacional, más específicamente en el campo de la predicción académica, está conformado por países desarrollados. Los principales estudios han sido realizados en España, EEUU, Canadá y Australia.

Dentro del problema en el cual se enmarca la presente investigación, es decir la predicción de los resultados finales de una asignatura, pueden destacarse los siguientes trabajos: “Mining LMS data to develop an ‘‘early warning system” for educators: A proof of concept", de los autores Leah P. Macfadyen, Shane Dawson (Universidad de la Columbia Británica, Vancuover, Canadá); y “Anticipating Students’ Failure As Soon As Possible”, de la autora Cláudia Antunes.

El primero de estos estudios refleja el análisis realizado a un curso de biología totalmente en línea (on-line) ofrecido por la Universidad de la Columbia Británica durante el año 2008. Los datos fueron extraídos de una plataforma basada en el LMS: Black Board Vista. Los mismos incluían algunos aspectos como el tiempo que los estudiantes habían dedicado a algunas de las actividades soportadas por la herramienta (evaluaciones, tareas, entre otras). Los resultados obtenidos destacan la fuerte relación existente entre las actividades de compromiso (foros, chat, juegos, entre otros) provistas por la herramienta y la nota final obtenida por los estudiantes (15).

Puesto que el estudio era basado en un curso completamente en línea, sus resultados no pudieron ser utilizados ni ajustados a los propósitos de esta investigación. Sin embargo, pudo utilizarse la idea de implementar un sistema de alerta temprana; además la forma en que se desarrolló el proceso de MD fue muy ilustrativa y útil para este trabajo.

El segundo estudio implicó el análisis de estudiantes enrolados en la asignatura Introducción a la Programación, en el Instituto Superior Técnico de Salamanca, España. Incluía 2050 instancias, cada una con dieciséis atributos: once evaluaciones semanales, una prueba, un proyecto, un examen, otro examen adicional y la calificación final del curso. De estos atributos, doce eran considerados observables, es decir su valor era conocido al momento de realizar la predicción; tres de ellos eran considerados no observables y la calificación obtenida al final de semestre era el atributo a predecir (clase).

El estudio comparaba los resultados de un clasificador “pesimista”, el cual solo utilizaba los atributos observables en la clasificación; y otro basado en reglas de asociación: clasificador “as soon as possible” (ASAP), que partiendo de los atributos observables, utilizaba reglas de asociación para aproximar los valores de aquellos no observables y luego realizaba la clasificación (con los valores reales y los aproximados). Ambos fueron entrenados utilizando árbol de decisión C4.5, implementado por Weka. La comparativa mostraba un desempeño ligeramente superior para el clasificador ASAP (16).

A pesar de lo interesante del estudio, las condiciones de su realización diferían en varios aspectos de las de la investigación aquí presentada: primero, el tamaño del conjunto de datos era mucho mayor; segundo, al enfocarse en una sola asignatura, podían asumir que el número de evaluaciones y por tanto la cantidad de atributos observables y no observables eran estándar. No obstante, la diferenciación entre de los atributos observables y no observables fue de gran utilidad para el enfoque de la presente investigación.

En Cuba se han llevado pocos estudios sobre el tema; ejemplos de ellos son los trabajos: “Variables psicosociales y su relación con el desempeño académico de estudiantes de primer año de la Escuela Latinoamericana de Medicina” por los investigadores Carlos Alberto Román y Yenima Hernández Rodríguez; “Resultados diferenciales de la prueba diagnóstica sobre gráficos según procedencia de educación media superior” de la doctora Sonia Damiani Cavero, Facultad de Ciencias Médicas “Salvador Allende”; y más específicamente en la Universidad de las Ciencias Informáticas “Análisis para la predicción del éxito o fracaso académico de estudiantes de la Universidad de las Ciencias Informáticas mediante la teoría de conjuntos aproximados” tesis de pregrado de la ingeniera Neyvis Remón González (17).

Ninguno de estos trabajos explora el problema de predicción que aquí se aborda. Sin embargo, la ingeniera Neyvis Remón afirma que el centro de procedencia de los estudiantes y el nivel escolar de sus padres estaban correlacionados al éxito académico de los alumnos. Por esta razón, se decidió incluir estos datos durante el proceso de extraccion de atributos potenciales.

1.2.1 **Otros elementos a considerar**

De acuerdo con los investigadores Romero y Ventura (18), existen cuatro tipos de problemas en los que se han enfocado la mayoría de los experimentos de predicción (clasificación) realizados en el área educacional, ellos son:

Predicción del éxito académico: la tarea era clasificar el éxito académico del estudiante en el nivel superior. Los objetivos eran predecir abandonos (del inglés, dropouts) al inicio de los estudios, la graduación en tiempo, el desempeño general o la necesidad de una enseñanza de refuerzo (19).

Selección de la siguiente tarea: el objetivo era predecir el desempeño del estudiante en la próxima tarea, dados sus respuestas a las tareas anteriores. Este es un problema importante en la evaluación computarizada adaptativa, donde la idea es seleccionar la próxima pregunta de acuerdo con el nivel de conocimientos actual del estudiante. Algunos predecían si la respuesta sería correcta o incorrecta, otros predecían la nota del estudiante en la próxima tarea (19) .

Habilidades metacognitivas, hábitos y motivación de los estudiantes: la tarea consistía en clasificar las habilidades metacognitivas y otros factores que influyen en el aprendizaje. Los objetivos eran predecir el nivel de motivación o compromiso del estudiante, sus estilos de aprendizaje, habilidad en el uso del sistema de aprendizaje, analizar el “juego” con el sistema o la estrategia de intervención recomendada (19) .

Los estudios de los tres grupos anteriores se alejan del objetivo de esta investigación. La presente se enmarca dentro del problema de la **Predicción de los resultados finales de un curso.** Según Romero y Ventura, la tarea aquí es predecir el aprobado/desaprobado en un curso, los abandonos o la nota del estudiante (19).

Tras la revisión de más de 30 experimentos los autores mencionados, llegaron a las siguientes conclusiones:

Los conjuntos de datos eran relativamente pequeños (50-350 filas, en promedio 200), porque estaban restringidas por el número de estudiantes que tomaban parte en el curso. Usualmente los datos estaban integrados por un solo año de estudiantes, pero si el curso se había mantenido inalterado, era posible conjugar datos de varios años.

Los principales atributos no sólo fueron los ejercicios, tareas y actividades del estudiante en el curso; sino que además se utilizaron datos demográficos y resultados de cuestionarios. El número de atributos en principio podía ser grande (> 50), pero fue reducido a entre 3 y 10 antes de aprender cualquier modelo.

Los métodos más comunes fueron los árboles de decisión, las redes de Bayes, las redes neuronales, los clasificadores K-nearest neighbors (k vecinos más cercanos) y los métodos basados en la regresión. La precisión promedio fue sólo del 72%, pero en los mejores casos estuvo cerca del 90%. Los factores más importantes que afectaron la precisión de la clasificación fueron el número de valores de clase utilizados (mejor para el caso binario) y cuan temprano se realizó la predicción (mejor al final del curso, cuando todos los atributos están disponibles) (19).

Distintos tipos de variables se han utilizado para la predicción de los resultados del curso. Algunos estudios destacan la importancia de las variables socio-demográficas; otros como el realizado en la Escuela Latinoamericana de Medicina (ELAM) (20) indican además la significación de las variables psicológicas. La mayor parte de las investigaciones utiliza datos provenientes de los registros internos (logs) en sistemas para el aprendizaje en línea. Los resultados en exámenes diagnósticos (21) y las respuestas a cuestionarios específicamente diseñados, también han sido utilizados por algunos investigadores. Todo esto unido, por supuesto, a los registros de las evaluaciones y actividad del estudiante en el curso.

En la Universidad de las Ciencias Informáticas (UCI) se registran los datos socio-demográficos de cada estudiante, como parte del proceso de inscripción. Además los profesores mantienen un registro actualizado de la asistencia, evaluaciones frecuentes y notas de los exámenes parciales y finales para cada uno de sus estudiantes. Con todos estos datos se pretende generar y aprender un modelo que permita predecir con exactitud y prontitud aquellos estudiantes en riesgo de desaprobar los exámenes finales, propiciando la intervención oportuna del profesor.

1.3 Descripción de las metodologías, tecnologías y herramientas a utilizar

Para el desarrollo de un software se hace necesario el uso de tecnologías que apoyen y propicien su avance de forma rápida y eficiente, estas facilitan el trabajo y agregan funcionalidades que permiten al equipo de desarrollo la obtención de un producto final que cumpla con las expectativas del cliente y a la vez incluya las nuevas tendencias en el campo de la informática. Es por ello, que se han estudiado las metodologías, herramientas y tecnologías que a continuación se describen, para la construcción de la solución propuesta en la investigación.

1.3.1 **Metodología de desarrollo de software**

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. En un proceso de desarrollo, la experiencia ha demostrado que la clave del éxito de un proyecto de software es la elección correcta de esta, pues puede conducir al programador a desarrollar un buen sistema de software. La idea no es tratar de ver cuál es mejor o peor, sino de cuándo usar una y cuándo la otra, pues esto va de acuerdo al tipo de proyecto, a los recursos con los que se cuentan y a la facilidad de interacción con el usuario real.

Tipos de metodologías

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo.

Entre las metodologías de desarrollo se encuentran:

Tradicionales:

- ✓ RUP (Rational Unified Process)
- ✓ MSF (Microsoft Solution Framework)
- ✓ Win-Win Spiral Model

- ✓ Iconix

Ágiles:

- ✓ XP (eXtreme Programming)
- ✓ Scrum
- ✓ Crystal
- ✓ DSDM (Dynamic Systems Development Method)
- ✓ FDD (Feature Deriven Development)
- ✓ Extreme Modeling

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

Para esta sección fueron descritas las metodologías SCRUM y XP:

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. Con Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Así mismo le permite en cualquier momento ajustar el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración. Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades. (22)

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes. (22)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (23).

Es utilizada en proyectos con requisitos muy cambiantes ya que se basan en proyectos pequeños, caracterizados por su sencillez, tanto en el aprendizaje como en su aplicación. Considerada ligera, flexible, predecible, de bajo riesgo, y no por ello menos científica. Entre otras ventajas pueden mencionarse los pocos requerimientos de documentación y planificación, así como la exigencia de tener siempre el cliente disponible para el desarrollo, implicando una mejor correspondencia entre el producto y la necesidad del negocio, promoviendo y propiciando un buen clima de trabajo en equipo. Uno de los objetivos fundamentales de XP es la satisfacción del cliente, ya que la metodología trata de brindar el software que este necesite y cuando lo necesite. Por tanto, se debe responder muy rápido a sus necesidades, incluso cuando los cambios sean al final de ciclo de la programación (23).

Fundamentos de la selección

En cuanto a las diferencias que existen entre ambas metodologías, XP propone que las tareas que se van entregando a los diferentes clientes sean modificadas; Scrum plantea que si lo que se termina funciona y está bien, se aparta y ya no se toca. XP distribuye el trabajo en parejas, en Scrum cada miembro trabaja de forma individual. El equipo de desarrollo de XP sigue estrictamente el orden de prioridad de las tareas definidas por el cliente, en Scrum el orden es modificable.

Teniendo en cuenta el proyecto a desarrollar y en correspondencia con las condiciones de trabajo, se determina que la metodología más adecuada es XP. A continuación algunas de las razones que justifican el uso de esta metodología.

- ✓ Se cuenta con un equipo de desarrollo pequeño (sólo dos personas) y poco tiempo para la implementación.

- ✓ El proyecto es pequeño y XP está concebida para ser utilizada dentro de proyectos que cumplan esta característica.
- ✓ El cliente forma parte del equipo de desarrollo y mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- ✓ El sistema será realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.

1.3.2 **Lenguaje de modelado**

Para modelar la integración se utilizará Unified Modeling Language (UML por sus siglas en inglés), este es un lenguaje útil para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (del inglés, OMG) y es el estándar de descripción de esquemas de software. UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación (24).

1.3.3 **Lenguajes de programación**

Un lenguaje de programación, es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se ponen a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. Representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona, son independientes de las computadoras a utilizar (25).

PHP, acrónimo de "Hipertexto Preprocesar" es un lenguaje interpretado por el servidor generando un HTML[1] con el resultado de sustituir las secuencias de instrucciones php por su salida. Ampliamente

[1] Lenguaje muy sencillo que se utiliza en la construcción de páginas web.

utilizado, de código abierto y de propósito general bajo licencia GPL[2]. Es un lenguaje de scripting adecuado para el desarrollo web y embebido en páginas HTML. El principal objetivo del lenguaje es permitir que los desarrolladores web escriban páginas generadas dinámicamente de forma rápida, pero se puede hacer mucho más con PHP (26).

Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos, permitiendo al desarrollador realizar varias funciones, desde generar documentos en pdf hasta analizar código XML[3]. Es multiplataforma y con capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL[4], lo cual permite la creación de aplicaciones web robustas (27).

Java es un lenguaje de programación y multiplataforma, lo que quiere decir que se ejecuta en la mayoría de los sistemas operativos, inclusive en sistemas operativos de móviles. Es un software de distribución libre, no es necesario pagar una licencia para poder desarrollar en él. Así mismo es un lenguaje muy completo y poderoso, se pueden realizar muchas tareas con el mismo, pues posee una librería y utilidades muy completas que facilitan la programación, aunque tiene la desventaja de ser un lenguaje de ejecución lenta, debido al uso de la máquina virtual de Java. Entre sus principales características se incluyen: Simple, Orientado a Objetos, Multihilo, Dinámico Arquitectura Neutral, Portable, Alto Rendimiento, Robusto, Seguro (28).

Fundamentos de la selección

Para el desarrollo del sistema se eligió el lenguaje de programación Java, ya que es una tecnología libre, por lo que no es necesario pagar para poder utilizarlo, posee versatilidad al momento de escribir código, sencillez en la sintaxis, e inclusive su seguridad.

[2] Licencia Publica General La licencia, puede ser usada por cualquiera, su finalidad es proteger los derechos de los usuarios finales (usar, compartir, estudiar, modificar).

[3] Lenguaje de Etiquetado Extensible, juega un papel fundamental en el intercambio de una gran variedad de datos.

[4] Sistema Administrativo Relacional de Bases de Datos (RDBMS, por sus siglas en inglés).

1.3.4 **Herramientas de desarrollo**

Un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (29).

NetBeans: Programa que sirve como IDE que permite programar en distintos lenguajes, es ideal para trabajar con el lenguaje de desarrollo JAVA (y todos sus derivados), además ofrece un excelente entorno para programar en PHP. También se puede descargar una vez instalado NetBeans, los complementos para programar en C++. La IDE de NetBeans es perfecta y muy cómoda para los programadores. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código (29).

Características de NetBeans

- ✓ Formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web.
- ✓ Ofrece documentación y recursos de formación exhaustivos.
- ✓ Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas.
- ✓ Es un IDE multiplataforma.
- ✓ Está desarrollado para usarse generalmente con lenguaje Java, aunque permite su uso con otros lenguajes de programación como PHP.
- ✓ Permite crear aplicaciones Web con PHP5 (30).

Eclipse es un IDE de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java así como también para PHP u otros, donde se encuentran todas las herramientas y funciones necesarias para trabajar, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar (31).

Características de Eclipse:

- ✓ Su principal inconveniente es el consumo de recursos del sistema, que es común a otros IDE en mayor o menor medida.
- ✓ Es totalmente libre y no es necesario comprar ninguna licencia comercial, lo cual lo hace atractivo para muchos.
- ✓ Es multiplataforma.
- ✓ Es un potente editor de texto con la capacidad de resaltar sintaxis, así como también una compilación en tiempo real y soportes (32).

Fundamentos de la selección

Se decide utilizar **NetBeans7.2.1** debido a que ofrece opciones de desarrollo con plugins y herramientas incorporadas, así como completamiento de código, facilitando el trabajo a los programadores para la creación de aplicaciones web con rapidez. Es una herramienta fácil de instalar y consume pocos recursos. El editor de código fuente que presenta es muy ágil y a la vez robusto, características que hacen que sea una excelente herramienta para el desarrollo de soluciones informáticas.

1.3.5 **Herramientas CASE**

Las herramientas Ingeniería de Software Asistida por Computación (CASE por sus siglas en inglés), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el ciclo de vida de desarrollo del software (33).

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación. Proporciona mecanismos para realizar la Ingeniería Inversa, es decir, que a partir del código se pueda obtener información sobre su diseño; adicionalmente permite generar código en diferentes lenguajes a partir de un diseño en UML, brinda la posibilidad de que varias personas trabajen a la vez, permitiendo que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y permite que tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo (34).

Visual Paradigm es una herramienta CASE que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el analista (35).

Características principales:

- ✓ Entorno de creación de diagramas para UML
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad en múltiples plataformas (35).

Fundamentación de la selección

Luego de un detallado estudio se decidió escoger para el modelado del sistema la herramienta CASE Visual Paradigm (versión 8.0), debido a las características del software y el entorno en que se desarrolla el sistema, ya que es una herramienta multiplataforma de modelado visual UML, muy potente y fácil de utilizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML, ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste.

1.3.6 Sistema Gestor de Base de Datos

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL (36).

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración (36).

Las características principales de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos (36).

PostgreSQL es el sistema de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de Multihilo para garantizar la estabilidad del sistema. Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD[5] y con código fuente disponible libremente. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (37).

Posee numerosas ventajas entre ellas se pueden mencionar:

- ✓ Instalación ilimitada.
- ✓ Mejor soporte que los proveedores comerciales.
- ✓ Estabilidad y confiabilidad legendarias.
- ✓ Extensible pues el código fuente está disponible para todos sin costo.
- ✓ Multiplataforma.
- ✓ Diseñado para ambientes de alto volumen.

[5] Familia de licencias de software libre permisiva. La licencia original se utilizó para la Berkeley Software Distribution (BSD).

- ✓ Altamente extensible, pues soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- ✓ Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

Fundamentación de la selección

Luego de realizar el estudio de los diferentes gestores de base de datos se determinó usar **PostgreSQL** en su versión 9.2 debido por ser el sistema de gestión de base de datos relacional, orientada a objetos de código abierto más avanzado del mundo. Es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Posee una gran escalabilidad, haciéndolo idóneo para ser usado en aplicaciones que realicen varias peticiones al día. Permite la gestión de usuarios, como también los permisos asignados a cada uno de ellos. Es altamente extensible pues soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Posee alta concurrencia permitiendo que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

1.3.7 **Herramienta de Minería de Datos**

Weka es una extensa colección de algoritmos de máquinas de conocimientos implementados en java. Estos algoritmos son útiles para ser aplicados sobre datos mediante los interfaces que ofrece o para embeberlos dentro de cualquier aplicación. Esta herramienta permite realizar un grupo de transformaciones necesarias sobre los datos, trae implementada un grupo de tareas de la MD: clasificación, regresión, agrupamiento, asociación y visualización. Weka está diseñada como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla (38).

Entre sus características se encuentra que, contiene un grupo de herramientas que permiten el análisis de los datos. Su licencia es GPL, lo que significa que este programa es de libre distribución y difusión. Además está programado en Java, es independiente de la arquitectura y funciona en cualquier plataforma sobre la que haya una máquina virtual de Java disponible (38).

RapidMiner es el líder mundial de código abierto para la minería de datos debido a su combinación de tecnología de primera calidad y gran número de funcionalidades. Esta aplicación cubre un amplio rango de minería de datos. Además de ser una herramienta flexible para aprender y explorar la minería de datos, la interfaz gráfica de usuario tiene como objetivo simplificar el uso para las tareas complejas de esta área (39).

Características de RapidMiner

- ✓ Es un sistema de prototipo para el descubrimiento del conocimiento y MD.
- ✓ Es un software de tipo Open-Source con licencia GNU GPL, basado en java.
- ✓ Trabaja bajo las plataformas Windows y Linux.
- ✓ La característica más importante es la capacidad de jerarquizar cadenas del operador y de construir complejos árboles de operadores (39).

SAS Enterprise Miner es una solución de minería de datos que proporciona gran cantidad de modelos y de alternativas. Permite determinar pautas y tendencias, explica resultados conocidos e identifica factores que permiten asegurar efectos deseados. Además, compara los resultados de las distintas técnicas de modelado, tanto en términos estadísticos como de negocio, dentro de un marco sencillo y fácil de interpretar (40).

SAS Enterprise Miner es la herramienta de minería de datos comercializada por SAS Institute, compañía que radica en Cary (Carolina del Norte, EE.UU.) y es solo uno de los muchos componentes del sistema integrado SAS (40).

Fundamentación de la selección

La herramienta seleccionada para realizar el proceso de Minería de Datos es Weka en su versión 3.7.6, pues a pesar de no ser la más usada, posee características acordes a las necesidades de esta investigación. A continuación se describen algunas de las razones que justifican el uso de Weka.

- ✓ Es de distribución libre y multiplataforma.
- ✓ Cuenta con una librería que puede ser embebida en la propia aplicación.

- ✓ Es una de las herramientas de MD más utilizadas. Muchos de los experimentos consultados hacían uso de esta herramienta.
- ✓ Cuenta con amplia documentación y ayuda en línea.
- ✓ Además, el conocimiento previo de la herramienta por parte de los autores, permitió un ahorro considerable de tiempo.

Conclusiones

En este capítulo se han abordado los principales conceptos vinculados al campo de acción, para lograr un mayor entendimiento del tema de investigación. Como resultado de la revisión de los estudios existentes en el mundo y en Cuba, puede concluirse que, si bien dichos experimentos proporcionan aportes e ideas valiosos, presentan peculiaridades que imposibilitan su aplicación al contexto de la UCI. Se decidió obtener un modelo que se adapte mejor a las características del problema planteado, tomando como marco referencial la literatura consultada. Se trataron los aspectos relacionados con el objeto de estudio, lenguajes de programación, herramientas, tecnologías y metodología. Además se definió Weka como herramienta de Minería de Datos a utilizar durante el proceso de creación del modelo.

Capítulo 2: Características del sistema

Características del sistema

Introducción

Después de haber analizado el estado del arte y elegido las herramientas y metodología a utilizar para el correcto desarrollo de la aplicación, están todas las condiciones creadas para hacer la propuesta de solución al problema existente. Para implementar la solución propuesta, se siguieron los pasos definidos por la metodología XP en las fases de exploración y planificación.

2.1 Modelo de Dominio

El modelo de dominio representa un modelo de conceptos y no un modelo de objetos. Es un diccionario visual de términos importantes del dominio, utiliza la notación UML de diagrama de estructura estática. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema de software (41).

2.1.1 **Diagrama de conceptos del dominio**

A continuación se muestra el modelo de dominio que recoge los principales conceptos que se describen en el diseño del registro electrónico inteligente y la relación que existe entre sí.

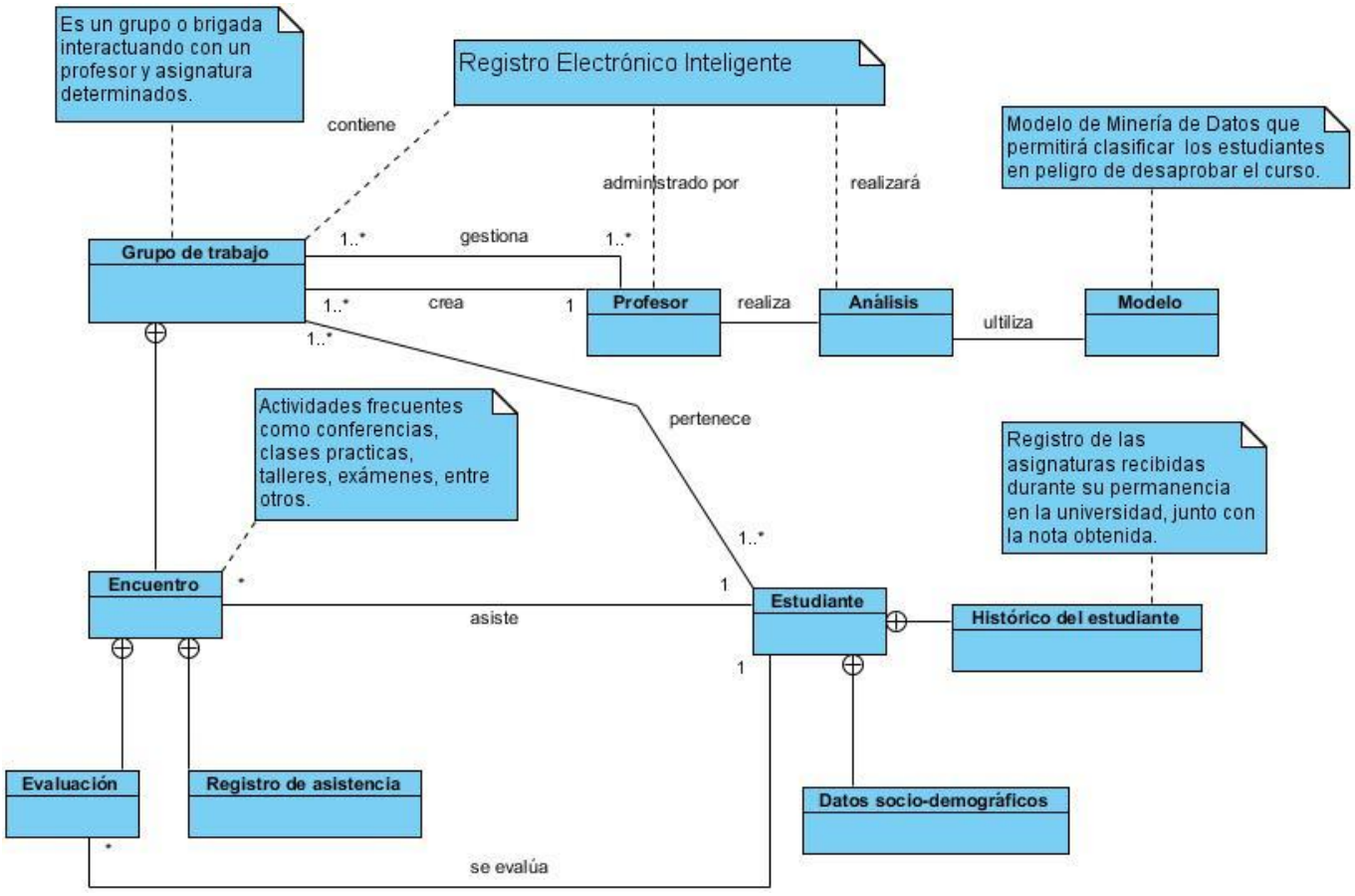


Figura 2: Diagrama de modelo de dominio.

2.1.2 **Definición de los conceptos del modelo del dominio**

Grupo de Trabajo: es la representación de un grupo de estudiantes interactuando con un profesor en una asignatura determinada.

Estudiante: representa un miembro de un grupo cuyos datos se almacenan para análisis.

Datos socio-demográficos: son los datos del estudiante referentes a (nombre, dirección, centro de procedencia, nombre del padre y la madre, ocupación laboral, entre otros).

Históricos del estudiante: representa el registro de evaluaciones y asignaturas acumuladas de los estudiantes durante su permanencia en el centro.

Profesor: es el encargado de realizar el análisis, registrar las asistencias y evaluaciones de los estudiantes en los encuentros.

Análisis: representa la utilización de algunas de las tareas de la minería de datos (clasificación, agrupamiento, entre otros), para obtener información valiosa sobre el desempeño de los estudiantes.

Modelo: modelo que permite clasificar los posibles estudiantes en peligro de suspender el curso.

Encuentro: son las actividades frecuentes que se realizan en el grupo como, conferencias, talleres, preguntas escritas, pruebas parciales, entre otros.

Registro de asistencia: es el registro de la asistencia del estudiante a los encuentros que se clasifica en (ausente, justificado y cuartelero).

Evaluación: son las evaluaciones que obtiene el estudiante en la clase.

2.2 Descripción del proceso de Clasificación

El proceso de clasificación en la Minería de Datos está compuesto por cuatro pasos básicos:

- ✓ **La recopilación de los datos**. Se acumulan los datos disponibles. Estos pueden encontrarse en una misma locación o pueden estar distribuidos.
- ✓ **Pre-procesamiento de los datos**. Incluye la limpieza de los datos, la extracción de características deseables y la selección del conjunto optimo de atributos.
- ✓ **Obtención del modelo.** Se corre uno o varios clasificadores en busca de un modelo efectivo.
- ✓ **Evaluación del modelo.**

2.2.1 **Recopilación de los Datos**

En busca de la solución al problema planteado, se realizó un estudio tomando como muestra 200 estudiantes de la Facultad 7, en la Universidad de las Ciencias Informáticas. La información comprendía los datos socio-demográficos del estudiante, su promedio y resultados obtenidos en las asignaturas hasta el momento; así como también datos tomados de los Registros de Asistencia y Evaluaciones Frecuentes (RAEF).

Los RAEF pertenecían al primer semestre del curso 2012-2013 y contenían la información de 7 grupos de la mencionada facultad. Las asignaturas representadas eran Matemática 1, Matemática Discreta 1, Probabilidades y Estadísticas e Investigación de Operaciones; todas con sistema de evaluación: 2 pruebas parciales, 1 examen final (ordinario), más evaluaciones frecuentes (preguntas escritas, evaluaciones orales, seminarios, talleres, entre otras). La información fue cargada en una base de datos PostgreSQL, diseñada para almacenar todos los datos recogidos.

2.2.2 **Pre-procesamiento de los datos**

El objetivo del preprocesamiento es mejorar la calidad de los datos y producir buenos atributos para la clasificación. Las principales tareas son la limpieza de los datos, la extracción de atributos y la selección de atributos. En la limpieza de los datos, se deben completar los valores faltantes y tratar de identificar y corregir los errores. En la extracción de atributos, se producen nuevos atributos mediante la transformación y la combinación de los originales. En la selección de atributos, se selecciona un conjunto de atributos óptimo (42).

Limpieza de los datos

Los datos obtenidos contenían muy pocos errores; sin embargo los valores faltantes eran comunes, sobre todo en los RAEF. Se buscó la forma de corregir dichos errores y de completar aquellos valores que faltaban. Al terminar el proceso de limpieza de los datos se contaba con un total de 164 estudiantes cuyos datos eran completos (socio-demográficos, histórico de asignaturas aprobadas, índice general y evaluaciones del RAEF).

Extracción de atributos

La cantidad de total de atributos sobrepasaba los 30. Se realizó un proceso de discriminación en busca aquellas características que resultaran más prometedoras a priori. En principio se determinaron 10 atributos potenciales:

- ✓ **Centro de procedencia:** es el centro escolar de donde procede el estudiante IPU, IPUEC, IPVCE, EMCC, etc.).
- ✓ **Nivel escolar de la madre y del padre** (universitario, preuniversitario, secundario, etc.)**.**
- ✓ **Índice académico:** es el índice académico general del estudiante hasta el momento.

- ✓ **Porciento de asistencia.**
- ✓ **Porciento de participación en clases:** es una medida de cuan activo es el estudiante en clases.
- ✓ **Promedio de evaluaciones frecuentes:** promedio de todas las evaluaciones (preguntas escritas, evaluaciones orales, seminario, talleres, entre otros) del estudiante en los encuentros**.**
- ✓ **Nota en la primera prueba parcial.**
- ✓ **Nota en la segunda prueba parcial.**
- ✓ **Corte evaluativo 1.**
- ✓ **Corte evaluativo2.**
- ✓ **Resultado final del curso:** es el atributo a predecir (aprobado o desaprobado).

Nótese que para obtener los atributos: porciento de asistencia, porciento de participación y promedio de evaluaciones frecuentes, era necesario realizar transformaciones sobre otros atributos. Los dos primeros solo requerían unas cuantas sumas y divisiones; el tercero requería, además, transformar valores nominales (E, B, R, M) en sus equivalentes numéricos (5, 4, 3, 2). También de discretizar el atributo clase (resultado final del curso), pues la evidencia de los estudios revisados sugería que los resultados de la predicción mejoraban cuando la clase tomaba valores binarios.

El siguiente paso era determinar cuan temprano debía realizarse la alerta, garantizando el tiempo suficiente para ayudar a aquellos estudiantes en riesgo; y, además, una disponibilidad de datos razonable para la construcción del modelo.

Selección de atributos

Se determinó que la predicción podría realizarse una vez que se tuvieran los resultados del primer corte evaluativo y de la primera prueba parcial. Además se decidió adoptar un enfoque pesimista, es decir, solo se utilizarían aquellos atributos cuyos valores fueran conocidos al momento de la clasificación (observables). De este modo el conjunto de atributos potenciales se redujo a los siguientes:

- ✓ **Centro de procedencia del estudiante.**
- ✓ **Nivel Escolar del padre.**
- ✓ **Nivel escolar de la madre.**

- ✓ **Porciento de Asistencia.**
- ✓ **Promedio de evaluaciones frecuentes.**
- ✓ **Porciento participación.**
- ✓ **Nota 1ra Prueba Parcial.**
- ✓ **Nota Corte Evaluativo 1.**
- ✓ **Resultado final del curso:** clase a predecir.

Con este nuevo conjunto de posibles atributos se procedió a confeccionar un fichero de tipo attibute-relation file format (ARFF), conteniendo las 164 instancias con sus respectivos valores de atributos. Este tipo de fichero es el que utiliza la herramienta Weka para la entrada de los datos. Una vez cargados los datos en la interfaz Explorer de Weka, se procedió a realizar la selección del conjunto óptimo de atributos.

Para realizar la selección de atributos, Weka implementa 6 algoritmos evaluadores de subconjuntos de atributos, los cuales toman un subconjunto de atributos retornando una medida numérica que guía la búsqueda; y 10 algoritmos de búsqueda, los cuales atraviesan el espacio de búsqueda tratando de encontrar un buen subconjunto, la calidad la mide el evaluador de subconjuntos de atributos.

En este caso se escogió el evaluador de subconjuntos de atributos CfsSubsetEval, este algoritmo evalúa la habilidad de cada atributo individualmente y el grado de redundancia entre ellos, prefiriendo los conjuntos de atributos que estén altamente correlacionados con la clase pero con baja interrelación… (43).

Para la exploración del espacio de búsqueda se escogió el algoritmo BestFirst, un escalador de colina con vuelta atrás.

Los resultados arrojados sugerían que el conjunto de datos óptimo estaba compuesto por los atributos: Índice General, Nota Primer Corte Evaluativo y Nota Primera PP.

2.2.3 **Obtención del Modelo**

Selección del algoritmo

En la comparación de enfoques de predicción realizada en el Capitulo 1, pudo adelantarse al Naive Bayes como el algoritmo que mejor se ajustaba a la naturaleza del problema y a las condiciones del proceso. No obstante varios modelos fueron obtenidos con otros algoritmos a fin de realizar una evaluación más acertada. A continuación se muestran los resultados obtenidos así como una breve interpretación de los mismos.

- ✓ **Decision Table**

El modelo obtenido tiene buen porciento de certeza en general (85.9756 %), pero falla en la mitad de los desaprobados (**Figura 3**). Este tipo de resultado no es bueno, porque a pesar de su buen desempeño en cuanto a instancias correctamente clasificadas, su capacidad para detectar los estudiantes desaprobados es a penas del 50%.

```
Decision Table:

Number of training instances: 164
Number of Rules : 4
Non matches covered by Majority class.
               Best first.
               Start set: no attributes
               Search direction: forward
               Stale search after 5 node expansions
               Total number of subsets evaluated: 39
               Merit of best subset found:    87.195
Evaluation (for feature selection): CV (leave one out)
Feature set: 4,10

Time taken to build model: 0.17 seconds

                                === Stratified cross-validation ===
                                      === Summary ===

Correctly Classified Instances         141               85.9756 %
Incorrectly Classified Instances        23               14.0244 %
Kappa statistic                          0.5644
Mean absolute error                      0.2351
Root mean squared error                  0.3502
Relative absolute error                 63.4431 %
Root relative squared error             81.5368 %
Coverage of cases (0.95 level)          98.7805 %
Mean rel. region size (0.95 level)      91.4634 %
Total Number of Instances              164

                                === Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  PRC Area  Class
                 0.968     0.475      0.863     0.968      0.913      0.788      0.887    aprobado
                 0.525     0.032      0.84      0.525      0.646      0.788      0.63     desaprobado
Weighted Avg.    0.86      0.367      0.858     0.86       0.848      0.788      0.824

                                      === Confusion Matrix ===

                                     a   b   <-- classified as
                                   120   4 |   a = aprobado
                                    19  21 |   b = desaprobado
```

Figura 3: Resultado del algoritmo Decision Table.

- ✓ **ZeroR (reglas)**

Este modelo es el de peor resultado, ZeroR (reglas) falla en la clasificación de todos los desaprobados. En la **Figura 4** se muestra el resultado arrojado por el mismo

```
                          Scheme:       weka.classifiers.rules.ZeroR
        Relation:     REI
        Instances:    164
        Attributes:   10
                          Centro_Procedencia
                          Nivel_Escolar_Madre
                          Nivel_Escolar_Padre
                          Promedio
                          Asistencia
                          Porciento_participacion
                          Evaluaciones_Frecuentes
                          Nota_1PP
                          Corte_Evaluativo_1
                          Resultado
        Test mode:    10-fold cross-validation

                          === Classifier model (full training set) ===

        ZeroR predicts class value: aprobado

        Time taken to build model: 0 seconds

                          === Stratified cross-validation ===
                              === Summary ===

        Correctly Classified Instances         124               75.6098 %
        Incorrectly Classified Instances        40               24.3902 %
        Kappa statistic                          0
        Mean absolute error                      0.3706
        Root mean squared error                  0.4295
        Relative absolute error                100      %
        Root relative squared error            100      %
        Coverage of cases (0.95 level)         100      %
        Mean rel. region size (0.95 level)     100      %
        Total Number of Instances              164

                          === Detailed Accuracy By Class ===

                   TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  PRC Area  Class
                     1         1          0.756      1         0.861      0.49      0.753    aprobado
                     0         0          0          0         0          0.49      0.24     desaprobado
Weighted Avg.    0.756     0.756      0.572     0.756     0.651      0.49      0.628

                                  === Confusion Matrix ===

                                    a    b   <-- classified as
                                  124    0 |   a = aprobado
                                   40    0 |   b = desaprobado
```

Figura 4: Resultado del algoritmo ZeroR (reglas).

- ✓ **Naive bayes**

Es el que mejor se desempeña en la clasificación de los desaprobados (67.5%). La mayor parte de las instancias mal clasificadas son “falsos desaprobados” (17), lo cual es preferible en el caso del problema planteado. A continuación se muestra en la **Figura 5** los resultados.

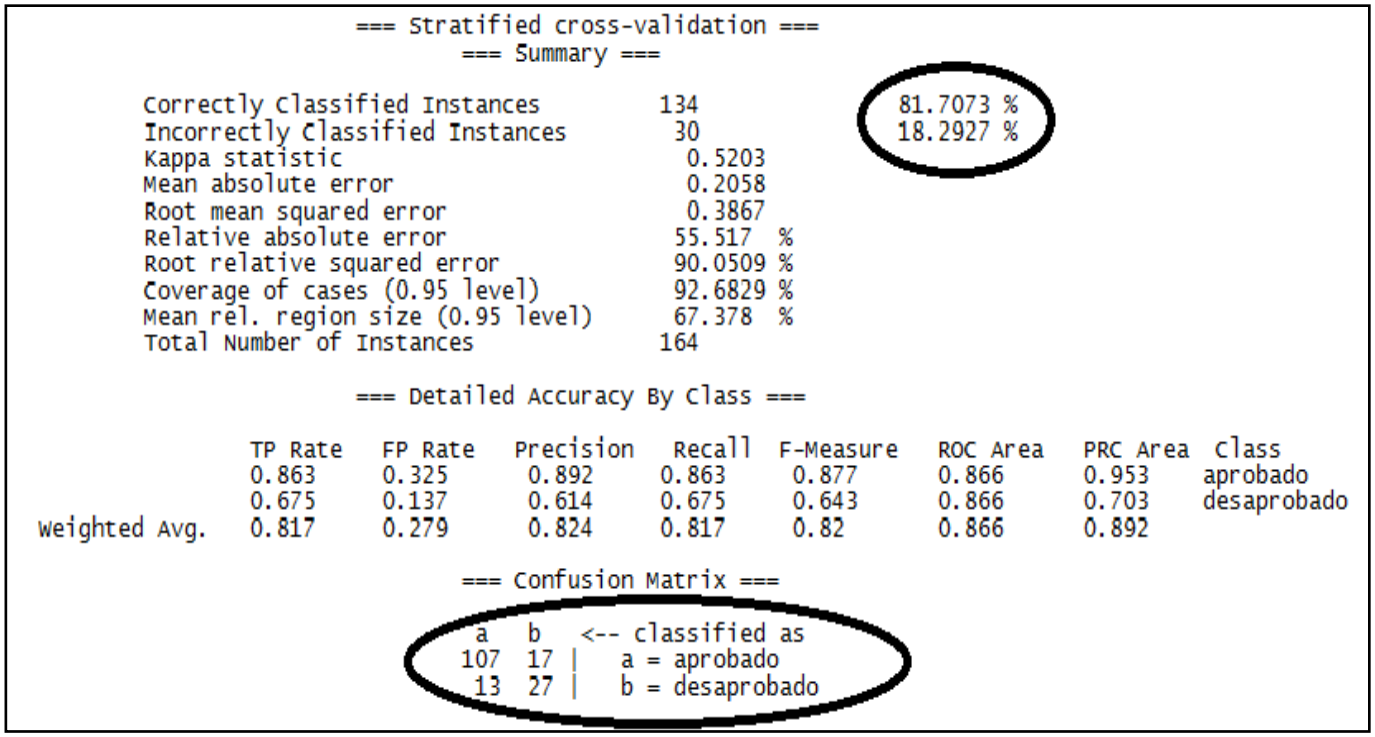
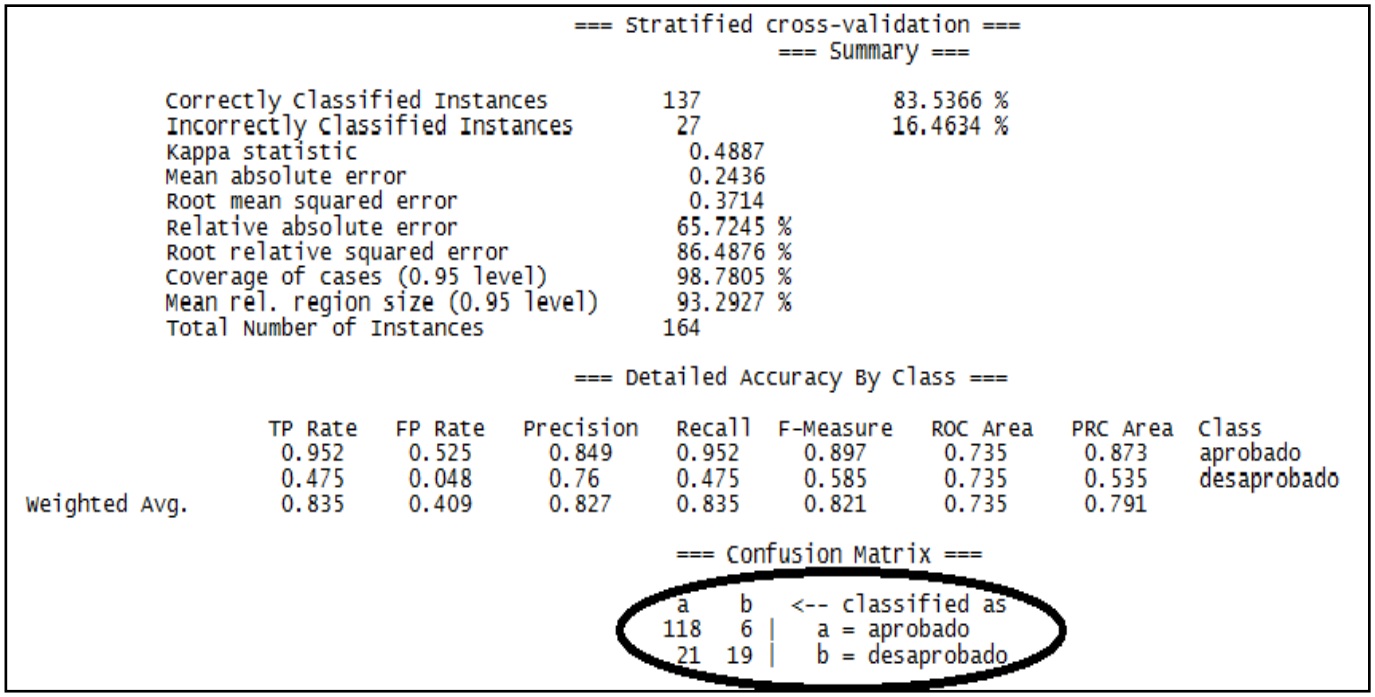


Figura 5: Resultado del algoritmo Naive Bayes.

- ✓ **Árbol de decisión J48.**

El porciento de certeza es mayor al del Naive de Bayes, pero teniendo en cuenta el pequeño número de instancias (164) y la propensión de los arboles de decisión al sobrentrenamiento este resultado no debe ser motivo de entusiasmo. Además el número de desaprobados clasificados erróneamente es mayor que el del Naive Bayes. En la **Figura 6** queda evidenciado el resultado**.**



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         137               83.5366 %
Incorrectly Classified Instances        27               16.4634 %
Kappa statistic                          0.4887
Mean absolute error                      0.2436
Root mean squared error                  0.3714
Relative absolute error                 65.7245 %
Root relative squared error             86.4876 %
Coverage of cases (0.95 level)          98.7805 %
Mean rel. region size (0.95 level)      93.2927 %
Total Number of Instances              164

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  PRC Area  Class
                 0.952     0.525       0.849     0.952      0.897      0.735     0.873    aprobado
                 0.475     0.048       0.76      0.475      0.585      0.735     0.535    desaprobado
Weighted Avg.    0.835     0.409       0.827     0.835      0.821      0.735     0.791

=== Confusion Matrix ===

   a   b   <-- classified as
 118   6 |   a = aprobado
  21  19 |   b = desaprobado
```

Figura 6: Resultado del algoritmo árbol de decisión J48.

Fundamentación de la selección del algoritmo adecuado.

Luego del análisis de los modelos antes expuestos y atendiendo a la revisión de los estudios relacionados en el campo de acción (Capítulo 1), se seleccionó el algoritmo Naive de Bayes por considerarse que muestra la solución más adecuada, en correspondencia con la naturaleza del problema a resolver y con los resultados predictivos arrojados en la herramienta Weka.

2.2.4 **Evaluación del modelo**

El método utilizado para evaluar el desempeño de los modelos descritos anteriormente, fue el “stratified 10-fold cross validation”. Este método es ideal para conjuntos de datos pequeños. Cuando se utiliza el stratified 10-cross validation, los datos se dividen aleatoriamente en 10 partes; en ellas la clase está representada más o menos en las mismas proporciones que en el conjunto de datos completos. Cada fracción de datos es reservada por turnos y el esquema de aprendizaje se realiza sobre los nueve décimos restantes; entonces el índice de error se calcula sobre el conjunto reservado. De modo que el

proceso de aprendizaje se realiza 10 veces sobre distintos conjuntos de entrenamiento. Finalmente, las 10 estimaciones de error se promedian para obtener un estimado general del error.

Selección de atributos II

Una vez escogido el Naive Bayes como algoritmo a utilizar, se procedió a realizar un segundo intento por lograr el conjunto de atributos que produjera el modelo más acertado. El método aplicado fue el siguiente: partiendo del conjunto de datos sugerido por la herramienta Weka (Índice General, Corte Evaluativo 1, 1ra Prueba Parcial) se fueron agregando atributos y obteniendo los respectivos modelos. Tras analizar de esta forma varias combinaciones de atributos, se obtuvo el conjunto de atributos siguiente: Índice General, Porciento de Asistencia, Promedio Participación, Promedio Evaluaciones Frecuentes, Corte Evaluativo 1, 1ra Prueba Parcial; el cual produjo los siguientes resultados:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         138    84.1463 %
Incorrectly Classified Instances        26    15.8537 %

Kappa statistic                          0.5909
Mean absolute error                      0.2128
Root mean squared error                  0.3752
Relative absolute error                 57.4133 %
Root relative squared error             87.3744 %
Coverage of cases (0.95 level)          96.3415 %
Mean rel. region size (0.95 level)      71.6463 %
Total Number of Instances              164

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  PRC Area  Class
                   0.871     0.25       0.915     0.871    0.893      0.864     0.953    aprobado
                   0.75      0.129      0.652     0.75     0.698      0.864     0.678    desaprobado
Weighted Avg.      0.841     0.22       0.851     0.841    0.845      0.864     0.886

=== Confusion Matrix ===

   a   b   <-- classified as
 108  16 |   a = aprobado
  10  30 |   b = desaprobado
```

Figura 7: Resultado del Naive Bayes luego de reajustar el conjunto de atributos.

2.3 Propuesta del sistema

Se implementará un sistema de alerta temprana, en el cual los profesores podrán informarse acerca del nivel docente de sus estudiantes. La aplicación deberá permitir la gestión de los datos asociados a los

estudiantes (datos socio-demográfico, históricos de asignatura). Además el profesor tendrá un control de sus grupos de trabajos a los cuales podrá agregar encuentros, evaluaciones.

El sistema deberá garantizar que los datos sean accedidos solo por los profesores con permisos para ello, garantizando la seguridad de los mismos. La información debe mostrarse de una forma que resulte útil y fácil de comprender.

La aplicación incorporará el modelo predictivo obtenido mediante el proceso de MD antes descrito. Con su ayuda, los profesores sabrán de antemano cuales estudiantes están en riesgo de desaprobar el examen final de su asignatura; permitiéndoles brindar atención diferenciada a estos estudiantes.

2.4 Personas relacionadas con el Sistema

En este epígrafe se identifica la persona que se relaciona con el sistema, esta se define como aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en la aplicación.

Personas relacionadas con el sistema	Justificación
Profesor	Se autentica en la aplicación y de este modo puede acceder a los servicios que se ponen a su disposición en el sistema.

Tabla1. Personas relacionadas con el sistema.

2.5 Fase de Exploración

La fase de exploración, es la fase en la que se define el alcance general del proyecto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, de igual forma se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. Toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (44).

2.5.1 **Historias de usuarios**

La identificación de las historias de usuario (HU) es la técnica utilizada para especificar los requisitos del software. El cliente describe brevemente en tarjetas de papel las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las HU es muy dinámico y flexible, cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas (45).

Para que el programador pueda dar respuesta a una historia de usuario, es necesario que se trace objetivos, que se podrían traducir en tareas de programación. Cada vez que se da cumplimiento a una tarea, se realizan un conjunto de pruebas de unidad para asegurarse que los componentes desarrollados funcionan.

A medida que se va dando cumplimiento a cada HU se presenta el software debidamente probado al cliente. Cuando se reúnen un número suficiente de funcionalidades que representan una versión útil y parcialmente completa de la aplicación se produce una liberación, lo cual es una versión funcional de la aplicación que aporta valor al negocio y que debe ser mantenida a la par que se desarrollan las siguientes funcionalidades.

A continuación se muestra el modelo de la plantilla que se llena para crear las historias de usuarios.

Historia de Usuario	
Número: Número de la HU	**Nombre:** El nombre de la HU, sirve para identificarla fácilmente entre los desarrolladores y los clientes.
Usuario: El usuario del sistema que utiliza o protagoniza la historia	
Prioridad en Negocio: Que tan importante es para el cliente.	**Riesgo en Desarrollo:** Que tan difíciles para el desarrollador.
Iteración Asignada: La iteración (liberación en nuestro proceso) a la que corresponde.	
Descripción: La descripción de la historia, detallando las operaciones del usuario y opcionalmente las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuarios, etc.	

Prototipo de Interfaz: Imagen de cada una de las interfaces relacionadas con la HU.

Tabla 2. Plantilla de historia de usuario.

En la fase de exploración del presente trabajo se identificaron las siguientes historias de usuario, que se describirán a continuación.

1. Autenticar Usuario.
2. Permitir al usuario crear un perfil de trabajo.
3. Gestionar estudiante.
4. Gestionar grupo de trabajo.
5. Gestionar profesor.
6. Preprocesar las tablas y sus campos en la Base de Datos.
7. Realizar análisis de los datos.
8. Permitir visualizar los resultados obtenidos.

A continuación se muestran las tablas que representan cada una de las historias de usuarios definidas para el desarrollo de la aplicación.

Historia de Usuario	
Número: 1	**Nombre:** Autenticar Usuario.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Bajo.
Iteración Asignada: 1.	
Descripción: El usuario introduce su usuario y contraseña. El sistema valida los datos y le permite al usuario acceder al sistema.	
Observaciones: Se ingresan los datos elementales: usuario y contraseña.	
Prototipo de Interfaz: Ver anexos.	

Tabla3. HU_Autenticar Usuario.

Historia de Usuario	
Número: 2	**Nombre:** Permitir al usuario crear un perfil de trabajo.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Medio.
Iteración Asignada: 1.	
Descripción: El profesor accede al sistema y se crea un perfil de trabajo; es decir, crea un grupo.	
Observaciones: Se ingresan los datos elementales para crear el perfil de trabajo.	
Prototipo de Interfaz: Ver anexos.	

Tabla 4. HU_Permitir al usuario crear un perfil de trabajo.

Historia de Usuario	
Número: 3	**Nombre:** Gestionar estudiante.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Medio.
Iteración Asignada: 1.	
Descripción: Esta historia tiene como objetivo: la gestión de los datos de los estudiantes. Debe permitir crear, obtener, eliminar y mostrar los datos de los estudiantes.	
Observaciones: Ingresar datos elementales que permitan crear, obtener, eliminar y mostrar los datos de los estudiantes.	
Prototipo de Interfaz: Ver anexos.	

Tabla 5. HU_Gestionar estudiante.

Historia de Usuario	
Número: 4	**Nombre:** Gestionar grupo de trabajo.

Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Bajo.
Iteración Asignada: 1.	
Descripción: Esta historia tiene como objetivo: gestionar un grupo de trabajo. Debe permitir crear, obtener, eliminar y mostrar un grupo de trabajo.	
Observaciones: Ingresar datos elementales que permitan crear, obtener, eliminar y mostrar los datos de un grupo de trabajo.	
Prototipo de Interfaz: Ver anexos.	

Tabla 6. HU_Gestionar grupo de trabajo.

Historia de Usuario	
Número: 5	**Nombre:** Gestionar profesor.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Bajo.
Iteración Asignada: 1.	
Descripción: Esta historia tiene como objetivo: la gestión de los datos de los profesores. Debe permitir crear, obtener, eliminar y mostrar los datos de los profesores.	
Observaciones: Ingresar datos elementales que permitan crear, obtener, eliminar y mostrar los datos de un profesor.	
Prototipo de Interfaz: Ver anexos.	

Tabla 7. HU_Gestionar profesor.

Historia de Usuario	
Número: 6	**Nombre:** Preprocesar las tablas y sus campos en la Base de Datos.

Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Medio.
Iteración Asignada: 2.	
Descripción: El sistema debe permitir al usuario preprocesar las tablas y campos en Bases de Datos, si tiene los privilegios, con el fin de obtener la estructura adecuada para el algoritmo a utilizar.	
Observaciones: Obtener la estructura adecuada para el algoritmo a utilizar.	
Prototipo de Interfaz: No contiene imagen porque el preprocesamiento de los datos es una acción invisible al usuario.	

Tabla 8. HU_Preprocesar las tablas y sus campos en la Base de Datos.

Historia de Usuario	
Número: 7	**Nombre:** Realizar el análisis de los datos.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Alto.
Iteración Asignada: 3.	
Descripción: Se realiza un análisis de los datos escogidos.	
Observaciones: Obtener datos para posterior análisis.	
Prototipo de Interfaz: No contiene imagen porque la realización de análisis es una acción invisible al usuario.	

Tabla 9. HU_Realizar el análisis de los datos.

Historia de Usuario	
Número: 8	**Nombre:** Permitir visualizar los resultados obtenidos.
Usuario: Profesor	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Medio.

Iteración Asignada: 3.
Descripción: El sistema visualiza los resultados obtenidos, para que el profesor tenga los conocimientos necesarios sobre el recorrido de un estudiante determinado.
Observaciones: Se muestra un porciento de las posibilidades de aprobar o desaprobar el curso
Prototipo de Interfaz: Ver anexos.

Tabla 10. HU_Permitir visualizar los resultados obtenidos.

2.5.2 **Requisitos no funcionales**

Los requisitos no funcionales son cualidades que el producto debe tener. Debe pensarse en estas cualidades como las características del entorno que hacen al producto atractivo, usable, rápido o confiable.

RNF1. Usabilidad

En la aplicación se garantizará un acceso fácil y rápido. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora en sentido general.

RNF2. Rendimiento

Garantizará tiempos de respuestas y velocidad de procesamiento de la información generalmente rápidos.

RNF3. Políticos Culturales

El sistema solo podrá ser utilizado en territorio cubano. El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza. El sistema debe estar diseñado en el lenguaje español.

RNF4. Seguridad

El usuario debe autenticarse antes de entrar al sistema, para garantizar el acceso controlado a la información. La información que se maneje en el sistema estará protegida de acceso no autorizado y divulgación. Además será objeto de cuidadosa protección contra corrupción y estados inconsistentes, de igual manera el origen y autoridad de los datos.

2.6 Fase de Planificación

En la fase de Planificación el cliente establece la prioridad de cada historia de usuario, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas y a partir de allí se define el cronograma.

La fase de planeamiento toma un par de días. El cronograma fijado en la etapa de planeamiento se realiza a un número definido de iteraciones. Cada iteración toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo seleccionando las historias que construirán la estructura para el sistema completo. El cliente decide las historias de usuario que se seleccionarán para cada iteración. Al final de la última iteración el sistema está listo para producción.

Las estimaciones de esfuerzo asociadas a la implementación de las historias de usuario se establecen utilizando como medida el punto, el cual equivale a una semana ideal de programación. Las historias de usuario generalmente valen de 1 a 3 puntos. Por otra parte, se mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias de usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias de usuario. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinando cuántos puntos se pueden completar. Al planificar según el alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación (46).

2.6.1 **Estimación de esfuerzos por Historias de Usuario**

Las estimaciones de esfuerzo para implementar las historias de usuario permiten tener una medida bastante real de la velocidad de progreso del proyecto y brindan una guía razonable a la cual ajustarse. Los resultados estimados se exhiben seguidamente:

No.	Historias de Usuario	Puntos estimados

HU_1	Autenticar Usuario.	1
HU_2	Permitir al usuario crear un perfil de trabajo.	1
HU_3	Gestionar estudiante.	1
HU_4	Gestionar grupo de trabajo.	1
HU_5	Gestionar profesor.	1
HU_6	Preprocesar las tablas y sus campos en la Base de Datos.	1
HU_7	Realizar análisis de los datos.	1
HU_8	Permitir visualizar los resultados obtenidos.	1
	Total	8

Tabla 11. Estimación de esfuerzos por Historias de Usuario.

2.7 Plan de Iteraciones

Después de ser descrita e identificadas las historias de usuario y estimado el esfuerzo propuesto para la realización de cada una de ellas, el siguiente paso es especificar cuáles historias de usuario serán implementadas para cada iteración del sistema.

Iteración 1

En esta iteración se implementarán las historias de usuario que por el grado de importancia que tienen para el cliente, tienen prioridad respecto a las otras. Al finalizar esta iteración se contará con las funcionalidades de autenticar usuario, permitir crear un perfil de trabajo, y la gestión de los datos de los estudiantes, de un grupo de trabajo y los profesores. Esta versión tiene como principal objetivo mostrarle al cliente cómo va quedando la aplicación, para comprobar el grado de aceptación que tiene el producto y constatar que es exactamente lo que ha requerido.

Iteración 2.

En esta iteración se tendrá en cuenta la implementación de las funcionalidades que no fueron tratadas en la primera iteración. Al término de esta, se tendrán implementada la funcionalidad reflejada en la historia de usuario referente al pre-procesamiento de las tablas y sus campos en la Base de Datos.

Iteración 3.

En esta iteración se implementarán las funcionalidades restantes que no fueron desarrolladas en las iteraciones anteriores. Con la culminación de las mismas se tendrán implementadas las peticiones del cliente referentes a: realizar el análisis de los datos y visualizar los resultados obtenidos luego del análisis. Al terminar esta iteración se contará con la versión 1.0 del producto final.

2.7.1 Plan de duración de las iteraciones

En esta sección se presentará el plan de duración de iteraciones. Este plan tiene como finalidad mostrar por cada iteración la duración y el orden en que serán implementadas las historias de usuario.

Iteración	Orden de las historias de usuario a implementar	Duración total de la iteración
1ra	✓ Autenticar Usuario. ✓ Permitir al usuario crear un perfil de trabajo ✓ Gestionar estudiante. ✓ Gestionar grupo de trabajo. ✓ Gestionar profesor.	9 semanas
2da	✓ Preprocesar las tablas y sus campos en la Base de Datos.	3 semanas
3ra	✓ Realizar análisis de los datos. ✓ Permitir visualizar los resultados obtenidos.	5 semanas
Total Semanas		17

Tabla 12. Plan de duración de las iteraciones.

2.7.2 Plan de entregas

Para facilitar la elaboración del plan de entregas para la fase de implementación, se acoplaron las funcionalidades referentes a un mismo tema en módulos. Estos módulos también facilitarán el trabajo del programador, ya que estos contribuyen a la organización del trabajo y a evitar la repetición innecesaria de código.

Módulos	Historias de Usuario que agrupa
Administración	Autenticar Usuario. Permitir al usuario crear un perfil de trabajo Gestionar estudiante. Gestionar grupo de trabajo. Gestionar profesor.
Preprocesamiento y Filtrado de Datos	Preprocesar las tablas y sus campos en la Base de Datos.
Análisis	Realizar análisis de los datos.
Visualización	Generar reportes Permitir visualizar los resultados obtenidos.

Tabla 13. Composición de módulos.

Módulos	**Iteración1** (3ra semana de marzo)	**Iteración2** (3ra semana de abril)	**Iteración3** (3ra semana de mayo)
Versión	0.1	0.2	1.0

Tabla 14. Plan de duración de entregas.

Conclusiones

En este capítulo se realizaron todos los pasos del proceso de clasificación de los datos hasta llegar a la obtención de un modelo basado en el algoritmo Naive de Bayes. Al evaluar el modelo, utilizando la técnica de evaluación stratified 10-fold cross validation, se obtuvo un porciento de certeza superior al 84%; de las 26 instancias mal clasificadas solo 10 eran falsos positivos, es decir, estudiantes desaprobados clasificados como aprobados. Por lo que puede concluirse que el desempeño del modelo obtenido es muy bueno.

Capítulo 2: Características del sistema

Se abordaron las peculiaridades de las fases de Exploración y Planificación y los artefactos que se generaron durante su desarrollo, entre ellos las historias de usuario y los planes de entregas e iteraciones. Ambas fases se repiten en cada iteración lo que posibilita realizar una estimación más exacta y real del esfuerzo necesario para cumplir con las historias de usuario negociadas y con las que el equipo de trabajo se ha comprometido.

Capítulo 3: Diseño e implementación del sistema

Diseño e implementación del sistema

Introducción.

En este capítulo se describen las fases de diseño e implementación desarrollada por la metodología XP, donde se detallan las tareas generadas por cada Historia de Usuario definida durante la fase de Planificación y se representa el patrón utilizado..

3.1 Diseño del sistema

XP propone realizar diseños simples y sencillos, hacerlo todo lo menos complejo posible para lograr que sea entendible e implementable. Realizar una correcta especificación de los nombres de métodos y clases, ayuda a comprender mejor lo diseñado y facilita las posteriores ampliaciones y la reutilización del código. Nunca se debe añadir funcionalidades extras al software aunque se piense que serán factibles en el futuro.

3.2 Tarjetas CRC

Las tarjetas CRC (Clase-Responsabilidades-Colaboración) representan objetos. La clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

Las tarjetas CRC se dividen en tres secciones: nombre de la clase, colaboradores (otras clases con las que trabaja en conjunto para llevar a cabo sus funcionalidades) y las responsabilidades (lo que la clase sabe o hace).

A continuación las tarjetas CRC agrupadas por clase:

Clase: Grupo de trabajo	
Responsabilidades	**Colaboraciones**

✓ Crear grupo de trabajo. ✓ Obtener grupo de trabajo. ✓ Eliminar grupo de trabajo. ✓ Modificar grupo de trabajo.	✓ Estudiante. ✓ Profesor. ✓ Controladora.

Tabla 15. Tarjeta CRC Grupo de trabajo.

Clase: Estudiante	
Responsabilidades	**Colaboraciones**
✓ Crear estudiante. ✓ Obtener datos del estudiante. ✓ Eliminar estudiante. ✓ Modificar datos del estudiante.	✓ Grupo de trabajo. ✓ Controladora.

Tabla 16. Tarjeta CRC Estudiante.

Clase: Profesor	
Responsabilidades	**Colaboraciones**
✓ Crear profesor. ✓ Obtener datos del profesor. ✓ Eliminar profesor. ✓ Modificar datos del profesor.	✓ Grupo de trabajo. ✓ Controladora.

Tabla 17. Tarjeta CRC Profesor.

Clase: Datos socio demográficos	
Responsabilidades	**Colaboraciones**

✓ Realizar consulta insertar datos. ✓ Realizar consulta eliminar datos. ✓ Realizar consulta actualizar datos.	✓ Estudiante. ✓ Controladora.

Tabla 18. Tarjeta CRC Datos socio demográficos.

Clase: Dirección	
Responsabilidades	**Colaboraciones**
✓ Insertar datos de la dirección. ✓ Eliminar datos de la dirección. ✓ Actualizar datos de la dirección.	✓ Datos socio demográficos. ✓ Controladora.

Tabla 19. Tarjeta CRC Dirección.

Clase: Histórico del estudiante	
Responsabilidades	**Colaboraciones**
✓ Insertar datos históricos del estudiante. ✓ Eliminar datos históricos del estudiante. ✓ Actualizar datos históricos del estudiante.	✓ Estudiante. ✓ Controladora.

Tabla 20. Tarjeta CRC Histórico del estudiante.

Clase: Evaluación	
Responsabilidades	**Colaboraciones**
✓ Crear evaluación. ✓ Obtener evaluación. ✓ Mostrar tipo de evaluación. ✓ Mostrar fecha de evaluación.	✓ Estudiante. ✓ Controladora.

Tabla 21. Tarjeta CRC Evaluación.

Clase: Encuentro	
Responsabilidades	**Colaboraciones**
✓ Insertar datos del encuentro. ✓ Eliminar datos del encuentro. ✓ Actualizar del encuentro.	✓ Estudiante. ✓ Controladora.

Tabla 22. Tarjeta CRC Encuentro.

Clase: Conexión	
Responsabilidades	**Colaboraciones**
✓ Conectar a la Base de Datos. ✓ Desconectar de la Base de Datos. ✓ Realizar consulta insertar. ✓ Realizar consulta eliminar. ✓ Realizar consulta actualizar. ✓ Realizar consulta seleccionar.	✓ Controladora.

Tabla 23. Tarjeta CRC Conexión.

Clase: Preprocesamiento	
Responsabilidades	**Colaboraciones**
✓ Construir dataset para el análisis.	✓ Controladora. ✓ Preprocesamiento.

Tabla 24. Tarjeta CRC Preprocesamiento.

Clase: Predicción de nota final

Responsabilidades	Colaboraciones
✓ Aplicar modelo a un conjunto de datos. ✓ Mostrar resultado de predicción.	✓ Controladora.

Tabla 25. Tarjeta CRC Predicción de nota final.

Clase: Controladora	
Responsabilidades	**Colaboraciones**
✓ Gestionar profesor. ✓ Gestionar estudiante. ✓ Gestionar grupo de trabajo. ✓ Realizar análisis de los datos. ✓ Obtener predicción.	✓ Profesor. ✓ Estudiante. ✓ Grupo de trabajo. ✓ Preprocesamiento. ✓ Conexión. ✓ Predicción de nota final.

Tabla 26. Tarjeta CRC Controladora.

3.3 Modelo de Datos

Un modelo de datos es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de reglas de integridad, las cuales definen un conjunto de estados consistentes. El cual puede ser usado como una herramienta para especificar los tipos de datos y la organización de los mismos. Además para la manipulación de consultas y datos, así mismo es el elemento clave en el diseño de la arquitectura de un manejador de BD. Este modelo representa la realidad a través de un esquema gráfico empleando la terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo (47).

Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente, que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo.

A continuación se muestra el modelo de datos utilizado:

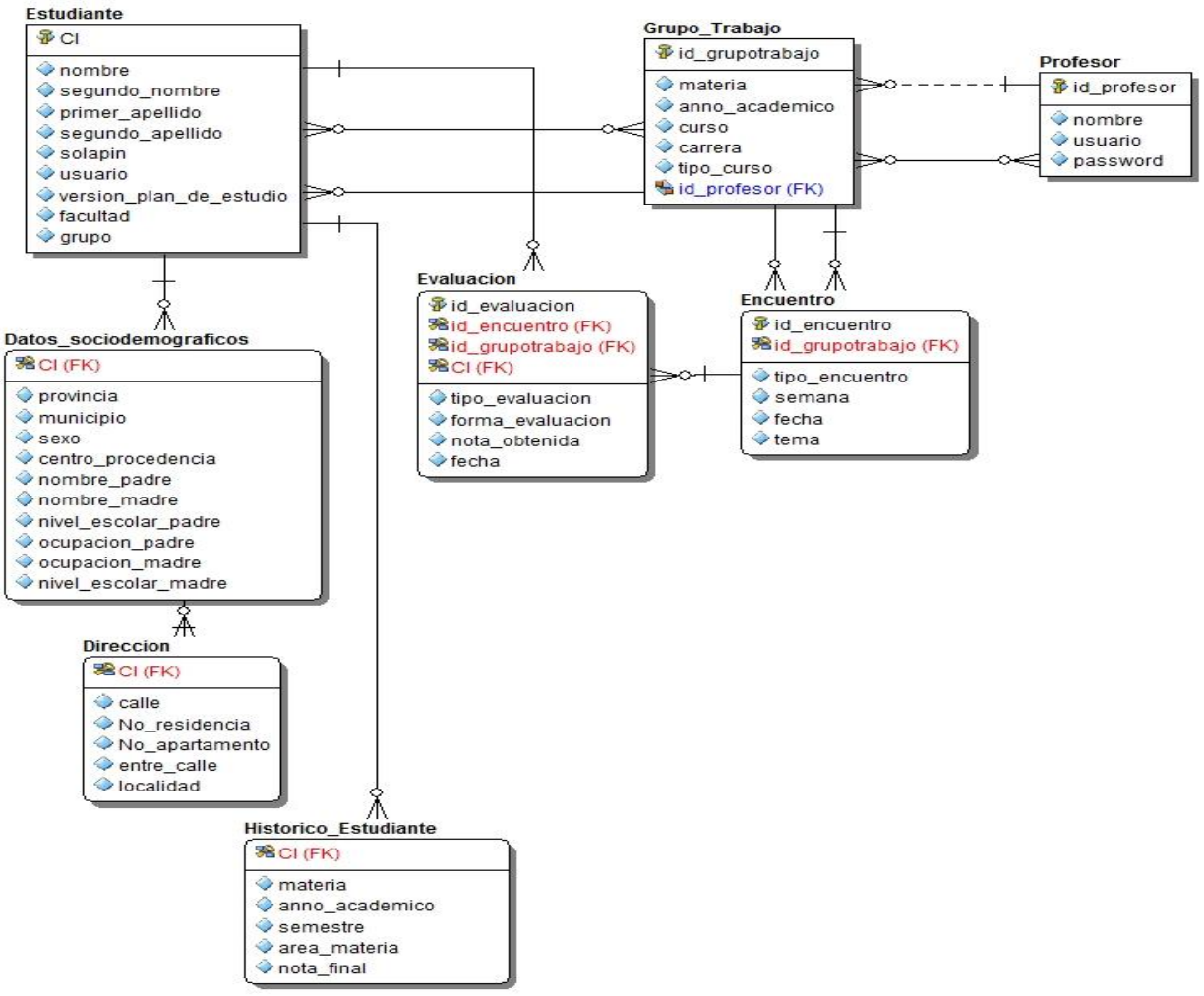


Figura 7: Diagrama Modelo de Datos.

3.4 Descripción de la arquitectura

La arquitectura es el conjunto de decisiones significativas sobre la organización de un sistema software. Incluye la selección de elementos estructurales y las interfaces mediante las que se conectan, la organización a gran escala de los elementos estructurales y la topología de su conexión, su comportamiento en las colaboraciones entre dichos elementos, los mecanismos importantes de que se dispone en el sistema y el estilo arquitectónico que guía su organización (48).

En otras palabras una arquitectura es esencial para el éxito o el fracaso de un proyecto, proporciona una visión global del sistema a construir, es una vista estructural de alto nivel que define el estilo arquitectónico o la combinación de ellos para la solución de un problema.

De acuerdo con la IEEE la arquitectura es:

- El nivel conceptual más alto de un sistema en su ambiente.
- La organización fundamental de un sistema descrita en:
  - ✓ Sus componentes.
  - ✓ Relación entre ellos y con el ambiente.
  - ✓ Principios que guían su diseño y evolución.

El sistema propuesto cuenta con el patrón de **Arquitectura en Capas**, este permite la reutilización de las capas, facilita la estandarización y la modularización del software y elimina las dependencias entre las capas, lo que quiere decir que los cambios aplicados sobre una capa, no afectan las demás.

Partir un sistema en capas tiene una cantidad importante de beneficios:

- ✓ Se puede entender una capa como un todo sin considerar las otras.
- ✓ Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos.
- ✓ Se minimizan dependencias entre capas.
- ✓ Las capas posibilitan la estandarización de servicios.
- ✓ Luego de tener una capa construida, puede ser utilizada por muchos servicios de mayor nivel. (49)

La siguiente figura muestra un esquema básico de una arquitectura siguiendo este patrón.

Figura 8: Arquitectura por Capas.

Presentación

Se refiere a la interacción entre el usuario y el software. Su principal responsabilidad es mostrar información al usuario, interpretar los comandos de este y realizar algunas validaciones simples de los datos ingresados (49). Se evidencia dentro de la clase jFrameInicio, que es una interfaz visual que se le muestra al usuario para que el mismo interactúe con la aplicación.

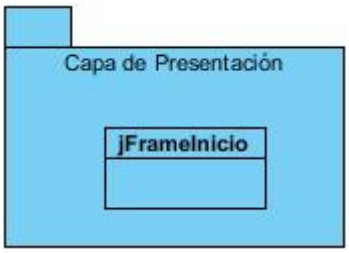


Figura 9: Capa de presentación.

Negocios

Esta capa contiene la funcionalidad que implementa la aplicación. Involucra cálculos basados en la información dada por el usuario y datos almacenados y validaciones. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica en la capa de negocios para su uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios, que coordina la conversación con los clientes del servicio e invoca cualquier flujo o componente de negocio (49). Se evidencia dentro de la clase Controladora, que realizan llamadas a las demás clases para obtener los datos y pasarlos a la capa de presentación para que los muestre al usuario. Responde a eventos, usualmente acciones del usuario e invoca cambios en la capa de datos y probablemente en la de presentación.

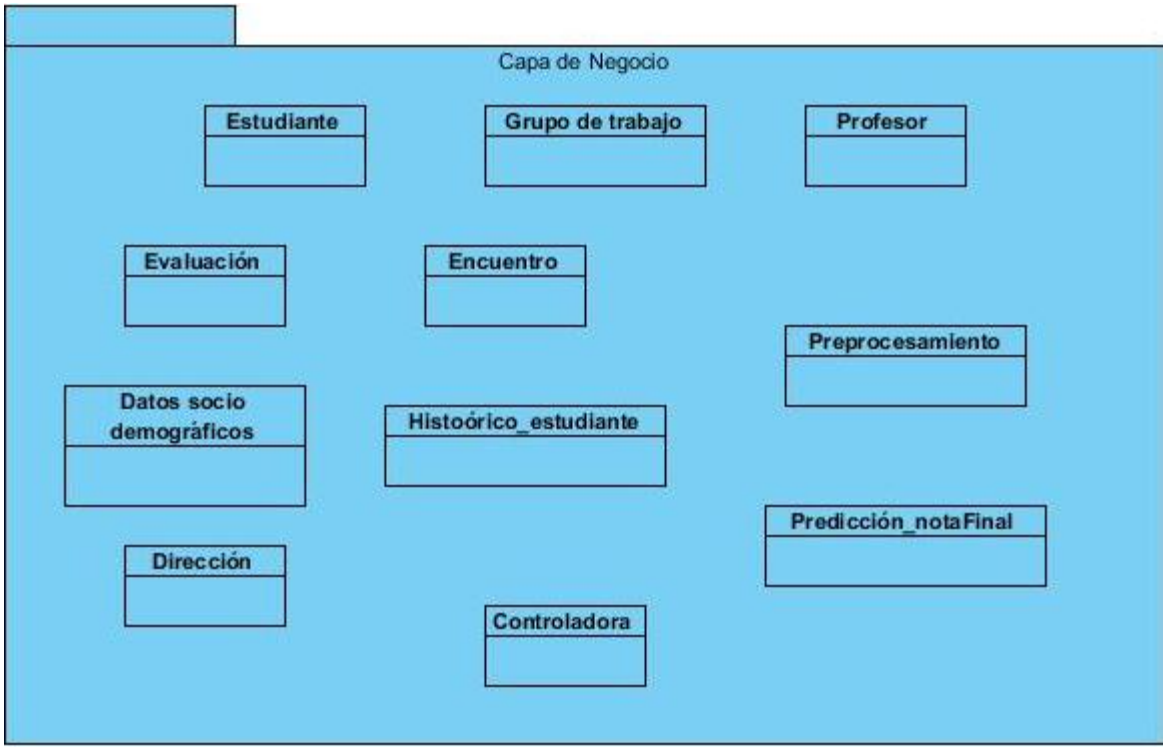


Figura 10: Capa de negocio.

Datos

Esta capa es la encargada de hacer persistir las entidades que se manejan en el negocio, el acceso a los datos almacenados, la actualización, y otras operaciones sobre ellos, aunque puede ofrecer servicios relacionados con la persistencia o recuperación de información más complejos (50). En esta capa se contienen todas las clases que tienen el código relacionado con el acceso a datos, para que este sea lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. Se incluirán consultas a las bases de datos y validaciones de entrada de datos de donde se obtendrán y se devolverán resultados para luego ser procesados. Las clases que pertenecen a este caso se muestran en la figura que se muestra a continuación.

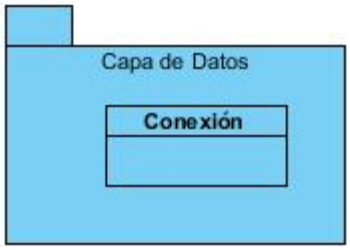


Figura 11: Capa de datos

3.5 Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Debe ser reutilizable, o sea que debe ser aplicable a diferentes problemas de diseño en distintos contextos. Existen varias clasificaciones de patrones, como por ejemplo los patrones GRASP[6] que se utilizan para la asignación de responsabilidades. Su uso es fundamental para un correcto diseño del software. Ayudan a la reutilización de diseño gráfico, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. Lo que supone una gran ventaja ya que disminuye los esfuerzos de desarrollo y mantenimiento, mejora la seguridad informática, eficiencia y consistencia del diseño, y proporciona un inmenso ahorro en la inversión. También ayudan a tener un software más flexible, modular y extensible (51). A continuación se describen los patrones utilizados:

Experto: Se encarga de asignar la responsabilidad al experto en la información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, lo que provee un bajo nivel de acoplamiento. Promueve clases sencillas y cohesivas que son más fáciles de mantener y comprender. El uso de este patrón se visualiza en la clase Grupo de trabajo, la cual gestiona las operaciones que conciernen a las funciones de crear, obtener y eliminar un grupo de trabajo. Además ésta posee los datos necesarios y contiene la información que se precisa para realizar cualquier representación en la interfaz de visualización.

Controlador: La aplicación del patrón Controlador consiste en asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente: el negocio, la organización global, o el

[6] Patrones generales de software para asignar responsabilidad.

sistema global. El uso de este patrón se evidencia en la clase Controladora que para evitar saturarse con demasiadas responsabilidades le asigna las mismas a otras clases.

Alta cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme. Fomenta la reutilización, mejorando la claridad y facilidad del diseño. La herramienta destinada a la predicción de posibles estudiantes con problemas docentes sigue los principios de alta cohesión ya que no se hace uso de las clases saturadas de métodos, y que las clases se encuentran agrupadas por funcionalidades, lo que las hace fácilmente reutilizable. El patrón alta cohesión se aprecia en la clase Predicción de nota final ya que es la que cuenta con la responsabilidad de mostrar el resultado de predicción y colaborar con otras clases para llevar a cabo las tareas de visualización.

3.6 Fase de Implementación

Esta fase en la metodología XP también es conocida como Fase de Iteraciones, la cual incluye varias iteraciones sobre el sistema antes de ser entregado, donde se desarrolla en cada iteración el conjunto de historias de usuario que se han seleccionado para la misma. Cada una de ellas se clasifica en tareas de ingeniería que vendrían a considerarse como las entradas de trabajo para cada equipo de programadores.

3.7 Tareas generales de la implementación (TI)

Para la implementación del sistema se llevan a cabo una serie de tareas que no se encuentran comprendidas en las Historias de Usuarios y que se han definido como tareas generales a realizar. Las tareas de la implementación (tareas de ingeniería) son muy importantes para el programador porque guían el proceso de desarrollo del sistema.

Iteración	Historia de Usuario	Tareas de implementación
1ra	Autenticar usuario.	✓ Introducir usuario y contraseña. ✓ Comprobar datos de usuario. ✓ Permitir o denegar el acceso a la aplicación.

1ra	Permitir al usuario crear un perfil de trabajo.	✓ Pedir datos para crear un profesor.
1ra	Gestionar estudiantes.	✓ Crear estudiante. ✓ Obtener datos del estudiante. ✓ Eliminar estudiante. ✓ Modificar datos del estudiante.
1ra	Gestionar grupo de trabajo.	✓ Crear grupo de trabajo. ✓ Obtener grupo de trabajo. ✓ Eliminar grupo de trabajo. ✓ Modificar grupo de trabajo.
1ra	Gestionar profesor.	✓ Crear profesor. ✓ Obtener datos del profesor. ✓ Eliminar profesor. ✓ Modificar datos del profesor.
2da	Preprocesar las tablas y sus campos en la Base de Datos.	✓ Obtener atributos relevantes.
3ra	Realizar el análisis.	✓ Obtener datos para el análisis.
3ra	Permitir visualizar los resultados obtenidos.	✓ Mostrar resultados del análisis de los datos.

Tabla 27. Tareas de ingeniería divididas por iteración.

3.7.1 **Descripción de las Tareas de ingeniería por Historias de Usuario.**

Tarea de programación

Número de Tarea:1	**Historia de Usuario No:** 1_Autenticar Usuario.
Nombre Tarea: Introducir usuario y contraseña.	
Tipo de Tarea: Desarrollo.	**Puntos Estimados:** 0.3
Fecha Inicio: 4/3/2013	**Fecha Fin:** 4/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos.	
Descripción: Se muestra una página solicitando los datos del usuario para entrar a la aplicación. El usuario introduce los datos necesarios (usuario y contraseña) para acceder a la aplicación.	

Tabla 28. TI Introducir usuario y contraseña.

Tarea de programación	
Número tarea: 2	**Historia de Usuario No:** 1_Autenticar Usuario.
Nombre tarea: Comprobar datos de usuario.	
Tipo de tarea: Desarrollo	**Puntos estimados**: 0.2
Fecha inicio: 6/3/2013	**Fecha fin:** 6/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos.	
Descripción: Se comprueba que los datos del usuario existen en la base de datos del sistema. De no ser así se le notifica al usuario.	

Tabla 29. TI Comprobar datos de usuario**.**

Tarea de programación	
Número tarea: 3	**Historia de Usuario No:** 1_Autenticar Usuario
Nombre tarea: Permitir o denegar el acceso a la aplicación.	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.2
Fecha inicio: 8/3/2013	**Fecha fin:** 8/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	

Descripción: Si los datos de autenticación son incorrectos se le informa al usuario y se le permite volver a insertar los datos requeridos (usuario y contraseña). Si los datos son correctos se le muestra al usuario una interfaz con los menús correspondientes a los privilegios de ese usuario.

Tabla 30. TI Permitir o denegar el acceso a la aplicación.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 2_Permitir al usuario crear un perfil de trabajo.
Nombre tarea: Pedir datos del grupo para crear perfil de trabajo	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.2
Fecha inicio: 11/3/2013	**Fecha fin:** 11/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Se muestra una interfaz en la que el usuario se crea un perfil de trabajo llenando varios campos de información asociados a las características del grupo como es el caso del: número, año, facultad, asignatura, entre otros. Luego se guardaran todos los datos.	

Tabla 31. TI Pedir datos del grupo para crear perfil de trabajo.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 3_Gestionar estudiante.
Nombre tarea: Crear estudiante	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.2
Fecha inicio: 13/3/2013	**Fecha fin:** 13/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite crear un estudiante.	

Tabla 32. TI Crear estudiante.

Tarea de programación

Número tarea: 2	**Historia de Usuario No:** 3_Gestionar estudiante.
Nombre tarea: Obtener datos del estudiante	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 18/3/2013	**Fecha fin:** 18/3/2013
Programador Responsable: : Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite obtener los datos de un estudiante.	

Tabla 33. TI Obtener datos del estudiante.

Tarea de programación	
Número tarea: 3	**Historia de Usuario No:** 3_Gestionar estudiante.
Nombre tarea: Eliminar estudiante	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 21/3/2013	**Fecha fin:** 21/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite eliminar un estudiante.	

Tabla 34. TI Eliminar estudiante.

Tarea de programación	
Número tarea: 4	**Historia de Usuario No:** 3_Gestionar estudiante.
Nombre tarea: Modificar datos del estudiante.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 25/3/2013	**Fecha fin:** 25/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite modificar los datos de un estudiante.	

Tabla 35. TI Modificar datos del estudiante.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 4_Gestionar grupo de trabajo.
Nombre tarea: Crear grupo de trabajo.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 28/3/2013	**Fecha fin:** 28/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite crear un grupo de trabajo.	

Tabla 36. TI Crear grupo de trabajo**.**

Tarea de programación	
Número tarea: 2	**Historia de Usuario No:** 4_Gestionar grupo de trabajo.
Nombre tarea: Obtener grupo de trabajo.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 1/4/2013	**Fecha fin:** 1/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite obtener un grupo de trabajo.	

Tabla 37. TI Obtener grupo de trabajo**.**

Tarea de programación	
Número tarea: 3	**Historia de Usuario No:** 4_Gestionar grupo de trabajo.
Nombre tarea: Eliminar grupo de trabajo.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 4/4/2013	**Fecha fin:** 4/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite eliminar un grupo de trabajo.	

Tabla 38. TI Eliminar grupo de trabajo.

Tarea de programación	
Número tarea: 4	**Historia de Usuario No:** 4_Gestionar grupo de trabajo.
Nombre tarea: Modificar grupo de trabajo.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 8/4/2013	**Fecha fin:** 8/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite modificar un grupo de trabajo.	

Tabla 39. TI Modificar grupo de trabajo.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 5_Gestionar profesor.
Nombre tarea: Crear profesor.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 11/4/2013	**Fecha fin:** 11/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite crear un profesor.	

Tabla 40. TI Crear profesor.

Tarea de programación	
Número tarea: 2	**Historia de Usuario No:** 5_Gestionar profesor.
Nombre tarea: Obtener datos del profesor.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 15/4/2013	**Fecha fin:** 15/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	

Descripción: Permite obtener los datos del profesor.

Tabla 41. TI Obtener datos del profesor.

Tarea de programación	
Número tarea: 3	**Historia de Usuario No:** 5_Gestionar profesor.
Nombre tarea: Eliminar profesor.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 18/4/2013	**Fecha fin:** 18/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite eliminar un profesor.	

Tabla 42. TI Eliminar profesor.

Tarea de programación	
Número tarea: 4	**Historia de Usuario No:** 5_Gestionar profesor.
Nombre tarea: Modificar datos del profesor.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 22/4/2013	**Fecha fin:** 22/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite modificar los datos del profesor.	

Tabla 43. TI Modificar datos del profesor.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 6_Permitir preprocesar las tablas y sus campos en Bases de datos.
Nombre tarea: Obtener atributos relevantes.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3

Fecha inicio: 25/4/2013	**Fecha fin:** 25/4/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Haciendo uso de algunos de los atributos seleccionados, se realizan transformaciones para obtener nuevos atributos con mayor relevancia para el algoritmo.	

Tabla 44. TI Obtener atributos relevantes.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 7_Realizar el análisis
Nombre tarea: Obtener datos para el análisis.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 29/4/2013	**Fecha fin:** 24/3/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite obtener los datos para realizar el análisis.	

Tabla 45. TI Obtener datos para el análisis.

Tarea de programación	
Número tarea: 1	**Historia de Usuario No:** 8_Permitir visualizar los resultados obtenidos.
Nombre tarea: Mostrar resultados del análisis de los datos.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 0.3
Fecha inicio: 2/5/2013	**Fecha fin:** 2/5/2013
Programador Responsable: Itamys Arelis Hodelin Valiente y David Mourlot Matos	
Descripción: Permite mostrar el resultado de los datos luego de un previo análisis realizados.	

Tabla 46. TI Mostrar resultados del análisis de los datos.

3.8 Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes) (52).

Describen la arquitectura física del sistema durante la ejecución, en términos de:

- ✓ procesadores
- ✓ dispositivos
- ✓ componentes de software

Describen la topología del sistema:

- ✓ la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos

En la figura que se muestra a continuación se presenta el diagrama de despliegue propuesto para el sistema.

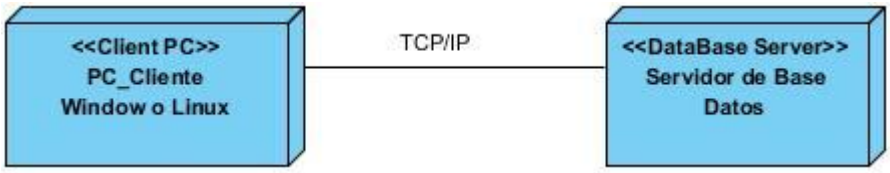


Figura 12: Diagrama de despliegue.

Conclusiones

En este capítulo se abordaron los procesos de diseño e implementación del sistema mediante la metodología ágil XP. Se utilizó como patrón la Arquitectura en Capas, lo que permitió organizar los

componentes del sistema en capas distintas según su misión. Por otro lado, se desarrollaron las tareas correspondientes para dar solución a las historias de usuario definidas para cada una de las iteraciones, lo que garantizó la organización del trabajo para los programadores.

Conclusiones generales

Una vez finalizada la investigación se arribaron a las siguientes conclusiones:

- El algoritmo Naive de Bayes es el más adecuado por su buen desempeño ante situaciones características del área educacional, tales como: conjuntos de datos pequeños o medianos; atributos nominales y numéricos; y datos faltantes o erróneos.

- El modelo obtenido presenta porcentajes de certeza superiores al 80%, por lo que se considera apto para ser aplicado en la predicción de estudiantes con riesgo de desaprobar el examen final de una asignatura.

- El análisis realizado sobre los datos con la herramienta Weka, mostro que el **índice general del estudiante** es la variable de mayor peso a la hora de determinar el aprobado o desaprobado de un estudiante en el examen final; la **calificación** obtenida en la **primera prueba parcial** y en el **primer corte evaluativo** son otros factores importantes, aunque con menor peso.

- Se llevó a cabo la Programación Extrema (XP) en sus 4 primeras fases: Exploración, Planificación, Diseño e Implementación, lo que permitió construir la arquitectura del sistema a desarrollar y elaborar los artefactos generados durante el ciclo de vida del proyecto, así como iniciar la fase de implementación con la mínima cantidad de errores, evitando de esta forma la pérdida de tiempo y recursos. La aplicación de la técnica “Identificación de las historias de usuario (HU)” permitió especificar los requisitos del software y que el programador trazara los objetivos para dar cumplimiento a los mismos.

- Se implementó un sistema de alerta temprana que identifica aquellos estudiantes con alto riesgo de desaprobar el examen final de la asignatura; esto permite a los profesores adoptar las medidas necesarias para corregir la situación. Además, el sistema permite validar la aplicación de la Minería de Datos Educacional al contexto de la UCI.

Conclusiones generales

De esta forma, se ha cumplido con el objetivo y las tareas trazadas en el trabajo de diploma, por lo que se obtuvo un sistema informático, que predice dada la trayectoria actual, aquellos estudiantes con alta probabilidad de presentar problemas docentes.

Recomendaciones

- Obtener para el análisis de los estudiantes datos también del Entorno Virtual de Aprendizaje (EVA), así como también cuestionarios, diagnósticos, entre otros.
- Crear otros modelos aplicando Minería de Datos para agregarlos a la aplicación.
- Vincular los resultados de esta investigación en otros niveles de enseñanza.

Referencias bibliográficas

Referencia Bibliográfica

1. **S. Sumathi, S.N. Sivanandam.** *Introduction to Data Mining and its Applications.* s.l. : Springer-Verlag Berlin Heidelberg , 2006.

2. **Brief, Issue.** *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* USA : s.n., April 10, 2012.

3. *The New Media Consortium (NMC) Horizon Report.* s.l. : Higher Education Edition, 2012.

4. *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* U.S. : Issue Brief, 2012.

5. **Russel y Norving.** *Artificial Intelligence. A modern Approach.*

6. **Kantardzic, Mehmed.** *Data Mining. Concepts, models, method and algorithms.* s.l. : IEEE press, 2003.

7. **Sumathi, S. S.N. Sivanandam.** *Introduction to Data Mining and its Applications.* s.l. : Springer-Verlag Berlin Heidelberg, 2008. págs. 50-62.

8. **Kantardzic, Mehmed.** *Data Mining. Concepts, models, method and algorithms.* s.l. : IEEE press.

9. **Hand, David, Heikki, Mannila y Padhraic, Smyth.** *Principles of Data Mining.*

10. **David, Hand, Heikki, Mannila y Padhraic, Smyth.** *Principles of Data Mining.* s.l. : The MIT Press, 2008.

11. **Witten, Ian H., Frank, Eibe y Hall, Mark A.** *Data Mining Practical Machine Learning Tools and Techniques.* USA : Third Edition, 2011.

12. **Ventura, C. y Romero, S.** *Educational data mining: A survey from 1995 to 2005.* 2006.

13. *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* s.l. : Brief, Issue, April 10, 2012.

Referencias bibliográficas

14. **Cristóbal Romero, Sebastián Ventura y Mykola, Pechen.** New York : CRC Press, 2011.

15. **Leah P., Dawson y Macfadyen, Shane.** *Mining LMS data to develop an "early warning system" for educators: A proof of concept.* 2010.

16. **Cristóbal, Sebastián Romero y Ventura Mykola, Pechen.** *Handbook of Educational Data Mining.* New York : s.n., 2011.

17. **Remón González, Neyvis y Millet, Tutor: MSc. Roberto.** *Análisis para la predicción del éxito o fracaso académico de estudiantes de la Universidad de las Ciencias Informáticas mediante la teoría de conjuntos aproximados.* La Habana : UCI, 2010.

18. **Romero y Ventura.** *Educational data mining.*

19. *Handbook of Educational Data Mining.* New York : s.n., 2011.

20. **ROMÁN COLLAZO, CARLOS ALBERTO y HERNÁNDEZ RODRÍGUEZ, YENIMA.** *Variables psicosociales y su relación con el desempeño académico de estudiantes de primer año de la Escuela Latinoamericana de Medicina.* Cuba : Revista Iberoamericana de Educación.

21. **Þenol Zafer ERDOĐAN Mehpare TÝMOR, JOURNAL OF AERO.** *A DATA MINING APPLICATION IN A STUDENT DATABASE.* julio 2005. NUMBER 2.

22. Metodologías SCRUM y XP. [En línea] 18 de abril de 2013. [Citado el: 15 de junio de 2013.] http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_SCRUM_y_XP.

23. **Penadés, Patricio Letelier y Carmen, Mª.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).*

24. Introducción a UML. [En línea] http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html.

25. **Fernández, Gerardo.** *Introducción a Extreme Programming.*

26. Manual de PHP. [En línea] 20012. http://www.php.net/manual/en/intro-whatcando.php.

Referencias bibliográficas

27. **Mehd iAchour, F.B.,Antony Dovgal, Nuno Lopes, Hann.** Manual PHP. [En línea] 2012. http://www.php.net/manual/en/intro-whatcando.php.

28. **V., Jorge.** Código de Programación. [En línea] 5 de Sep de 2010. http://codigoprogramacion.com/cursos/java/47-introjava.html.

29. EcuRed. [En línea] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n#Ejemplos_de_IDE_de_programaci.C3.B3n.

30. NetBeans. [En línea] https://netbeans.org/.

31. [En línea] [Citado el: 20 de marzo de 2013.] www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado..

32. Aplicaciones.og. [En línea] http://aplicaciones.org/eclipse-ide-de-desarrollo-open-source/.

33. EcuRed. [En línea] http://www.ecured.cu/index.php/Herramienta_CASE.

34. **Nobrega, Maria.** Blogia. [En línea] http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php.

35. EcuRed. [En línea] http://www.ecured.cu/index.php/Visual_Paradigm.

36. **Pecos, Daniel.** PostGreSQL vs. MySQL. [En línea] http://danielpecos.com/docs/mysql_postgres/x57.html.

37. **rafaelma.** PostgreSQL-es. [En línea] 10 de febrero de 2010 - 22:29. http://www.postgresql.org.es/sobre_postgresql.

38. **Frank, Ian H., Witten y Eibe.** *Data Mining: Practical machine learning tools with Java implementation.Morgan Kaufmann.* San Francisco : s.n., 2011.

39. **Cardoso García, Yanet y Pérez Aramillo, Antonio Miguel.** La Habana : Universidad de las Ciencias Informaticas.

Referencias bibliográficas

40. **Cubero, Juan Carlos y Berzal, Fernando.** *Sistemas Inteligentes de Gestión. Guión de Prácticas de Minería de Datos. Práctica 1. Herramientas de Minería de Datos.* Granada : Dpto de Ciencias de la Computación en I.A.

41. **Miranda Ramos, Gianny.** *Componente de software para la firma digital de documentos jurídicos tratados en formato electrónico en el proyecto Tribunales.* La habana : s.n., 2010.

42. **Romero y Ventura.** *Handbook of Educational Data Mining.* 2011.

43. **H, Ian y Witten, Eibe.** *Data Mining: Practical machine learning tools with Java implementation.Morgan Kaufmann.* 2011.

44. **Penadés Patricio, Letelier y Mª Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).*

45. **Jeffries, R., Anderson, A. y Hendrickson, C.** *"Extreme Programming Installed".*

46. **Armstrong, Eric y otros, y.** *Java Web Services Tutorial.* s.l. : Addison Wesley, 2002.

47. **Montano Rodríguez, Yasniel.** *Desarrollo del diseñador y generador de formularios web a partir de un modelo relacional.* La Habana : s.n., 2011.

48. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.*

49. **Parra, Ernesto Marquina y Jose David.** *Guía de Patrones, Prácticas y Arquitectura .NET.* 2008.

50. **Daniel Fernández, Lanvin.** Estudios de Doctorado Avances en Informática - Curso Tecnologías WEB. [En línea]

51. **Saavedra, Jorge.** El mundo informático. [En línea] 8 de mayo de 2007. [Citado el: mayo de 26 de 2013.]

52. Scribd. [En línea] http://es.scribd.com/doc/19808824/diagramas-de-despliegue-2222 ..

53. **Witten, Ian H. y Frank, Eibe.** *Practical Machine Learning Tools and Techniques.* 2011.

Bibliografía

1. **S. Sumathi, S.N. Sivanandam.** *Introduction to Data Mining and its Applications.* s.l. : Springer-Verlag Berlin Heidelberg , 2006.

2. **Brief, Issue.** *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* USA : s.n., April 10, 2012.

3. *The New Media Consortium (NMC) Horizon Report.* s.l. : Higher Education Edition, 2012.

4. *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* U.S. : Issue Brief, 2012.

5. **Russel and Norving.** *Artificial Intelligence. A modern Approach.*

6. **Kantardzic, Mehmed.** *Data Mining. Concepts, models, method and algorithms.* s.l. : IEEE press, 2003.

7. **Sumathi, S. S.N. Sivanandam.** *Introduction to Data Mining and its Applications.* s.l. : Springer-Verlag Berlin Heidelberg, 2008. pp. 50-62.

8. **Kantardzic, Mehmed.** *Data Mining. Concepts, models, method and algorithms.* s.l. : IEEE press.

9. **Hand, David, Heikki, Mannila and Padhraic, Smyth.** *Principles of Data Mining.*

10. **David, Hand, Heikki, Mannila and Padhraic, Smyth.** *Principles of Data Mining.* s.l. : The MIT Press, 2008.

11. **Witten, Ian H., Frank, Eibe and Hall, Mark A.** *Data Mining Practical Machine Learning Tools and Techniques.* USA : Third Edition, 2011.

12. **Ventura, C. and Romero, S.** *Educational data mining: A survey from 1995 to 2005.* 2006.

13. *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics.* s.l. : Brief, Issue, April 10, 2012.

Bibliografía

14. **Cristóbal Romero, Sebastián Ventura and Mykola, Pechen.** New York : CRC Press, 2011.

15. **Leah P., Dawson and Macfadyen, Shane.** *Mining LMS data to develop an "early warning system" for educators: A proof of concept.* 2010.

16. **Cristóbal, Sebastián Romero and Ventura Mykola, Pechen.** *Handbook of Educational Data Mining.* New York : s.n., 2011.

17. **Remón González, Neyvis and Millet, Tutor: MSc. Roberto.** *Análisis para la predicción del éxito o fracaso académico de estudiantes de la Universidad de las Ciencias Informáticas mediante la teoría de conjuntos aproximados.* La Habana : UCI, 2010.

18. **Romero and Ventura.** *Educational data mining.*

19. *Handbook of Educational Data Mining.* New York : s.n., 2011.

20. **ROMÁN COLLAZO, CARLOS ALBERTO and HERNÁNDEZ RODRÍGUEZ, YENIMA.** *Variables psicosociales y su relación con el desempeño académico de estudiantes de primer año de la Escuela Latinoamericana de Medicina.* Cuba : Revista Iberoamericana de Educación.

21. **Þenol Zafer ERDOĐAN Mehpare TÝMOR, JOURNAL OF AERO.** *A DATA MINING APPLICATION IN A STUDENT DATABASE.* julio 2005. NUMBER 2.

22. Metodologías SCRUM y XP. [Online] abril 18, 2013. [Cited: junio 15, 2013.] http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_SCRUM_y_XP.

23. **Penadés, Patricio Letelier and Carmen, Mª.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).*

24. Introducción a UML. [Online] http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html.

25. **Fernández, Gerardo.** *Introducción a Extreme Programming.*

26. Manual de PHP. [Online] 20012. http://www.php.net/manual/en/intro-whatcando.php.

Bibliografía

27. **Mehd iAchour, F.B.,Antony Dovgal, Nuno Lopes, Hann.** Manual PHP. [Online] 2012. http://www.php.net/manual/en/intro-whatcando.php.

28. **V., Jorge.** Código de Programación. [Online] Sep 5, 2010. http://codigoprogramacion.com/cursos/java/47-introjava.html.

29. EcuRed. [Online] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n#Ejemplos_de_IDE_de_programaci.C3.B3n.

30. NetBeans. [Online] https://netbeans.org/.

31. [Online] [Cited: marzo 20, 2013.] www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado..

32. Aplicaciones.og. [Online] http://aplicaciones.org/eclipse-ide-de-desarrollo-open-source/.

33. EcuRed. [Online] http://www.ecured.cu/index.php/Herramienta_CASE.

34. **Nobrega, Maria.** Blogia. [Online] http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php.

35. EcuRed. [Online] http://www.ecured.cu/index.php/Visual_Paradigm.

36. **Pecos, Daniel.** PostGreSQL vs. MySQL. [Online] http://danielpecos.com/docs/mysql_postgres/x57.html.

37. **rafaelma.** PostgreSQL-es. [Online] febrero 10, 2010 - 22:29. http://www.postgresql.org.es/sobre_postgresql.

38. **Frank, Ian H., Witten and Eibe.** *Data Mining: Practical machine learning tools with Java implementation.Morgan Kaufmann.* San Francisco : s.n., 2011.

39. **Cardoso García, Yanet and Pérez Aramillo, Antonio Miguel.** La Habana : Universidad de las Ciencias Informaticas.

Bibliografía

40. **Cubero, Juan Carlos and Berzal, Fernando.** *Sistemas Inteligentes de Gestión. Guión de Prácticas de Minería de Datos. Práctica 1. Herramientas de Minería de Datos.* Granada : Dpto de Ciencias de la Computación en I.A.

41. **Miranda Ramos, Gianny.** *Componente de software para la firma digital de documentos jurídicos tratados en formato electrónico en el proyecto Tribunales.* La habana : s.n., 2010.

42. **Romero and Ventura.** *Handbook of Educational Data Mining.* 2011.

43. **H, Ian and Witten, Eibe.** *Data Mining: Practical machine learning tools with Java implementation.Morgan Kaufmann.* 2011.

44. **Penadés Patricio, Letelier and Mª Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).*

45. **Jeffries, R., Anderson, A. and Hendrickson, C.** *"Extreme Programming Installed".*

46. **Armstrong, Eric and otros, y.** *Java Web Services Tutorial.* s.l. : Addison Wesley, 2002.

47. **Montano Rodríguez, Yasniel.** *Desarrollo del diseñador y generador de formularios web a partir de un modelo relacional.* La Habana : s.n., 2011.

48. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** *El Lenguaje Unificado de Modelado.*

49. **Parra, Ernesto Marquina and Jose David.** *Guía de Patrones, Prácticas y Arquitectura .NET.* 2008.

50. **Daniel Fernández, Lanvin.** Estudios de Doctorado Avances en Informática - Curso Tecnologías WEB. [Online]

51. **Saavedra, Jorge.** El mundo informático. [Online] mayo 8, 2007. [Cited: 26 mayo, 2013.]

52. Scribd. [Online] http://es.scribd.com/doc/19808824/diagramas-de-despliegue-2222 ..

53. **Witten, Ian H. and Frank, Eibe.** *Practical Machine Learning Tools and Techniques.* 2011.

54. [Online] [Cited: marzo 20, 2013.] http://weka.wikispaces.com/Programmatic+Use

Bibliografía

55. [Online] [Cited: marzo 8, 2013.] http://weka.sourceforge.net/doc/

56. [Online] [Cited: marzo 13, 2013.] https://platform.netbeans.org/

57. [Online] [Cited: febrero 26, 2013.] http://docs.oracle.com

58. [Online] [Cited: marzo 20, 2013.] http://rapid-i.com/

59. [Online] [Cited: abril 8, 2013.] http://www.kdnuggets.com/software/suites.htm

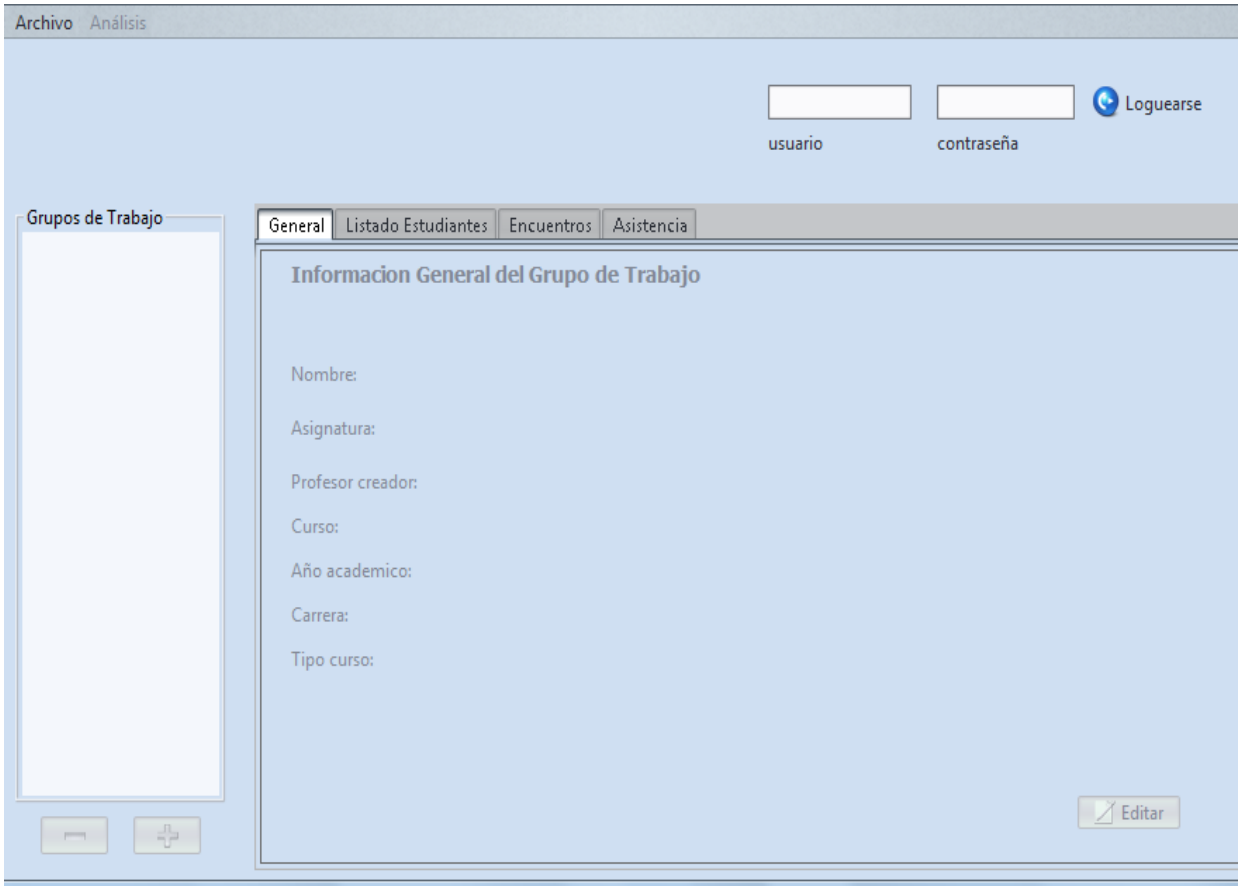
60. [Online] [Cited: abril 23, 2013.] http://msdn.microsoft.com/es-es/library/

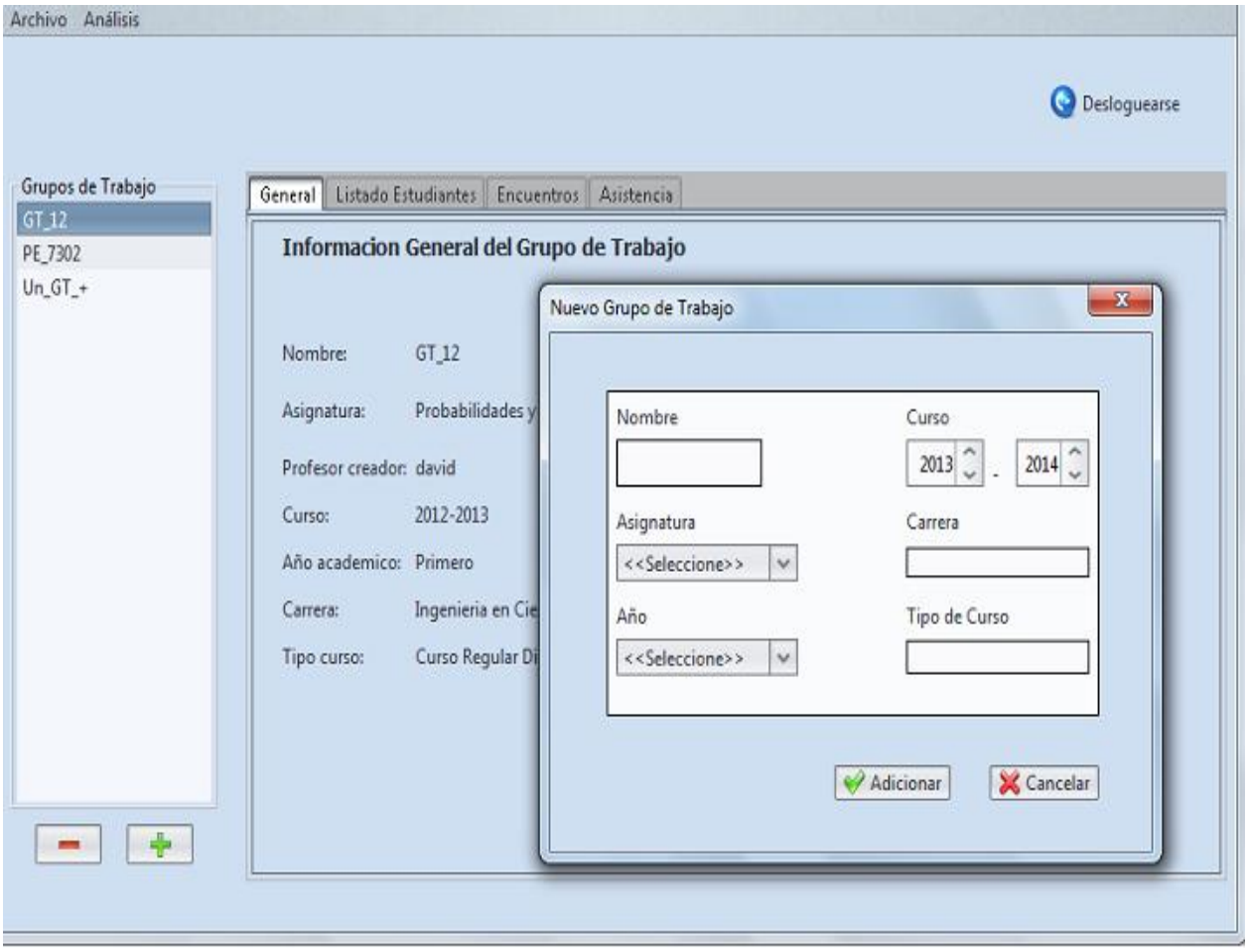
61. [Online] [Cited: mayo 28, 2013.] www.educause.edu/

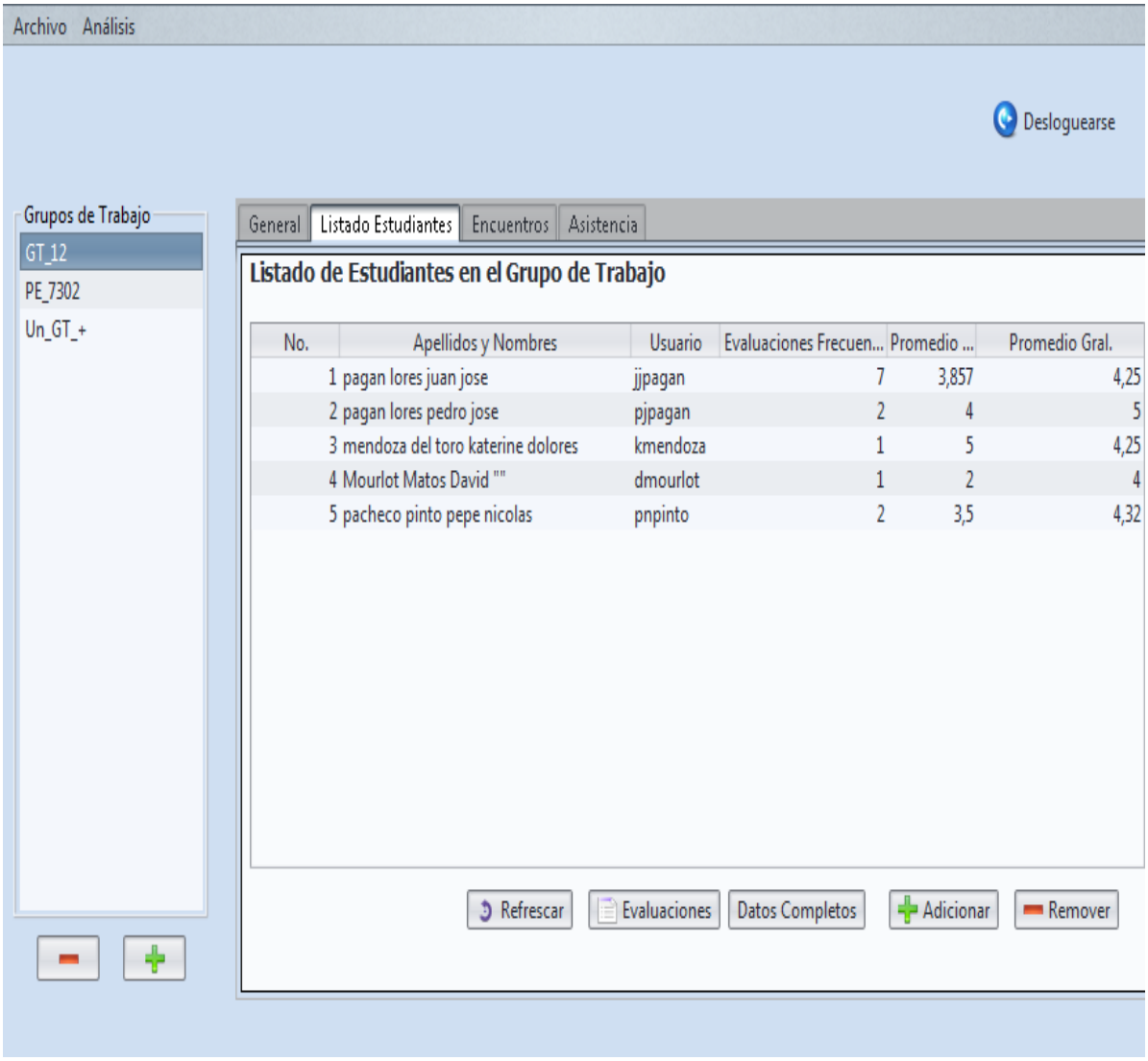
62. [Online] [Cited: mayo 22, 2013.] www.hp.com/hpinfo/grants/catalyst.html

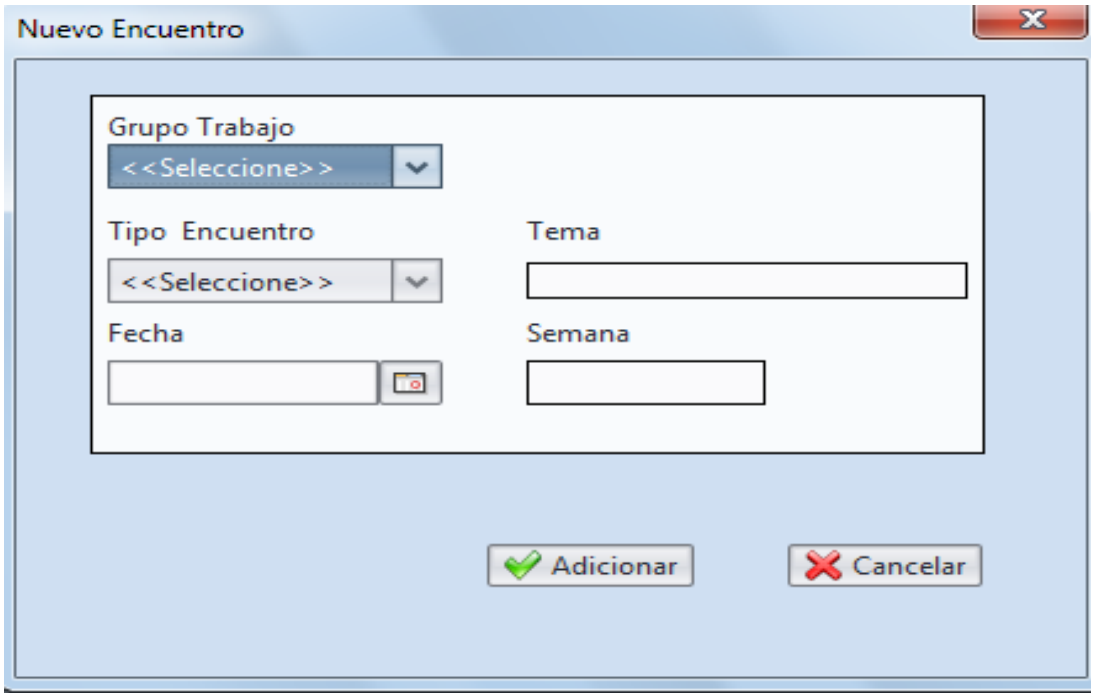
Anexos

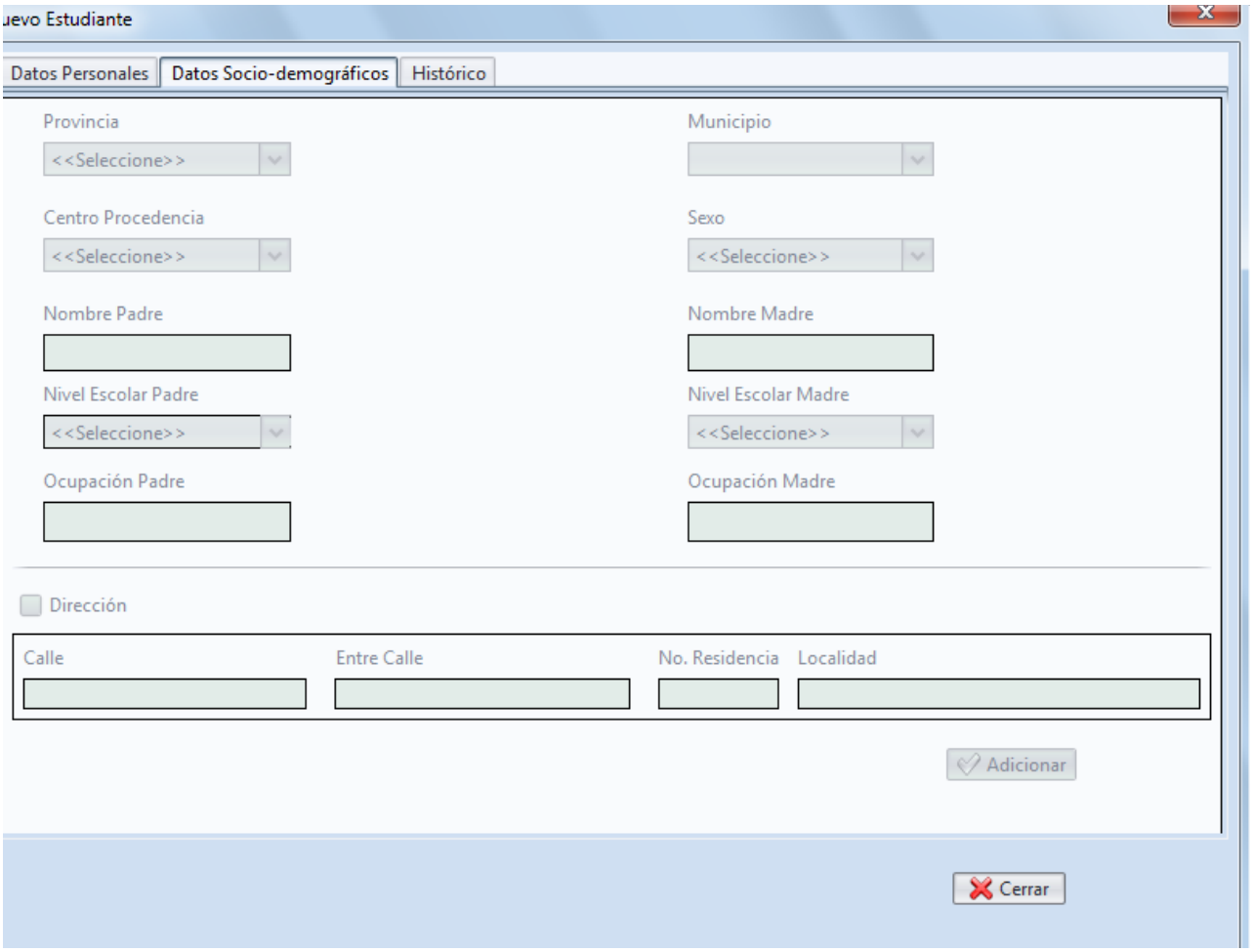
Anexo donde se reflejan las principales interfaces del sistema

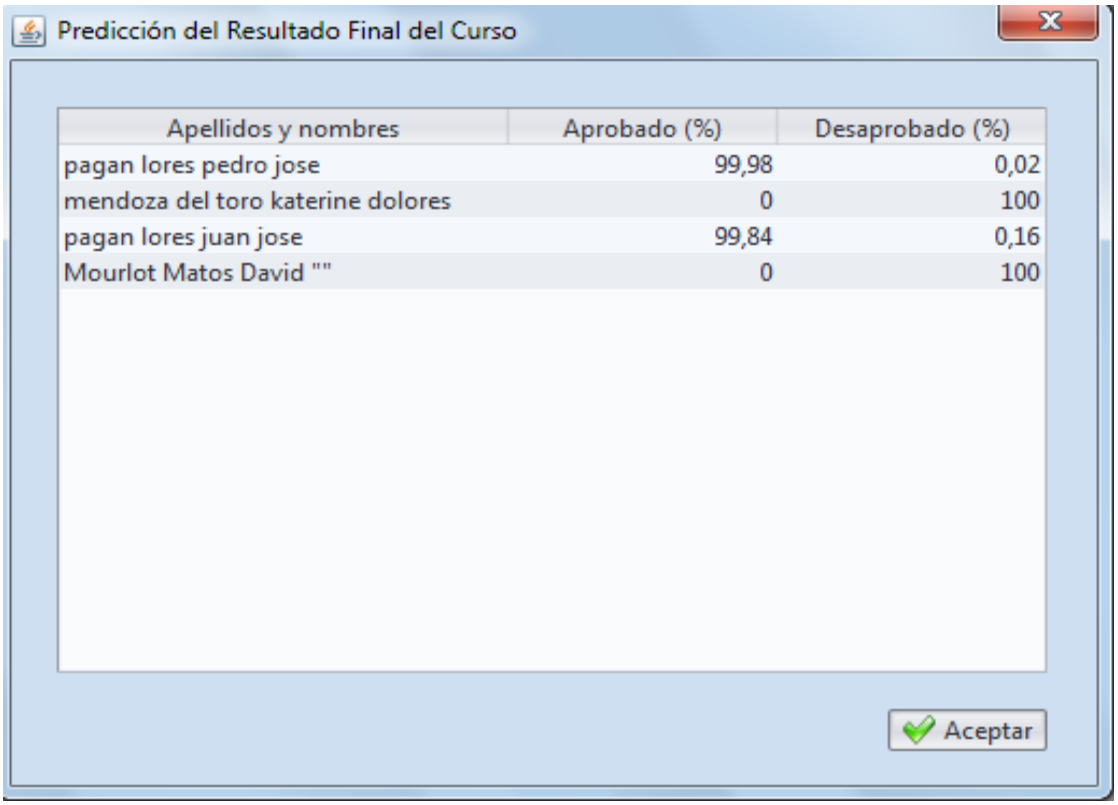



Archivo Análisis
Desloguearse
Grupos de Trabajo
GT_12
PE_7302
Un_GT_+
General
Listado Estudiantes
Encuentros
Asistencia
Informacion General del Grupo de Trabajo
Nombre: GT_12
Asignatura: Probabilidades y
Profesor creador: david
Curso: 2012-2013
Año academico: Primero
Carrera: Ingenieria en Cie
Tipo curso: Curso Regular Di
Nuevo Grupo de Trabajo
Nombre
Curso
2013
2014
Asignatura
<<Seleccione>>
Carrera
Año
<<Seleccione>>
Tipo de Curso
Adicionar
Cancelar


Archivo Análisis
Desloguearse
Grupos de Trabajo
GT_12
PE_7302
Un_GT_+
General
Listado Estudiantes
Encuentros
Asistencia
Listado de Estudiantes en el Grupo de Trabajo
No. Apellidos y Nombres Usuario Evaluaciones Frecuen... Promedio ... Promedio Gral.
1 pagan lores juan jose jjpagan 7 3,857 4,25
2 pagan lores pedro jose pjpagan 2 4 5
3 mendoza del toro katerine dolores kmendoza 1 5 4,25
4 Mourlot Matos David "" dmourlot 1 2 4
5 pacheco pinto pepe nicolas pnpinto 2 3,5 4,32
Refrescar
Evaluaciones
Datos Completos
Adicionar
Remover


Nuevo Encuentro
Grupo Trabajo
<<Seleccione>>
Tipo Encuentro
<<Seleccione>>
Tema
Fecha
Semana
Adicionar
Cancelar


uevo Estudiante
Datos Personales
Datos Socio-demográficos
Histórico
Provincia
<<Seleccione>>
Municipio
Centro Procedencia
<<Seleccione>>
Sexo
<<Seleccione>>
Nombre Padre
Nombre Madre
Nivel Escolar Padre
<<Seleccione>>
Nivel Escolar Madre
<<Seleccione>>
Ocupación Padre
Ocupación Madre
Dirección
Calle
Entre Calle
No. Residencia
Localidad
Adicionar
Cerrar


Predicción del Resultado Final del Curso
Apellidos y nombres
Aprobado (%)
Desaprobado (%)
pagan lores pedro jose
99,98
0,02
mendoza del toro katerine dolores
0
100
pagan lores juan jose
99,84
0,16
Mourlot Matos David ""
0
100
Aceptar