Aplicación para la predicción de resultados en la prueba Saber 11°

Autor

Jorge Iván Durán Páez

Código 200859307

Universidad del Valle

Escuela de Ingeniería de Sistemas y Computación

Ingeniería de Sistemas

Tuluá - Valle

2014

Aplicación para la predicción de resultados en la prueba Saber 11°

Autor

Jorge Iván Durán Páez

jorge.ivan.duran@correounivalle.edu.co

Código 200859307

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Director

Edgar Alberto Molina Rojas

edgarmolina2@gmail.com

Asesor

David Alejandro Przybilla

dav.alejandro@gmail.com

Universidad del Valle

Escuela de Ingeniería de Sistemas y Computación

Ingeniería de Sistemas

Tuluá - Valle

2014

Nota de aceptación

Presidente del Jurado

Jurado 1

Jurado 2

Tuluá, Enero de 2014

Resumen

El proceso de minería de datos, que permite descubrir nuevo y útil conocimiento a partir del análisis exhaustivo de información recolectada en un cierto intervalo del tiempo, puede ser aplicado a distintos contextos de la vida cotidiana. Para el caso de estudio que se desarrolló durante este proyecto, el contexto fue la educación. Específicamente se planteó lograr predecir puntajes de la prueba Saber 11° desarrollada por el Instituto Colombiano para la Evaluación de la Educación (ICFES).

En este trabajo se presenta el desarrollo del proyecto de minería de datos, se usó como fuentes de información las bases de datos del ICFES, las cual contienen información histórica sobre resultados e información personal de los evaluados.

Como resultado final se presenta la aplicación web PrediXaber11, donde las personas pueden acceder y ejecutar consultas. La aplicación usará la información suministrada del evaluado para entregar los posibles resultados en cada una de las áreas académicas evaluadas por el ICFES.

Palabras clave: Minería de datos, Saber 11°, Aplicación Web, KDD, Weka, ICFES, Java.

Dedicatoria

A mis padres, fuente de inspiración y fortaleza en cada uno de los proyectos que emprendo. A mi hermano, un compañero de lucha cuando los obstáculos aparecen y de diversión cuando la alegría nos encara. Y a los muy pocos, pero muy fieles amigos, que en el camino he encontrado. Siempre estarán y siempre estaré.

Jorge

Agradecimientos

A la Universidad de Valle, por su proyecto de ofrecer carreras de nivel profesional en ciudades no capitales. A los profesores que más que ser eso, se convirtieron en compañeros de charlas incansables sobre nuestro entorno académico y profesional.
A compañeros de estudio, presión, preocupación y alegría, que encontré en la universidad y que espero la vida nos encuentre de manera constante.
Cada peldaño pisado para alcanzar la meta, requirió de muchas personas que se hace imposible recordar, así que simplemente: Gracias, a todo el que sienta que aporto a mi carrera.

Índice general

Índice de figuras

Índice de tablas

Lista de anexos

Capítulo 1

Introducción

1.1 Descripción general

En el ámbito de la educación en Colombia, la prueba Saber 11°[1] es de gran importancia para medir la calidad de la enseñanza que se está impartiendo en los colegios del país. La prueba Saber 11° es mayoritariamente presentada por estudiantes de grado 11 de los colegios del país, pero no es exclusiva de estos, cualquier persona puede presentarla, siempre y cuando ya haya obtenido título de bachiller.

Los resultados que se obtienen en la prueba no han presentado mejorías importantes en los últimos 5 años [1–4], causando preocupación por la calidad de la educación media en Colombia.

En [4] se presenta un análisis a los resultados obtenidos por los estudiantes del departamento de Valle del Cauca en las pruebas Saber 5°, 9°[2] y 11°. Para el caso de la prueba Saber 11°, se presentan los resultados obtenidos en los años de 2002 a 2009, una comparación de estos resultados con los del promedio nacional de cada una de las áreas evaluadas y la categorización[3] de los colegios en los años 2010 y 2011.

En las conclusiones y recomendaciones incluidas en [4], se establece la necesidad de brindar una educación que sea pertinente con las deficiencias académicas de los estudiantes, para así poder emprender planes de mejoramiento en las instituciones educativas y reducir los bajos rendimientos en la prueba Saber 11°.

En el presente documento se presenta el desarrollo de la construcción de una aplicación, para

[1] http://www2.icfes.gov.co/examenes/saber-11o

[2] http://www2.icfes.gov.co/examenes/pruebas-saber

[3] ftp://ftp.icfes.gov.co/SABER11/SB11-CLASIFICACION-PLANTELES/Clasificacion%20planteles%20SB11.pdf

lograr determinar cuáles son los posibles resultados que obtendrán los estudiantes que están próximos a presentar la prueba Saber 11° y así conocer las áreas académicas en las cuales pueden presentar deficiencias.

1.2 Problema

1.2.1 Descripción del problema

Para la construcción de una aplicación, que permita conocer previamente como podría ser el puntaje de un estudiante en una o varias de las áreas académicas evaluadas en la prueba Saber 11°, se cuenta con las bases de datos del ICFES[4][5], que almacenan información histórica sobre aspectos personales de los evaluados como por ejemplo: calendario académico del colegio al cual pertenece, carácter académico del colegio, ubicación del colegio (departamento y municipio), valor mensual de la pensión del colegio, si el hogar cuenta con servicio de alcantarillado y recolección de basuras, cantidad de automóviles que poseen en el hogar, si tiene o no computador, cantidad de televisores en el hogar, etnia a la cual pertenece, genero, fecha de nacimiento, si trabaja o no, nivel de educación de los padres, ocupación de los padres, etc. Y además información sobre los puntajes obtenidos por ellos en las áreas académicas evaluadas[6].

El ICFES suministra estas bases de datos en archivos de Access[7], un archivo por cada prueba realizada desde el año 2000 (se realizan 2 pruebas cada año), pero estas bases de datos no son homogéneas, ya que durante las 24 pruebas registradas hasta el momento, la información almacenada de los evaluados no se ha mantenido constante. Durante 12 años la información ha sido recolectada a partir de encuestas en las cuales no siempre se han realizado las mismas preguntas. Por ejemplo, el dato sobre el nivel educativo de los padres se registró durante los años 2000 a 2004, pero no se registró en los años 2005 a 2007 y en 2008 vuelve a registrarse hasta ahora. Además los tipos de datos también se han modificado a lo largo de los años, en la época de 2000 a 2004 el valor "3" indicaba que el nivel de estudio de los padres llegaba a básica primaria, pero desde 2008 hasta ahora existen los valores "9" y "10" que significan "Primaria Completa" y "Primaria Incompleta" respectivamente. Causando esto que dentro de las bases de datos se encuentren muchos valores nulos y una falta de concordancia con los

[4] `ftp://ftp.icfes.gov.co/SABER11/`

[5] Estas bases de datos son de acceso gratuito pero previamente se deben registrar los datos personales en `http://64.76.89.156/index.php/bdicfes/solicitudregistro`, para poder acceder al ftp.

[6] `ftp://ftp.icfes.gov.co/SABER11/SB11-Diccionario_de_Datos-v1-6.pdf`.

[7] `http://office.microsoft.com/es-es/access`

tipos de datos en muchos de los 84 atributos que se almacenan en las 24 bases de datos.

Por consiguiente, en el proceso de construcción de una aplicación que permitiera conocer cómo serían los resultados de un estudiante al momento de presentar la prueba Saber 11°, fue necesario realizar la reestructuración de estas bases de datos, para que su información fuera concordante y no almacenara datos nulos, y así posteriormente utilizarlas como insumo en la construcción de la aplicación.

1.2.2 Formulación del problema

¿Cómo construir una aplicación de software, usando las bases de datos suministradas por el ICFES, que permita conocer previamente los resultados de un estudiante en la prueba Saber 11°?

1.3 Justificación

Los puntajes obtenidos en la prueba Saber 11° son de gran importancia tanto para los estudiantes que la presentan, como para las instituciones educativas y para el gobierno nacional de Colombia, ya que estos sirven como indicador para estimar la calidad de la educación media en el país[8]. Un estudiante que desea continuar con su vida académica ingresando a la educación universitaria, se ve en la necesidad de obtener puntajes que le permitan no solo cumplir con los mínimos puntajes necesarios de inscripción en la carrera de su predilección, sino que también le permitan ingresar a esta en la universidad.

En las instituciones educativas, alcanzar un nivel medio o alto en la clasificación otorgada por el ICFES es de gran importancia para su prestigio dentro de la sociedad académica, los colegios utilizan además estos puntajes como un indicador de autoevaluación para medir la calidad de sus prácticas pedagógicas.

Como se puede observar en la educación media del país, prácticamente todos los involucrados obtienen beneficios si se mejoran los puntajes obtenidos en la prueba Saber 11°. Un primer paso para trabajar en estas mejoras es tener la capacidad de detectar las deficiencias de los estudiantes. Con la aplicación que se propone construir en este documento, se podría dar ese primer paso.

[8] `http://www.icfes.gov.co/examenes/saber-11o/objetivos`

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar una aplicación que permita predecir los puntajes que obtendrá un estudiante en la prueba Saber 11°.

1.4.2 Objetivos específicos

1. Aplicar el proceso de extracción, transformación y carga (Extract, transform and load, ETL) a las bases de datos suministradas por el ICFES.

2. Construir un clasificador, utilizando técnicas de minería de datos, a partir de la información procesada.

3. Implementar una interfaz en donde los usuarios puedan realizar consultas parametrizadas.

1.4.3 Resultados Esperados

En la tabla 1.1 se pueden observar cuales fueron los resultados que se deseaba obtener al momento de cumplir con cada uno de los objetivos planteados.

1.5 Alcance

La presente propuesta establece la construcción de una aplicación que, usando solamente la información almacenada en las bases de datos históricas del ICFES, logre entregar a los usuarios información sobre cómo podrían ser los puntajes que obtendrá un estudiante en cada una de las áreas académicas evaluadas en la prueba Saber 11°.
Para que un usuario pueda conocer el posible resultado de un estudiante, deberá realizar una consulta en donde ingresará datos personales del estudiante, estos datos personales que se deben ingresar serán definidos en el proceso de construcción del clasificador, pero no serán distintos a los registrados en las bases de datos del ICFES.
Después de ser consultada la aplicación, la información que contendrá la respuesta estará constituida por los puntajes que podría obtener un estudiante en cada una de las áreas académicas evaluadas en la prueba Saber 11°: lenguaje, matemáticas, biología, química, física, filosofía, ciencias sociales e ingles.

Objetivo Especifico	**Resultados Esperados**
1	• Bases de datos del ICFES, con estructuras homogéneas y sin datos nulos. • Data mart construido con la información procesada en las bases de datos del ICFES.
2	• Informe sobre algoritmos de clasificación en el proceso de descubrimiento del conocimiento. • Selección del algoritmo de clasificación que mejor se adapta a la cantidad y el tipo de datos con los que se cuenta en el data mart. • Clasificador construido en base a la selección hecha.
3	• Documentación de los procesos de diseño, codificación y pruebas de la interfaz de consultas.

Tabla 1.1: Relación de los objetivos específicos con sus resultados esperados.

El proyecto se desarrollará llevando a cabo la propuesta metodológica presentada por José Hernández en [5].

1.6 Estructura del documento

El documento se encuentra organizado de la siguiente manera:
En el Capítulo 2 se encuentra el marco conceptual y referencial en los que se fundamentó el desarrollo de este trabajo de grado.
En el Capítulo 3 se relaciona como se realizó el proceso de Extracción, Transformación y Carga de las bases de datos recolectadas.
En el Capítulo 4 se presenta el proceso de construcción de los clasificadores utilizados y los resultados obtenidos con cada uno de ellos.
En el Capítulo 5, se involucran los aspectos del desarrollo de la aplicación web construida para que los usuarios puedan realizar sus consultas.
En el Capítulo 6, se concluye el proyecto y se presentan las ideas para posibles trabajos futuros.

Capítulo 2

Marco Referencial

2.1 Marco conceptual

2.1.1 Descubrimiento del conocimiento en bases de datos (KDD)

En [6] se define como:

> Es el proceso de utilizar la información contenida en los sistemas de almacenamiento de datos para identificar patrones significativos, validos, novedosos, potencialmente útiles y comprensibles para el usuario. El proceso global consiste en transformar información de bajo nivel en conocimiento de alto nivel. El proceso KDD es interactivo e iterativo conteniendo los siguientes pasos: comprender el dominio de aplicación, extraer la base de datos objetivo, preparar los datos, minería de datos, interpretación y utilizar el conocimiento descubierto

.

2.1.2 Minería de datos

En [6] se define como:

> Es un campo interdisciplinar con el objetivo general de predecir las salidas y revelar relaciones en los datos. Las tareas propias de la minería de datos pueden ser descriptivas, (i.e. descubrir patrones interesantes o relaciones describiendo los datos), o predictivas (i.e. clasificar nuevos datos basándose en los anteriormente disponibles). Para ello se utilizan herramientas automáticas que emplean algoritmos sofisticados para descubrir principalmente patrones ocultos, asociaciones,

anomalías, y/o estructuras de la gran cantidad de datos almacenados en los Data Warehouses u otros repositorios de información, y filtran la información necesaria de las grandes bases de datos

.

2.1.3 ICFES

Instituto Colombiano para la Evaluación de la Educación, entidad especializada en ofrecer servicios de evaluación de la educación en todos sus niveles, y en particular apoyar al Ministerio de Educación Nacional en la realización de los exámenes de Estado y en adelantar investigaciones sobre los factores que inciden en la calidad educativa, para ofrecer información pertinente y oportuna para contribuir al mejoramiento de la calidad de la educación en Colombia [7].

2.1.4 Prueba Saber 11°

Antes conocido como Examen del ICFES, es un examen de estado que evalúa a los estudiantes que están terminando su ciclo de educación media. La prueba tiene como finalidad apoyar los procesos de selección y admisión que realizan las instituciones de educación superior. Además de este propósito, la prueba busca:

- Brindar al estudiante información que contribuya a la selección de su opción profesional.
- Proporcionar información a las instituciones de educación básica y media sobre el desempeño de los estudiantes.
- Contribuir al desarrollo de estudios de tipo cultural, social y educativo.
- Servir de criterio para otorgar beneficios educativos.

Las áreas académicas evaluadas actualmente por la prueba Saber 11° son:

- Lenguaje
- Matemáticas
- Biología
- Química

- Física
- Filosofía
- Ciencias sociales
- Inglés
- COMPONENTE FLEXIBLE (solo se presenta una de las siguientes opciones)
  - Profundización en lenguaje
  - Profundización en matemáticas
  - Profundización en biología
  - Profundización en ciencias sociales
  - Interdisciplinar violencia y sociedad
  - Interdisciplinar medio ambiente

2.1.5 Data Warehouse

Es un repositorio de datos operacionales seleccionados y adaptados subjetivamente, que puede responder consultas de tipo ad hoc, estadísticas o analíticas. Está situado en el centro de los sistemas de apoyo a la toma de decisiones de una organización y contiene datos históricos, resumidos y detallados de esta. Es esencial para una inteligencia de negocios efectiva, para la formulación e implementación de estrategias donde la gran cantidad de datos requieren ser procesados de manera rápida para comprender su significado e impacto. Permite una fácil organización y mantenimiento de los datos para una rápida recuperación y análisis de la manera en que sean requeridos [8].

2.1.6 Data Mart

Desde un Data Warehouse, los datos fluyen hacia los departamentos de la organización, que personalizan la información almacenada para que sea útil en los sistemas de información de cada uno de ellos. Estos componentes individuales con características personalizadas son conocidos como Data Mart. En otras palabras, es un cuerpo de datos de un sistema de información, que posee la estructura fundamental de un Data Warehouse. Es una subsección de un Data Warehouse y más popular que este [8].

2.2 Antecedentes o Estado del Arte

En el área de la educación, los trabajos presentados a continuación se orientaron a encontrar posibilidades de bajos rendimiento académicos y así lograr conocer perfiles o factores que hacen que los estudiantes no logren las metas propuestas en sus estudios. Este trabajo de grado también se orientó a encontrar estas posibilidades de bajos rendimientos, pero no aplicado a estudiantes de universidades, sino a los estudiantes que presentan la prueba Saber 11°.

También se observó la variedad de algoritmos utilizados para construir los clasificadores, en algunos trabajos se utilizó más de un algoritmo para lograr encontrar comparaciones de rendimiento y mejores niveles de precisión con cada algoritmo. En este trabajo se evaluaron distintos algoritmos para la construcción de clasificadores, con el fin de encontrar información sobre cómo se comportan con la cantidad de datos usados y con la estructura que posean estos datos.

En este trabajo se utilizaron bases de datos que contenían más de 5 millones de registros[1], recolectados a lo largo de 11 años y que presentan una gran variación en la integridad de los atributos de estos registros. Por eso este trabajo centró su primera parte en la homogeneización de estas bases de datos. En los trabajos revisados, la cantidad de datos utilizados en los procesos de investigación, no eran tantos como los que se utilizaron en este trabajo.

En conclusión los trabajos aquí presentados sirvieron de base para la realización de este trabajo, ya que sus fases de investigación fueron las mismas en la mayoría de los trabajos, iguales a las que se aplicaron en este trabajo. Además, los resultados obtenidos en ellos, muestran la utilidad de la construcción de clasificadores para predecir resultados de estudiantes en exámenes o procesos de su vida académica.

2.2.1 Detección de patrones de bajo rendimiento académico y deserción estudiantil con técnicas de minería de datos [9]

Su objetivo fue determinar en la comunidad universitaria perfiles de bajo rendimiento académico y deserción estudiantil aplicando técnicas de descubrimiento del conocimiento, a partir de los datos almacenados en las bases de datos durante los últimos 15 años. Este proceso se apoyó con TariyKDD[2], una herramienta de minería de datos de distribución libre.

Aunque el trabajo de investigación este orientado a detectar bajos rendimientos de estudian-

[1] `ftp://ftp.icfes.gov.co/SABER11/SB11-FTP_Algunos_Totales_de_Control-v1-1.pdf`

[2] `http://developer.berlios.de/projects/tariykdd/`

tes en la universidad, su proceso de investigación es similar al que se esperaba llevar en este proyecto, por lo tanto sirvió de guía para el desarrollo de este. Además de que la información utilizada para predecir el comportamiento académico de los estudiantes se basa en información socio-económica, que es la misma con la que se cuenta en las bases de datos del ICFES para construir el clasificador.
Este trabajo desarrolló de manera organizada y bien estructurada, cada uno de los pasos planteados en [5]. El algoritmo de minería de datos utilizado fue arboles de clasificación con C4.5 [5, 16], los resultados entregados por esta investigación mostraron cuales eran los perfiles de los estudiantes que podrían prever un bajo rendimiento.
La investigación se realizó en la Universidad de Nariño de Colombia. Las bases de datos recolectadas contenían información sobre el desempeño académico e información personal de 46173 estudiantes, acumuladas durante 15 años.
Inicialmente se contaba con 69 atributos en las bases de datos que describían las características de los estudiantes, pero después de pasar por el proceso de limpieza y transformación de los datos, y utilizando TariyKDD para seleccionar los atributos más relevantes, se llegó a una lista de 26 atributos y una cantidad de registros de 20329.
Finalmente, se aplicaron las técnicas de minería de datos para clasificación, en este caso el algoritmo C4.5 y se obtuvieron las reglas que indicaban que tipo de situaciones personales de un estudiante podrían llevar a obtener un bajo rendimiento en la universidad.
Unos ejemplos de las reglas obtenidas son:

- Si el estrato socioeconómico es 2, el ponderado de exámenes de estado ICFES está entre 50 y 70, es del Sur de Nariño, está en primer semestre y pertenece a la facultad de Ciencias Humanas, entonces su rendimiento es Bajo. El 68 % con estas características se clasifican de esta manera.

- Si la edad de ingreso es menor o igual a 18 años, el estrato socioeconómico es 2, género masculino, el ponderado ICFES está entre 50 y 70, vive con la familia, es del Sur de Nariño, está en primer semestre, está en la facultad de Ciencias Naturales y Matemáticas, entonces su rendimiento es Bajo. El 67 % con estas características se clasifican de esta manera.

2.2.2 Minería de datos en la educación [10]

En este documento se describe el uso de la minería de datos aplicada a entornos educativos y su uso pedagógico. De manera muy clara y especifica brinda una vista de cómo se debe

realizar un proceso de minería de datos en educación y su importancia.
Además muestra un ejemplo: "Identificación de características de fracasos escolares en institutos". En este se usan arboles de decisión porque permiten encontrar cuales son las variables que tienen mayor relación con la variable que se desea predecir. El algoritmo de árboles de decisión utilizado fue CHAID [11] (Chi-Squared Automatic Interaction Detection). CHAID realiza comparaciones en pares para encontrar la variable de predicción más altamente relacionada con la variable raíz. En sistemas de muchas variables, tener esta función implementada en un ordenador es esencial para procesar amplios conjuntos de datos (Big Data) [10]. El resultado entregado fue un árbol, el cual se debió analizar para determinar su información.

2.2.3 Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo de los k vecinos más cercanos [12]

Se aplicaron técnicas de minería de datos para buscar conocer previamente si los estudiantes eran propensos a abandonar su carrera en la Universidad Tecnológica de Izúcar de Matamoros, México, tomando como base de análisis los datos del estudio socioeconómico del EXANI-II[3], elaborado por el CENEVAL[4], mismo que se aplica desde el año 2003 en la institución. Para esta investigación se utilizaron específicamente dos algoritmos: el algoritmo de árboles de clasificación C4.5 y el algoritmo de los k vecinos más cercanos [16].
En el proceso de transformación de los datos, el más largo en muchos de los trabajos que se realizan, incluyendo este, se modificó e integró toda la información encontrada en distintos formatos de almacenamiento, además no tenía un estructura clara, porque constaba de información de estudiantes de 14 cuatrimestres, por supuesto en cada registro no se realizaban las mismas preguntas, además de que muchas de estas se pueden responder como "No lo sé", después de superar esta etapa y aplicar los algoritmos previstos, se encontró que el algoritmo de los k vecinos más cercanos funciona bien con pocos datos (477 instancias), pero al momento de probarlo con 6525 instancias, el algoritmo C4.5, tuvo resultados más confiables (98,98 %).
Aquí también se realizó la creación de una herramienta que pueda ser accedida por cualquier usuario y le informa sobre la probabilidad de que un estudiante deserte. Los resultados mostraron que la edad, la situación económica y el nivel de inglés, tienen fuerte relación con que el estudiante deserte de la universidad.

[3] `http://www.ceneval.edu.mx/ceneval-web/content.do?page=1738`
[4] `http://www.ceneval.edu.mx/ceneval-web/content.do?page=1702`

2.2.4 Modelo predictivo para la determinación de causas de reprobación mediante minería de datos [13]

En este trabajo, realizado en la Universidad Tecnológica de Puebla, México, se llevó a cabo el análisis de los datos que permitirán generar un modelo que ayude a predecir, desde que los alumnos ingresan a la Universidad, las causas que los llevarán a reprobar, así como las materias con mayor riesgo de ser reprobadas.

El algoritmo de clasificación utilizado fue C4.5, y se recolectaron datos con información de todas las carreras ofrecidas por la universidad. Este proceso de nuevo fue demorado, porque sus fuentes de almacenamiento poseían distintas estructuras y no existía una homogeneidad en los datos.

Los resultados obtenidos mostraron que de las 157 materias distintas que ofrece la universidad en sus carreras, 64 materias tienen un porcentaje de reprobación menor a 40 % y por lo tanto no generan un árbol de predicción. En las materias con un porcentaje mayor al 50 % se determinó que el factor principal de reprobación es el profesor que imparte la materia y también la edad de los estudiantes.

2.2.5 La metodología del Data Mining. Una aplicación al consumo de alcohol en adolescentes [14]

Aunque en esta investigación el objetivo no estaba focalizado en descubrir conocimiento en el área de la educación. Es un trabajo interesante el cual sirvió de guía en este trabajo de grado, ya que en él se realizó la construcción de 3 algoritmos distintos de predicción: redes neuronales, arboles de decisión y Naive Bayes [5].

El trabajo se centra en demostrar la importancia que tiene el descubrimiento del conocimiento en bases de datos al momento de predecir comportamientos en distintas áreas como educación, finanzas, comercio, telecomunicaciones, salud, entre otras.

Estos algoritmos fueron aplicados en un conjunto de datos que contenía 7030 registro que informaban sobre el consumo de alcohol en jóvenes de entre 14 y 18 años con una cantidad de 20 variables que incluían información de la personalidad como los constructos de autoestima, impulsividad, conducta antisocial y búsqueda de sensaciones.

Los mejores resultados se obtuvieron con el modelo construido con redes neuronales, 64,1 % de precisión al momento de predecir si un joven consumía o no alcohol. Los otros resultados fueron una precisión de 62,3 % con árboles de decisión usando el algoritmo CART [5] y una precisión de 59,9 % usando Naive Bayes.

2.3 Marco Teórico

2.3.1 Metodología para el proceso del descubrimiento del conocimiento (KDD)

Para la realización de este trabajo se utilizó el conjunto de fases presentadas en [5] donde establece que: "Este proceso es iterativo e interactivo. Es iterativo ya que la salida de alguna de las fases puede hacer volver a pasos anteriores y porque a menudo son necesarias varias iteraciones para extraer conocimiento de alta calidad. Es interactivo porque el usuario (experto en el dominio del problema) interviene en la toma de muchas decisiones".
La teoría del proceso del desarrollo de un proyecto de descubrimiento del conocimiento en bases de datos, define los siguientes pasos:

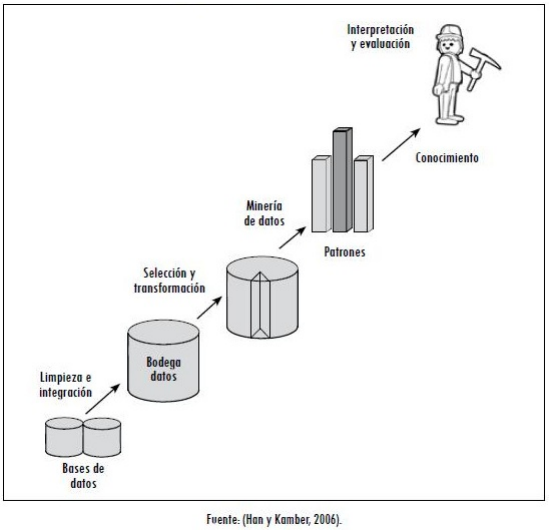

Figura 2.1: Proceso de descubrimiento del conocimiento en bases de datos.

2.3.1.1 Selección (*Extraction*) [5, 9]

En esta etapa del proceso, es donde se investigan las fuentes, en las cuales se podrán encontrar los datos que me servirán para general un almacén de datos con información útil. Se recolectan todos los datos obtenidos en estas fuentes y se organizan en bases de datos que se usaran para la etapa de transformación.
Comúnmente en los trabajos orientados a la academia [9, 10, 12, 13], estos datos son la información de los estudiantes de la institución a lo largo del tiempo. Los datos pueden incluir información acerca de la situación socio-económica del estudiante, sus notas, su en-

torno académico, información familiar, entre muchas otras que defina la institución que es importante recolectar.
En [13], donde se buscaba encontrar las reglas que indicaran si un estudiante era propenso a reprobar una materia, se seleccionaron como datos útiles: calificaciones por área del examen de admisión EXANI II, datos relevantes del estudio socio-económico, calificación del test de intereses vocacionales (KUDER), calificación del test de coeficiente intelectual (RAVEN), estilos de aprendizaje, evaluación a profesores, asignaturas cursadas y su promedio por cuatrimestre.

2.3.1.2 Transformación (*Transformation*) [5, 9]

En esta etapa el proceso se basa en realizar primero una limpieza de los datos, la limpieza permite obtener datos sin valores nulos o anómalos, además se estandarizan los datos para que sean del mismo tipo.
Para eliminar los datos nulos, se puede omitir este atributo o en algunos casos, cuando no son muchos los registros carentes de este valor, se pueden aplicar técnicas estadísticas como la media para insertar un valor valido y así no eliminar el registro o el atributo de la base de datos.
Estos procesos de limpieza, integración y agregación de los datos, entregan como resultado una base de datos modificada con los atributos relevantes en los registros, para responder al objetivo de la investigación, muchas veces esto genera que la cantidad de registros que se tenían inicialmente, se disminuya significativamente. Pero esto permitirá obtener resultados mucho más confiables, ya que no habrá datos que generen conflictos a la hora de generar clasificaciones con respecto a algunos atributos.

2.3.1.3 Carga (*Load*) [5, 9]

En esta etapa se transfieren los datos procesados en las fases anteriores al medio de almacenamiento escogido por el equipo de desarrollo. Estos medios de almacenamiento pueden ser bases de datos, Data Warehouse, Data Mart, entre otros.

2.3.1.4 Minería de datos [5, 9]

El objetivo de esta etapa es la búsqueda y descubrimiento de patrones insospechados y de interés utilizando diferentes técnicas de descubrimiento tales como clasificación, clustering, patrones secuenciales, asociación, entre otras [9]. Las diferentes técnicas utilizadas pertenecen a campos como la inteligencia artificial y estadísticas. Algunos algoritmos usados comúnmente

son: arboles de decisión C4.5, ID3 [15] y CHAID (Detección Automática de Interacción basada en Chi-Cuadrado), algoritmo de los k vecinos más cercanos, Naive Bayes, redes neuronales, algoritmo de reglas de asociación Apriori, algoritmo de inducción de reglas como el Prism, diferentes versiones de algoritmos evolutivos, programación genética basada en gramática.

2.3.1.5 Interpretación y evaluación de los resultados [5, 9]

En esta etapa se interpretan los patrones descubiertos y posiblemente se retorna a los anteriores pasos o etapas para posteriores iteraciones. Esta etapa puede incluir la visualización de los patrones extraídos, la remoción de los patrones redundantes o irrelevantes y la traducción de los patrones útiles en términos que sean entendibles para el usuario final de la aplicación [9].

2.3.2 Clasificación

En la minería de datos, probablemente la tarea más popular por su objetivo es la clasificación. En esta tarea el objetivo es construir una función que permita, usando los atributos de una instancia, conocer la clase a la cual puede pertenecer la instancia. Para la construcción de los clasificadores se utiliza un conjunto de datos de entrenamiento, los cuales contiene un grupo de atributos predictores y un atributo a predecir, normalmente conocido como la clase a predecir.
Formalmente, en [16] se define como la tarea de aproximar una *función objetivo* desconocida, que describe cómo instancias del problema deben ser clasificadas de acuerdo a un experto en el dominio,

$$\Phi : I \times C \to \{T, F\}$$

por medio de una función llamada el *clasificador*,

$$\Theta : I \times C \to \{T, F\}$$

donde

$$C = \left\{c_1, ..., c_{|c|}\right\} \text{ es un conjunto de categorías predefinido}$$

e

$$I \text{ es un conjunto de instancias del problema}$$

Comúnmente cada instancia $i_j \epsilon I$ es representada como una lista $A = \{a_1, a_2, ..., a_{|A|}\}$ de valores característicos, conocidos como *atributos*, i.e. $i_j = \{a_1, a_2, ..., a_{|A|j}\}$.
Si

$$\Phi : i_j \times c_i \to T, \text{ entonces es llamado un ejemplo positivo de } c_i$$

mientras si

$$\Phi : i_j \times c_i \to F \text{ éste es llamado un ejemplo negativo de } c_i$$

Para generar automáticamente el clasificador de c_i es necesario un proceso inductivo, llamado el *aprendizaje*, el cual por observar los atributos de un conjunto de instancias preclasificadas bajo c_i o $\bar{c_i}$, adquiere los atributos que una instancia no vista debe tener para pertenecer a la categoría. Por tal motivo, en la construcción del clasificador se requiere la disponibilidad inicial de una colección Ω de ejemplos tales que el valor de $\Phi(i_j, c_i)$ es conocido para cada $\langle i_j, c_i \rangle \epsilon \Omega \times C$. A la colección usualmente se le llama *conjunto de entrenamiento* (Tr).

2.3.3 Evaluación de clasificadores

Al momento de evaluar un clasificador, lo que se busca es conocer como es clasificada cada una de las instancias de un conjunto de evaluación. En estas instancias se conoce la clase a la cual pertenecen. El clasificador, procesa el conjunto de evaluación y retorna la clase a la que cada instancia debería pertenecer según los atributos predictores que contiene.
Usando las clasificaciones reales y las retornadas después de la evaluación, se generan 2 valores de evaluación de la calidad del clasificador. Estos 2 valores son Precisión y Exhaustividad (Presicion and Recall).
La precisión es la fracción de instancias correctamente clasificadas en una clase y la exhaustividad en la fracción de instancias de una clase que se clasificaron correctamente. [17]
Teniendo que:

$$A = Conjunto\ de\ instancias\ de\ la\ clase\ X\ en\ el\ conjunto\ de\ evaluación$$
$$B = Conjunto\ de\ instancias\ clasificadas\ en\ la\ clase\ X\ por\ el\ clasificador$$

Entonces

$$Presición = \frac{|A \cap B|}{|B|}$$
$$Recall = \frac{|A \cap B|}{|A|}$$

Cuanto más cerca estén estos valores a 1, mejor será la calidad del clasificador.

2.3.4 Generador de árboles de decisión C4.5

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica. De tal manera que la decisión final se puede tomar siguiendo las condiciones desde la raíz hasta alguna de las hojas[18].
El problema de la construcción de un árbol puede ser explicado de una manera recursiva. Primero se selecciona un atributo para ser ubicado en la raíz del árbol, y se crea una rama por cada posible valor que puede tomar este atributo. Esto divide el conjunto de datos de entrenamiento en distintas secciones con cada valor del nodo raíz. Y el proceso se repite de manera recursiva por cada una de las nuevas ramas, usando las instancias que están presentes en cada nuevo nodo. La condición de parada es cuando todas las instancias en un nodo tienen la misma clasificación[19].

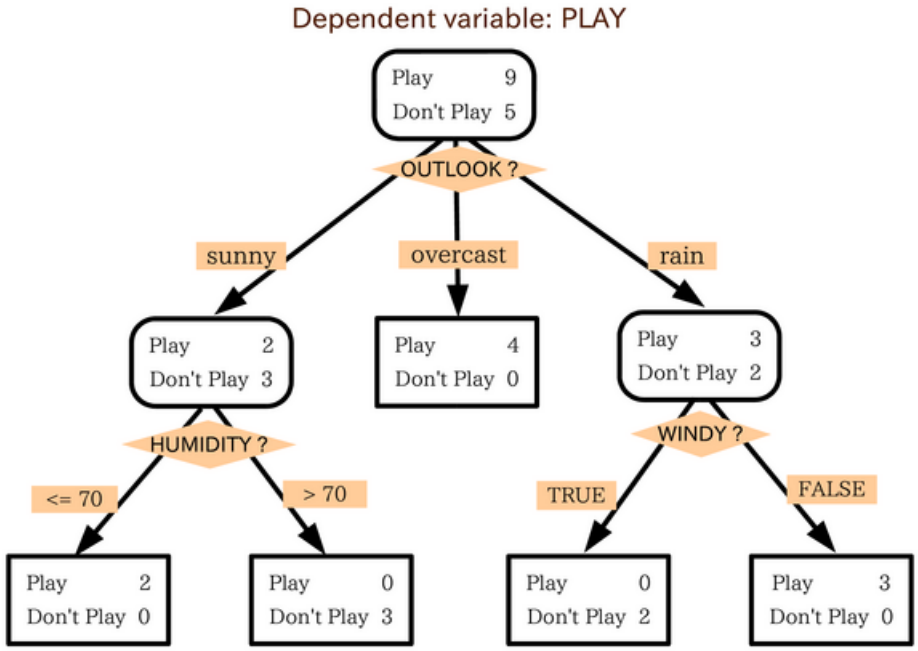

Figura 2.2: Representación gráfica de un árbol de decisión.

El problema de decidir que atributo debe ser ubicado en cada nodo se resuelve usando el factor de ganancia de información. En cada iteración se evalúa la ganancia de información que genera cada uno de los atributos y se selecciona el que obtenga un mayor valor. El valor de ganancia de información es calculado usando Gain Ratio.
Teniendo un conjunto S de instancias, y $frec(C_i, S)$ que es el número de casos de S que pertenecen a la clase C_i y $|S|$es el número de instancias en el conjunto. Entonces la entropía de S es:

$$H(S) = -\sum_{i=1}^{k}(\frac{frec(C_i,S)}{|S|} * log_2(\frac{frec(C_i,S)}{|S|}))$$

Después se procede a encontrar la entropía de cada uno de los atributos de las instancias. Posteriormente, conociendo la entropía de cada subconjunto $|S|$ (S_j es el subconjunto de instancias que tienen igual valor en el atributo que se desea evaluar) se calcula la ganancia de información que cada atributo genera.

$$H_X(S) = \sum_{j=1}^{n}(\frac{|S_j|}{|S|} * H(S_j)) \text{ Entropía del atributo } X \text{ en el conjunto } S.$$
$$G(X) = H(S) - H_x(S) \text{ Ganancia de información del atributo } X.$$

Se selecciona el atributo con mayor ganancia para ser ubicado en el nodo que se está iterando, y se procede a calcular el siguiente nodo. Las siguientes iteraciones no tendrán en cuenta en las instancias los atributos que han ido siendo seleccionados previamente[20].

2.3.5 K vecinos más cercanos

La regla del vecino más cercano es simple, este asigna a la instancia que se quiere clasificar, la clase del ejemplo más próximo utilizando una función de distancia. Esta función usualmente es la función de distancia Euclidiana, aunque una alternativa también frecuentemente usada es la función de distancia Manhattan.

El problema con esta sencilla manera de clasificar las instancias, es que toma mucho tiempo comparar, cada una de las instancias de entrenamiento con la instancia a clasificar, para determinar cuál es la más cercana. La manera de reducir el tiempo que toma esta comparación es representando los datos de entrenamiento como un árbol, este árbol es llamado un kD-tree[21], porque almacena un grupo de puntos en un espacio k-dimensional. Siendo k el número de atributos de las instancias.

El uso de los kD-tree ha ayudado a solucionar el problema de la comparación uno por uno, ya que el árbol divide el espacio en dimensiones que agrupan instancias con atributos similares, así cada evaluación no debe compararse contra cada instancia, sino que se compara contra las dimensiones, y se asigna a la clase de la dimensión con la que tuvo una mayor similitud[19].

Este algoritmo tiene 3 elementos principales: el conjunto de instancias de entrenamiento S, una métrica para calcular la distancia entre los objetos $d(x_i, x')$ y el valor de vecinos mas cercanos k.

Para clasificar una nueva instancia $z = (x', y')$ el algoritmo calcula las distancias con las instancias de entrenamiento $(x, y)\epsilon S$ para determinar la lista de vecinos más cercanos. x

representa el conjunto de atributos predictores de una instancias e y representa la clase de la instancia.
Una vez se tiene la lista de los vecinos más cercanos se selecciona la clase de mayor frecuencia entre estos:

$$Clase\,Mayoritaria = y' = \underset{v}{argmax} \sum_{(x_i,y_i)\epsilon S_z} I(v, y_i)$$

donde v es el valor de la clase, y_i es el valor de la clase del i-ésimo vecino de la lista, S_z es el conjunto de las distancias entre las instancias de entrenamiento y la instancia de evaluación z y $I(a,b)$ es una función que retorna 1 si a y b son iguales, 0 si no[22, 23].

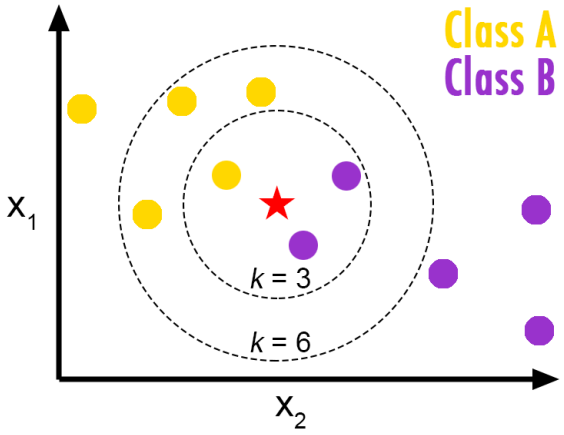

Figura 2.3: En esta representación, si k=3 el algoritmo retornará la clase B pero si k=6 el algoritmo retornará la clase A. Ya que se selecciona la clase mayoritaria entre los vecinos cercanos.

2.3.6 Naive Bayes

Los métodos bayesianos combaten uno de los problemas que poseen las técnicas de minería de datos, el manejo de la incertidumbre. Evitan este problema utilizando la teoría de la probabilidad para cuantificar la incertidumbre.
La inducción de modelos probabilísticos a partir de los datos conocidos, permite que se realice un razonamiento sobre los nuevos datos a observar. Además de que puede calcular la probabilidad asociada a cada una de las hipótesis. Y permite actualizar la creencia en un conjunto de hipótesis, cada que se estudia una nueva instancia, es decir con cada nueva instancia, su probabilidad mejora ya que es acumulativa[24].

El fundamento principal del clasificador Naive Bayes es la suposición de que todos los atributos son independientes conocido el valor de la clase. Esto da lugar a un modelo grafico probabilístico donde la raíz es la clase y todos los atributos son hijos de este nodo.
Teniendo en cuenta estos conceptos, el teorema de Bayes es representado por:

$$P(h|O) = \frac{P(O|h)*P(h)}{P(O)}$$

Que indica la probabilidad de que una hipótesis h suceda, teniendo en cuenta un conjunto de observaciones O.
En un problema de clasificación el conjunto de observaciones es $O = \{A_1, A_2, ..., A_n\}$ y la hipótesis h es cada una de las clases posibles $h = \{c_1, c_2, ..., c_n\}$. Se debe calcular la probabilidad de cada una de las clases en el conjunto h y seleccionar la que entregue el valor más alto.

$$C_{MAP} = \underset{c \epsilon h}{arg\,max} \frac{P(O|c)*P(c)}{P(O)} = \underset{c \epsilon h}{arg\,max} P(O|c) * p(c)$$

*MAP = Máximum A Posteriori, valor de probabilidad máxima obtenida después de calcular todas las clases en el conjunto h.

Teniendo en cuenta que Bayes supone que cada atributo es independiente, entonces la probabilidad puede calcularse de la siguiente manera:

$$C_{MAP} = \underset{c \epsilon h}{arg\,max} P(c) * \prod_{i=1}^{|O|} P(A_i|c)$$

La probabilidad de cada uno de los atributos viene dada por el número de instancias en las cuales el atributo A_i toma el valor a_i y la clase C toma el valor c_i sobre la cantidad de veces que la clase C toma el valor c_i[5, 25].

$$P(a_i|c_i) = \frac{frec(a_i, c_i)}{frec(c_i)}$$

De esta manera el algoritmo Naive Bayes se entrena y es capaz de clasificar una nueva instancia.

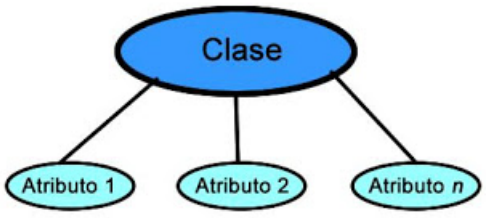

Figura 2.4: Representación gráfica del algoritmo Naive Bayes.

2.3.7 Funciones de base radial

Las redes neuronales artificiales (RNA) tienen distintas aplicaciones, entre ellas se encuentra la clasificación [26]. Su objetivo es tratar de emular la capacidad humana de procesar información.

Las RNA pueden ser entrenadas usando métodos de aprendizaje supervisado o no supervisado [27]. Una de las redes encontradas en el aprendizaje supervisado es la red de Funciones de Base Radial. Las redes RBF constan de 3 capas, en la primera se reciben los atributos de entrada a ser clasificados, en la segunda capa o capa oculta se utiliza una función de cálculo para realizar la transformación, no lineal, desde el espacio de la capa de entrada al espacio de la capa intermedia. En la tercera capa se encuentra los valores posibles de la clase a clasificar.

En la capa oculta se calcula la distancia euclidiana existente entre la instancia de entrada y el vector de puntos, denominados centros, aprendido por la red. Teniendo esta distancia, entonces se le aplica una función de tipo radial con forma gaussiana.

Los parámetros que debe aprender una red RBF son los centros de la función o funciones gaussianas y los pesos de la función lineal que seleccionará la salida de la red.

Si la instancia de entrada es x y M es el número de neuronas en la capa oculta entonces:

$$\phi(x) = (\phi_1(x), \phi_2(x), ..., \phi_M(x))^T$$

Donde $\phi_i(x)$ representa la distancia euclidiana entre el i-ésimo centro y las entradas $exp(-\lambda_i \|x - c_i\|^2)$. Y la salida de la red viene dada por:

$$y = w^T * \phi(x)$$

Donde w es el vector de pesos entre la capa oculta y la de salida.

El cálculo de los centros de la red se realiza principalmente usando el algoritmo de los K vecinos más cercanos. Donde K se corresponde con la cantidad de neuronas en la capa oculta.

Una de las desventajas que presentan las redes RBF es que necesitan de una mayor cantidad de instancias de entrenamiento, comparadas con otras redes como el Perceptron Multi Capa, para realizar clasificaciones con una mayor precisión. Pero son más rápidas de entrenar[5, 28].

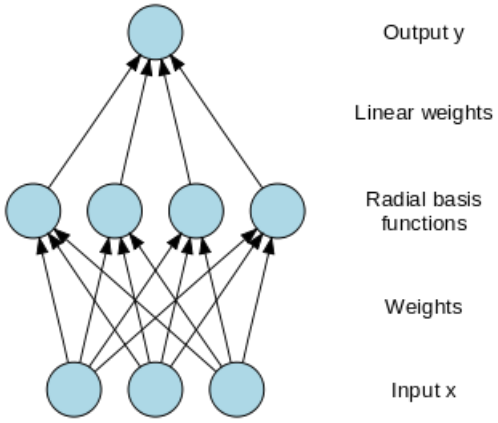

Figura 2.5: Estructura de las capas de una red neuronal de funciones de base radial.

Capítulo 3

Proceso de Selección, Transformación y Carga

Como primer paso para el desarrollo del proyecto, teniendo en cuenta las fases de la metodología de descubrimiento del conocimiento [5], se debió proceder a la preparación de los datos, esto es más conocido como proceso de "Extracción, Transformación y Carga" (ETL, por sus siglas en ingles). Durante esta fase se trabajó en homogeneizar la información contenida en las bases de datos fuente. Se analizó cual era la mejor forma de estandarizarlos y la manera de almacenarlos para posteriormente usarlos en el procesamiento de los algoritmos de descubrimiento del conocimiento.

3.1 Selección

Las fuentes de información, donde se contenían los datos necesarios para la creación del clasificador, fueron de fácil recolección, ya que estas son suministradas directamente por el ICFES de manera libre. El servicio brindado por el ICFES para el uso de estas bases de datos para promover la investigación ayuda a que se puedan realizar trabajos como este con el fin de determinar anomalías, conductas, mejoras, etc. sobre la calidad de la educación en Colombia.

La información recolectada estaba constituida por 24 bases de datos, que contenían información de cada uno de los exámenes aplicados a lo largo de 12 años desde el año 2000 hasta el 2011 (Actualmente la información del año 2012 ya se encuentra disponible, pero al momento de realizar este proceso de ETL no lo estaba, así que no se tomó como insumo para la construcción de los clasificadores), durante cada año se realizan 2 pruebas Saber 11°. La cantidad

de registros almacenados en estas 24 bases de datos se encontraba alrededor de 5'700.000.
Las bases de datos fueron obtenidas en formato Access, un tipo de archivo que se utiliza para administrar bases de datos y pertenece a Microsoft. Para su primer procesamiento se decidió transportar la información a tablas administradas por el motor de bases de datos MySQL[1], esto se hizo, porque para la posterior transformación de los datos, los scripts de procesamiento se construyeron usando el lenguaje de programación Python[2] y este tiene mejor soporte a la hora de conectarse con bases de datos MySQL.

3.2 Transformación

Teniendo la información recolectada se procedió a trasformar los datos obtenidos.
Esta transformación consta de estandarizar el formato en que son presentados los datos para su procesamiento, una coherencia entre el formato de presentación de los datos mejora la calidad de los resultados obtenidos, ya que estos tienen un valor de representación específico para cada caso que se puede presentar con cierto atributo, en caso de que para un atributo existan varios valores para indicar la misma situación o que para varias situaciones exista solo un valor valido, afecta la precisión con la que pueden trabajar los clasificadores.
Para la transformación del formato de los datos se utilizó como guía el diccionario de datos suministrado por el ICFES[3], en donde se muestra el historial de cómo se registró la información en las respectivas bases de datos. En este diccionario de puede apreciar la relación de las campos indagados, el año en que se registró y los valores con los que se registraba en cada año.
Los datos transformados y las modificaciones aplicadas se presentan a continuación:

- Atributo COLE_INST_VLR_PENSION indica con un código el valor mensual que pagó o paga el evaluado actualmente de pensión en el colegio. Este atributo está presente en las 24 bases de datos, pero presenta variación en el formato de presentación, la tabla 3.1 muestra esta variación.

  Al analizar la variación en la representación del dato se puede notar que a medida que pasa el tiempo los rangos de especificación del valor de la pensión se han ido disminuyendo. Pero se pueden visualizar 3 rangos que son constantes "No paga", "Entre

[1] `http://www.mysql.com/`
[2] `http://www.python.org/`
[3] `ftp://ftp.icfes.gov.co/SABER11/SB11-Diccionario_de_Datos-v1-6.pdf`

Periodo 2000-2004		**Periodo 2005-2007**		**Periodo 2008**		**Periodo 2009-2011**	
Código	Valor	Código	Valor	Código	Valor	Código	Valor
1	No paga	0	No paga	0	No paga	0	No paga
2	Menos de $30.000	1	Menos de $33.000	8	Menos de $90.000	8	Menos de $87.000
3	Entre $30.000 y $50.000	2	Entre $33.000 y menos de $50.000	9	Entre $90.000 y menos de $120.000	9	Entre $87.000 y menos de $120.000
4	Entre $50.000 y menos de $70.000	3	Entre $50.000 y menos de $70.000	10	Entre $120.000 y menos de $150.000	10	Entre $120.000 y menos de $150.000
5	Entre $70.000 y menos de $100.000	4	Entre $70.000 y menos de $100.000	11	Entre $150.000 y menos de $250.000	11	Entre $150.000 y menos de $250.000
6	Entre $100.000 y menos de $150.000	5	Entre $100.000 y menos de $150.000	12	$250.000 o más	12	$250.000 o más
7	Entre $150.000 y menos de $250.000	6	Entre $150.000 y menos de $250.000				
8	Más de $250.000	7	Más de $250.000				

Tabla 3.1: Variación de la representación del atributo COLE_INST_VLR_PENSION en las bases de datos.

$150.000 y menos de $250.000" y "$250.000 o más", por lo tanto siempre existió un valor que representara estos 3 niveles.

El problema aparece al momento de querer encontrar similitudes entre los rangos intermedios, se encuentra una representación de los datos casi igual entre el primer periodo con el segundo periodo y entre el tercer periodo con el cuarto periodo. Al momento de tomar la decisión se tuvo en cuenta cual transformación causaría una mejor perdida en la calidad de la información, por lo tanto se decidió dejar la representación utilizada en el periodo 2005-2007, ya que al momento de transformar los datos se mantiene una mayor especificidad, la cual no se mantendría si se hubieran tomado los valores de

alguno de los últimos periodos.

Las transformaciones que tomaron los valores de los otros periodos se muestran en la tabla 3.2.

Periodo 2000-2004		**Periodo 2008**		**Periodo 2009-2011**	
Código Anterior	Nuevo Código	Código Anterior	Nuevo Código	Código Anterior	Nuevo Código
1	0	0	0	0	0
2	1	8	3	8	3
3	2	9	4	9	4
4	3	10	5	10	5
5	4	11	6	11	6
6	5	12	7	12	7
7	6				
8	7				

Tabla 3.2: Transformación del atributo COLE_INST_VLR_PENSION para ser registrado en el nuevo almacenamiento de datos.

- Atributo ECON_SN_COMPUTADOR indica si el evaluado tiene o no tiene computador en la casa, en el años 2008 se preguntó en conjunto si la persona tenia servicio de internet y la cantidad de computadores que tenía. La variación de la representación de este atributo se presenta en la tabla 3.3.

Periodo 2008-I		**Periodo 2008-II**		**Periodo 2009-2011**	
Código	Valor	Código	Valor	Código	Valor
0	No	0	No	0	No
1	Sí, con internet	3	Uno	3	Sí
2	Sí, sin internet	4	Dos o más		

Tabla 3.3: Variación de la representación del atributo ECON_SN_COMPUTADOR en las bases de datos.

El análisis de las representaciones deja visualizar que el factor importante de la información es conocer si el estudiante posee o no computador, por lo tanto los valores que se deciden dejar para que representen este valor en el nuevo almacenamiento de datos, solo deben indicar si la respuesta fue Sí o No, esta situación solo se encontraba en el último periodo, por lo tanto fue esta representación la que se decidió preservar. Las transformaciones aplicadas a los otros periodos se muestran en la tabla 3.4.

Periodo 2008-I		**Periodo 2008-II**	
Código Anterior	Nuevo Código	Código Anterior	Nuevo Código
0	0	0	0
1	3	3	3
2	3	4	3

Tabla 3.4: Transformación del atributo ECON_SN_COMPUTADOR para ser registrado en el nuevo almacenamiento de datos.

- Atributos:
  - ESTU_NACIMIENTO_ANNO
  - ESTU_NACIMIENTO_DIA
  - ESTU_NACIMIENTO_MES

  representan la fecha de nacimiento del estudiante. Estos 3 atributos tendrían un mayor significado si se presenta como ESTU_EDAD, por lo tanto se realizó la transformación de estos para almacenar el valor numérico de la edad del estudiante.
  La transformación se realizó calculando la fecha de presentación de examen con estos valores registrados, como la fecha exacta de los exámenes es desconocida, no se puede dar un 100 % de garantía de que todas las edades fueron correctamente registradas pero el rango de error no es mayor a un mes y esto es solo aplicable a los estudiantes que nacieron en los meses de abril o septiembre, esto es porque aunque no se tenga clara la fecha exacta de aplicación del examen si es conocido que el primer examen del año se realiza en abril y el segundo en septiembre.

  Por ejemplo si un evaluado nació el 15 septiembre de 1995 y presentó el segundo examen del año 2011, se calcula que este estudiante tiene una edad de 16 años, pero desconocemos el día en que se realizó el examen, si fue antes del día 15, entonces el cálculo de la edad es erróneo pero solo por unos pocos días, pero si se realizó después del día 15, entonces el cálculo es correcto.

  Como se puede apreciar, a pesar de no existir una precisión del 100 % con todas las edades registradas, si se puede confiar en que en la gran mayoría de los casos estas son exactas y las que no lo son, tiene un error de pocos días.

- Atributo ESTU_TRABAJA indica si la persona trabaja, si el trabajo es remunerado y la cantidad de tiempo que dedica a trabajar. También presenta variaciones en sus

registros, estas variaciones se presentan en la tabla 3.5.

Periodo 2000-2004I		**Periodo 2008-2011**	
Código	Valor	Código	Valor
N	No	0	No
S	Sí	1	Si, con remuneración en dinero y/o especie
		2	Si, como ayudante sin remuneración
		3	Si, para contribuir a pagar su matrícula y/o los gastos del hogar
		4	Si, por ser práctica obligatoria del programa de estudios
		5	Si, para adquirir experiencia y/o recursos para sus gastos personales
		6	Si, menos de 20 horas a la semana
		7	Si, 20 horas o más a la semana

Tabla 3.5: Variación de la representación del atributo ESTU_TRABAJA en las bases de datos.

Al analizar la variación de la representación se observó que si se conservaban los valores de representación del segundo periodo era muy difícil encontrar un valor concordarte para el "Sí" del primer periodo, mientras que si se decidía conservar los valores de representación del primer periodo, todos los valores del segundo periodo tendrían una correspondencia en la transformación. Por lo tanto se decidió conservar la representación del primer periodo.

La transformación por lo tanto fue, el valor 0 se convirtió en "N" y los demás valores numéricos se convirtieron en "S".

- Atributos FAMI_COD_EDUCA_MADRE y FAMI_COD_EDUCA_PADRE representa el nivel de educación de los padres, presentan variaciones en su representación. Las variaciones se presentan en la tabla 3.6.

  Se observó que en el segundo periodo existe una mayor precisión sobre los datos almacenados, si se trasladan estos al primer periodo, se puede perder información importante, por lo tanto se decidió transformar los valores del primer periodo a los del segundo. Sin embargo del primer periodo, los valores "1", "2" y "5" no tienen correspondencia en el segundo periodo, por lo tanto estos se conservaron sin modificación alguna. Los nuevos valores de representación se pueden observar en la tabla 3.7.

- Atributo FAMI_COD_OCUP_MADRE y FAMI_COD_OCUP_PADRE representan la actividad a la que se dedican los padres del evaluado. Presentan variaciones en su

Periodo 2000-2004I		**Periodo 2008-2011**	
Código	Valor	Código	Valor
0	Ninguno	0	Ninguno
1	No tuvo Escuela	9	Primaria Incompleta
2	Preescolar	10	Primaria Completa
3	Básica Primaria	11	Secundaria (bachillerato) Incompleta
4	Básica Secundaria	12	Secundaria (bachillerato) Completa
5	Media Vocacional	13	Educación técnica o tecnológica incompleta
6	Tecnológico o Técnico	14	Educación técnica o tecnológica Completa
7	Universitario	15	Educación Profesional Incompleta
8	Postgrado	16	Educación Profesional Completa
		17	Postgrado
		99	No sabe

Tabla 3.6: Variación de la representación de los atributos FAMI_COD_EDUCA_MADRE y FAMI_COD_EDUCA_PADRE en las bases de datos.

Periodo 2000-2004I	
Código Anterior	Nuevo Código
0	0
1	1
2	2
3	10
4	12
5	5
6	14
7	16
8	17

Tabla 3.7: Transformación de los atributos FAMI_COD_EDUCA_MADRE y FAMI_COD_-EDUCA_PADRE para ser registrados en el nuevo almacenamiento de datos.

representación. Estas variaciones son mostradas en la tabla 3.8.

Los valores entre los 2 periodos guardan muchas similitudes, en la mayoría de los casos solo varían los nombres como se registró el valor. El segundo periodo presenta algunos valores no existentes en el primero y viceversa. Para preservar los valores actuales se decidió transformar los valores del primer periodo al segundo, del primer periodo los

Periodo 2000-2004I		**Periodo 2008-2011**	
Código	Valor	Código	Valor
0	Fallecidos (solo se preguntó esta opción en 2000)	13	Empresario(a)
1	Empresarios	14	Pequeño empresario(a)
2	Administradores o gerentes	15	Empleado con cargo como director(a) o gerente general
3	Profesionales independientes	16	Empleado(a) de nivel directivo
4	Profesionales empleados	17	Empleado(a) de nivel técnico/profesional
5	Trabajadores independientes	18	Empleado(a) de nivel auxiliar o administrativo
6	Trabajadores empleados	19	Obrero u operario empleado(a)
7	Rentistas	20	Profesional independiente
8	Obreros	21	Trabajador por cuenta propia
9	Jubilados	22	Hogar
10	Hogar	23	Pensionado(a)
11	Estudiantes	24	Rentista
12	Personas que en la actualidad no devengan ingreso por ningún concepto o están buscando trabajo	25	Estudiante
		26	Otra actividad u ocupación
		99	NO sabe

Tabla 3.8: Variación de la representación de los atributos FAMI_COD_OCUP_MADRE y FAMI_COD_OCUP_PADRE en las bases de datos.

valores “0” y “12” no tienen una representación en el segundo periodo, así que estos permanecieron con ese valor. La transformación de los códigos se puede visualizar en la tabla 3.9.

- Atributo FAMI_ING_FMILIAR_MENSUAL representa el ingreso mensual en salarios mínimos que devengan en total lo habitantes del hogar del evaluado. Este atributo presenta variaciones a lo largo de los años por lo tanto debió ser transformado. Estas variaciones se muestran en la tabla 3.10.

  Los rangos de variación entre los 2 periodos son muy similares entre sí, se decidió conservar los valores del segundo periodo, ya que el código 7 de este, agrupa los códigos 7, 8 y 9 del primer periodo, por lo tanto es una transformación más concordarte que intentar llevar el código 7 del segundo periodo a la representación de algún código en el primer periodo. El código 6 del segundo periodo agrupa al 100 % el código 5 y al

Periodo 2000-2004I	
Código Anterior	Nuevo Código
0	0
1	13
2	15
3	20
4	17
5	21
6	19
7	24
8	19
9	23
10	22
11	25
12	12

Tabla 3.9: Transformación de los atributos FAMI_COD_OCUPA_MADRE y FAMI_COD_-OCUPA_PADRE para ser registrados en el nuevo almacenamiento de datos.

Periodo 2000-2004I		**Periodo 2008-2011**	
Código	Valor	Código	Valor
0	Menos de 1 SM (Salarios Mínimos)	1	Menos de 1 SM
1	Entre 1 y menos de 2 SM	2	Entre 1 y menos de 2 SM
2	Entre 2 y menos de 3 SM	3	Entre 2 y menos de 3 SM
3	Entre 3 y menos de 5 SM	4	Entre 3 y menos de 5 SM
4	Entre 5 y menos de 7 SM	5	Entre 5 y menos de 7 SM
5	Entre 7 y menos de 9 SM	6	Entre 7 y menos de 10 SM
6	Entre 9 y menos de 11 SM	7	10 o más SM
7	Entre 11 y menos de 13 SM		
8	Entre 13 y menos de 15 SM		
9	15 o más SM		

Tabla 3.10: Variación de la representación del atributo FAMI_ING_FMILIAR_MENSUAL en las bases de datos.

33 % el código 6 del primer periodo, así que como solo concordaba con una parte del código 6 del primer periodo, este se transformara al código 7 del segundo periodo. Las transformaciones aplicadas se presentan en la tabla 3.11.

- Atributos FAMI_SN_LEE_ESCRIBE_MADRE y FAMI_SN_LEE_ESCRIBE_PADRE representa si los padres del evaluado pueden o no leer. Este valor ha sido registrado de

Periodo 2000-2004I	
Código Anterior	Nuevo Código
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	7
8	7
9	7

Tabla 3.11: Transformación del atributo FAMI_ING_FMILIAR_MENSUAL para ser registrado en el nuevo almacenamiento de datos.

2 maneras distintas, pero en el diccionario de datos del ICFES no se indica entre que rangos de tiempo se utilizó cada una.

Estas representaciones se pueden visualizar en la tabla 3.12.

Primera representación		**Segunda representación**	
Código	Valor	Código	Valor
0	No	N	No
1	Si	S	Sí
99	No lo sabe	99	No lo sabe

Tabla 3.12: Variación de la representación de los atributos FAMI_SN_LEE_ESCRIBE_MADRE y FAMI_SN_LEE_ESCRIBE_PADRE en las bases de datos.

A pesar de no conocer la variación de los periodos, se conocía los valores y entre los 2 periodos las opciones eran idénticas, entonces se decidió conservar la primera representación para la construcción del nuevo almacenamiento de datos.

- Atributos COLE_CODIGO_COLEGIO, COLE_CODIGO_INST, ECON_ZONA, ESTU_ANNO_EGRESO, ESTU_MES_EGRESO, ESTU_IES_COD_DESEADA, ESTU_CARR_COD_DESEADA y ESTU_CODIGO_RESIDE_MCPIO se eliminaron ya que fueron almacenados con códigos de los cuales el ICFES no entrega el significado de su valor. Por lo tanto, así estos atributos llegaran a ser relevantes no se les hubiera podido dar una interpretación apropiada.

Las anteriores transformaciones fueron aplicadas a los atributos que se utilizaron para predecir las puntuaciones en las distintas áreas evaluadas, es decir, los valores que fueron utilizados como atributos predictores. Pero los valores que representan las puntuaciones también necesitaron ser modificados, los atributos a predecir.
Para transformar estos datos se utilizó discretización, ya que estos estaban presentados en valores numéricos, unos en un intervalo de [1, 100] y otros de [1, 10], se procedió a crear rangos de puntajes y asignarles un identificador a cada uno. Esta transformación se muestra en la tabla 3.13.

- Áreas académicas que se califican con un puntaje de 1 a 100:
  - Lenguaje
  - Matemáticas
  - Biología
  - Química
  - Física
  - Filosofía
  - Ciencias sociales
  - Inglés
  - Interdisciplinar violencia y sociedad
  - Interdisciplinar medio ambiente
- Áreas académicas que se califican con un puntaje de 1 a 10:
  - Profundización en lenguaje
  - Profundización en matemáticas
  - Profundización en biología
  - Profundización en ciencias sociales

En los atributos que indican los puntajes de las áreas evaluadas también se debieron aplicar cambios en su representación.

- TEMA_GEOGRAFIA y TEMA_HISTORIA representaban los puntajes obtenidos por los evaluados en las áreas de geografía e historia, pero esto se modificó a partir del

Discretización aplicada a los puntajes de las áreas académicas		
Calificación de 1 a 100	Calificación de 1 a 10	Nuevo código
[1, 10)	1	A
[10, 20)	2	B
[20, 30)	3	C
[30, 40)	4	D
[40, 50)	5	E
[50, 60)	6	F
[60, 70)	7	G
[70, 80)	8	H
[80, 90)	9	I
[90, 100]	10	J

Tabla 3.13: Transformación aplicada a los atributos que representaban los puntajes obtenidos por los evaluados en las distintas áreas académicas.

2006 en donde se introdujo la prueba de ciencias sociales que combina las 2 pruebas anteriores. Por lo tanto en las bases de datos anteriores a 2006 se promedió estos 2 puntajes y se registró como uno solo que se llamó TEMA_CIENCIAS_SOCIALES.

- TEMA3_IDIOMA_P representa el puntaje obtenido en la prueba de idioma, antes del 2006II se podía escoger entre inglés, alemán y francés para presentar la prueba, pero a partir del 2007 solo se presenta en inglés, por lo tanto los valores de alemán y francés se almacenaran como nulo para evitar contenidos innecesarios.

- TEMA2_PROFUNDIZACION_P y TEMA4_INTERDISCIPLINAR estas áreas presentaban distintas opciones al evaluado, en la prueba de profundización se podía escoger entre Biología, Matemática, Historia, Lenguaje y Ciencias Sociales. Y en la prueba interdisciplinar se podía escoger entre Medio Ambiente, Violencia y Sociedad, y Medios de Comunicación y Cultura. Pero a partir del 2010II solo se escoge una entre 5 opciones (las pruebas de profundización en historia y comunicación y cultura desaparecieron), por lo tanto, estas 2 pruebas que ya no se evalúan no se conservaron en el nuevo almacenamiento de datos.

3.3 Carga

Como último paso de la primera fase, fue necesaria la creación de un almacenamiento de datos, en donde la información sea registrada con las transformaciones aplicadas, para que

cumpla la condición de ser homogénea en sus atributos.
El modo físico de almacenamiento utilizado fue una tabla en el motor de base de datos de MyQSL, la cual después de la eliminación de algunos parámetros, los cuales no servían como atributos predictores, ya que eran llaves primarias o su valor no podía ser interpretado para darle un significado, constó de 88 atributos de los cuales 78 fueron atributos predictores y los 10 restantes fueron atributos a predecir (también conocidos como "clases").
El diseño lógico de este almacenamiento se muestra en la figura 3.1.

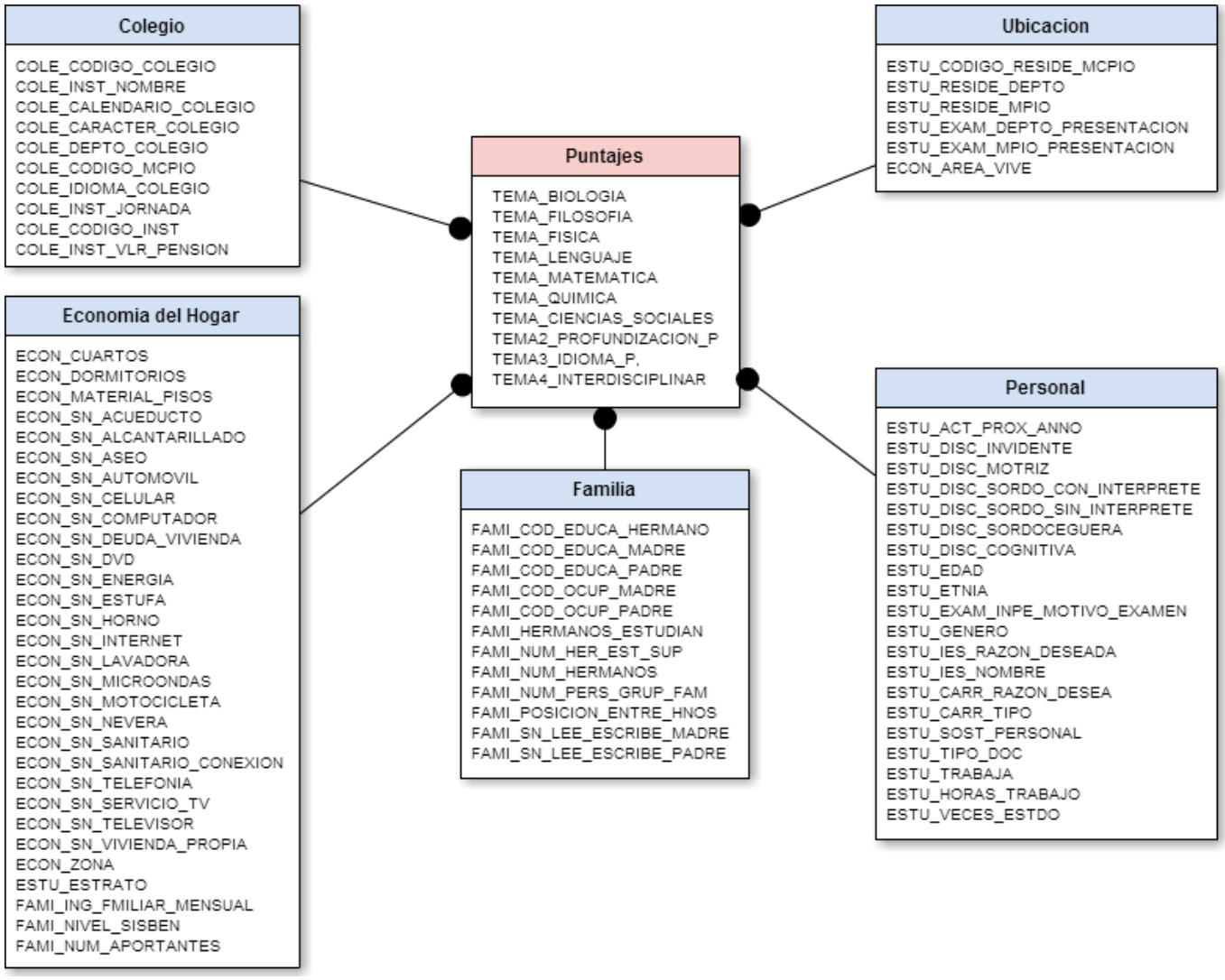

Figura 3.1: Diseño logico del modelo de almacenamiento de datos.

En el diseño lógico del modelo dimensional (figura 3.1) se pueden observar las dimensiones y la tabla de hechos.

Las dimensiones, que contienen los atributos que ayudan a responder las preguntas que se pueden plantear sobre el modelo, constan de la información que define las particularidades de un evaluado. Estas dimensiones son:

- **Colegio**: brinda información sobre el colegio en el que estudia o estudió el evaluado, por ejemplo el carácter, el calendario, la jornada escolar, entre otros.
- **Economía del hogar**: brinda información sobre el ambiente económico del evaluado, como por ejemplo las características físicas de su hogar, como el material de los pisos y los sistemas de servicios públicos disponibles en él. Además de informar también sobre los ingresos monetarios y la cantidad de personas en las que deben repartirse estos ingresos.
- **Familia**: brinda información sobre las características de la familia del evaluado, como por ejemplo las ocupaciones y niveles educativos de sus padres y hermanos.
- **Ubicación**: brinda información sobre la zona geográfica donde vive el evaluado, como el municipio y departamento, y si vive en una zona rural o urbana.
- **Personal**: brinda información sobre las características físicas y de personalidad del estudiantes, como su edad, el genero, información sobre discapacidades, las razones que lo llevaron a escoger una carrera, entre otras.

Para una mayor información sobre el significado de cada uno de estos atributos y de la forma en cómo se almacenan en las bases de datos, se puede observar el diccionario de datos del ICFES.
En la tabla de hechos, la cual almacena los atributos que contienen la información con la cual se responderá a las preguntas planteadas al modelo, se pueden observar los registros que indican el puntaje en cada una de las áreas académicas evaluadas.
Cómo se relacionó en la sección 2.1.4, las áreas académicas son 14, 8 de obligatoria presentación para todos los evaluados y 6 áreas electivas de las cuales se debe elegir una. Estas 6 electivas constan de 4 profundizaciones y 2 pruebas interdisciplinares, por eso en los atributos de la tabla hechos se encuentran los atributos TEMA2_PROFUNDIZACION_P, TEMA4_INTERDISCIPLINAR, estos fueron interpretados con la ayuda de un código que indicaba cual prueba escogió cada evaluado.
Así, con el procesamiento de esta información, que fue recolectada, homogeneizada y almacenada nuevamente, se pudo proceder a trabajar con los distintos algoritmos de clasificación

elegidos para evaluar en el trabajo de grado. La descripción del desarrollo de esta etapa de evaluación se encuentra en el siguiente capítulo.

Capítulo 4

Evaluación de Clasificadores

Después de tener cargado el nuevo almacenamiento de datos con la información homogeneizada, se procedió a evaluar los algoritmos de clasificación para elegir los más confiables para construir la interfaz de consultas, que permita conocer los posibles resultados de un evaluado. El objetivo de este trabajo de grado era entregar información sobre los posibles resultados de un estudiante en cada una de las áreas académicas evaluadas por la prueba Saber 11°. Estas áreas son en total 14, así que para cada una de estas áreas se debió construir un clasificador que predijera con el mayor índice de confianza el resultado a obtener por el evaluado.

4.1 Selección de atributos

El primer paso para la construcción de estos clasificadores fue filtrar la cantidad de atributos predictores disponibles. Al momento de realizar esta fase se contaba con 78 atributos predictores, varios de estos atributos constaban con una gran cantidad de registros nulos, ya que la frecuencia con la que se almacenó en las bases de datos fue muy baja.
Para la selección de estos atributos a ser filtrados se evaluó primero la cantidad de veces en las que este fue registrado, y utilizando la suite de algoritmos de minera de datos Weka[1] se realizó un filtro de selección de atributos utilizando algoritmos especializados para esta tarea.

4.1.1 Selección manual

La tabla 4.1 muestra los atributos que se seleccionaron de manera manual, el criterio para esta primera selección fue que al menos estuvieran presentes en 12 (50 %) de las bases de

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

datos consultadas.

ID	Atributo	Cantidad
4	COLE_CODIGO_MCPIO*	24
6	COLE_INST_JORNADA	24
7	COLE_INST_VLR_PENSION	24
36	ESTU_DISC_MOTRIZ	24
37	ESTU_DISC_SORDO_CON_INTERPRETE	24
38	ESTU_DISC_SORDO_SIN_INTERPRETE	24
41	ESTU_EDAD	24
47	ESTU_GENERO	24
35	ESTU_DISC_INVIDENTE	22
44	ESTU_EXAM_DEPTO_PRESENTACION	22
51	ESTU_RESIDE_DEPTO	22
1	COLE_CALENDARIO_COLEGIO	17
2	COLE_CARACTER_COLEGIO	17
48	ESTU_IES_RAZON_DESEADA	17
49	ESTU_CARR_RAZON_DESEA	17
55	ESTU_TRABAJA	17
58	FAMI_COD_EDUCA_MADRE	17
59	FAMI_COD_EDUCA_PADRE	17
60	FAMI_COD_OCUP_MADRE	17
61	FAMI_COD_OCUP_PADRE	17
63	FAMI_ING_FMILIAR_MENSUAL	17
68	FAMI_NUM_PERS_GRUP_FAM	17
3	COLE_DEPTO_COLEGIO	16
5	COLE_IDIOMA_COLEGIO	15
43	ESTU_ETNIA	15
46	ESTU_EXAM_MPIO_PRESENTACION	15
52	ESTU_RESIDE_MPIO	15
54	ESTU_TIPO_DOC	14

Tabla 4.1: Atributos predictores seleccionados de manera manual.

* Originalmente el atributo registrado 24 veces era COLE_INST_NOMBRE, pero como este tenía alrededor de 12500 opciones, se volvió complicado trabajar con él, por lo tanto se decidió utilizar este para conocer el municipio al que el colegio pertenece y así el atributo COLE_CODIGO_MCPIO se pudo registrar las 24 veces.

Tras aplicar este primer filtro la cantidad de atributos predictores considerados útiles, se redujo a 28. Después, usando estos 28 atributos, se procedió a utilizar los algoritmos de selección de atributos predictores disponibles en Weka.

4.1.2 Selección automática

Weka brinda la capacidad de realizar esta selección de atributos utilizando métodos evaluadores que se dividen en 2 tipos: los evaluadores de subconjuntos y los prorrateadores de atributos. El primer tipo se encarga de evaluar cual subconjunto de datos es el mejor para predecir la clase, este utiliza un método de búsqueda del tipo forward o backward. El segundo tipo no evalúa subconjuntos, si no que asigna a cada atributo un valor que indica el nivel de correlación que posee con la clase a predecir, este al contrario de utilizar un método de búsqueda, utiliza un ranker para indicar la importancia de cada atributo.

El primer paso para poder utilizar Weka fue construir los archivos arff[2] estos son los archivos que usa Weka para transmitir la información a los algoritmos. En la primera prueba se construyó un archivo arff con 1000 registros, con los datos obtenidos desde el almacenamiento de datos construido en el capítulo anterior. La idea de realizar esta prueba inicial era determinar los tiempos que demoraba cada posible combinación entre los métodos de evaluación y los métodos de búsqueda que ofrece Weka. Después de realizar esta prueba se optó por la utilización de 6 combinaciones que fueron las que lograron manipular la cantidad de datos presentados, en un tiempo aceptable, teniendo en cuenta que más adelante se ejecutarían pruebas con archivos que contenían hasta 500.000 registros.

La razón para que algunas combinaciones no fueran seleccionadas, fue porque nunca se logró obtener una solución en el tiempo otorgado (20 minutos) o lanzaron error al momento de procesar el archivo. El tiempo de 20 minutos, fue tomado como un límite justificable, ya que de los algoritmos de los cuales si se obtuvo respuesta, no tomaron un tiempo mayor a 3 minutos.

La tabla 4.2 muestra las combinaciones seleccionadas.

Como ya se explicó que se construyó un clasificador por cada área evaluada, la selección de atributos también se aplicó a cada una de estas clases a predecir, es decir que por cada área se crearon los archivos arff para ser analizados en los algoritmos.

Los archivos ARFF creados para realizar cada prueba fueron construidos seleccionando de manera aleatoria registros de la base de datos homogeneizada. Además estos archivos fueron creados con distintas cantidades de registros, con el fin de tener una mejor visión sobre el comportamiento de los algoritmos a medida que se aumentaba la cantidad de registros. Estas cantidades fueron 1.000, 10.000, 100.000 y 500.000. Es decir, por cada área evaluada se crearon 4 archivos arff que fueron ejecutados en las 6 combinaciones distintas, dando así 24 evaluaciones a analizar.

[2] `http://weka.wikispaces.com/ARFF`

ID	Método de evaluación	Método de búsqueda
1	CfsSubsetEval[3]	BestFirst[4]
2	CfsSubsetEval	GeneticSearch[5]
3	CfsSubsetEval	GreedyStepwise[6]
4	ClassifierSubsetEval[7]	GeneticSearch
5	GainRatioAttributeEval[8]	Ranker[9]
6	InfoGainAttributeEval[10]	Ranker

Tabla 4.2: Combinaciones utilizadas en Weka para seleccionar atributos predictores.

4.2 Creación de vistas minables

Posterior a la ejecución de los algoritmos de selección automática, se reevaluó la calidad de los atributos predictores y su capacidad de ser realmente aportantes al momento de predecir el posible puntaje en un área académica. Se decidió que los atributos ESTU_EXAM_MPIO_PRESENTACION y ESTU_EXAM_DEPTO_PRESENTACION no serían tenidos en cuenta para la construcción de los clasificadores, ya que al analizar la base de datos, en más del 90 % de los casos el valor de estos 2 atributos era equivalente al valor de los atributos COLE_CODIGO_MCPIO y COLE_DEPTO_COLEGIO respectivamente, por lo tanto se generaba redundancia en los datos y esto ocasiona que la calidad de las predicciones disminuya.
También el atributo ESTU_TIPO_DOC que indica el tipo de documento con el que la persona se inscribió, no entregaba información realmente útil. Al analizar la base de datos alrededor del 95 % de los registros se dividían entre "Tarjeta de Identidad" o "Cedula de Ciudadanía", los otros valores posibles no eran comunes. Este atributo se solapa con el atributo ESTU_EDAD, ya que el valor "Tarjeta de Identidad" los tenían los evaluados menores de 18 años y "Cedula de Ciudadanía" los evaluadores mayores de 18 años. Por lo tanto, es claro que conociendo la edad de un evaluado, se puede conocer qué tipo de documento utiliza, así que la edad absorbió este atributo y por eso no fue tenido en cuenta a la hora de construir los clasificadores.

[3] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/CfsSubsetEval.html
[4] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/BestFirst.html
[5] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/GeneticSearch.html
[6] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/GreedyStepwise.html
[7] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/ClassifierSubsetEval.html
[8] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/GainRatioAttributeEval.html
[9] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/Ranker.html
[10] http://weka.sourceforge.net/doc.stable/weka/attributeSelection/InfoGainAttributeEval.html

Después de analizar los resultados de la ejecución de los algoritmo de selección de atributos, se procedió a construir las vistas minables que iban a ser procesadas en cada uno de los algoritmos de construcción de clasificadores.

Para la construcción de estas vista minables, se realizó una selección entre los 28 atributos filtrados inicialmente y los atributos que tuvieron una mayor presencia en los resultados de los algoritmos de selección de atributos. En algunas áreas se crearon 2 vistas minables para tener una mejor perspectiva, respecto a distintos comportamientos. No se definió un límite mínimo de selección de los atributos en los resultados de los algoritmos, la cantidad de atributos que compone cada vista minable se decidió teniendo en cuenta el número de veces que fue seleccionado el más frecuente.

Los procesos de minería de datos involucran, además de los algoritmos, el conocimiento que se tenga sobre el problema, en ocasiones algunos atributos pueden no ser relevantes al momento de ejecutar los algoritmos, pero es conocido por las personas involucradas en el proceso, que estos atributos sí determinan un factor importante en el caso de estudio. Teniendo en cuenta este criterio, algunos atributos que tuvieron poca relevancia en los algoritmos de selección, fueron escogidos para ser predictores, ya que se consideró que su significado podría beneficiar la calidad de las predicciones.

La tabla 4.3 muestra los atributos procesados en los algoritmos de selección de atributos, para el caso del área de biología, y la frecuencia con la que los algoritmos de selección le dieron importancia. Cabe aclarar que en el caso de los algoritmos de evaluación que utilizan un método de búsqueda "Ranker" siempre aparecieron todos los valores, pero la puntuación asignada a algunos de ellos era de 0.0 por lo tanto estos no se contaron como atributos seleccionados por ese algoritmo.

En la tabla 4.3 se observa los atributos ordenados con base a la relevancia que tuvieron en los algoritmos de selección. Para crear las vistas minables se seleccionaron los de mayor frecuencia.

El criterio aplicado, en el caso de biología, a la selección de los atributos fue que estos se encontraran en al menos 12 de las combinaciones, es decir en el 50 % de estas, 10 atributos cumplieron con este criterio.

Después de analizar estos primeros 10 atributos se pudo notar que existían 2 que indicaban la misma información pero con diferente granularidad, estos eran COLE_DEPTO_COLEGIO y COLE_CODIGO_MCPIO, ambos indican la ubicación del colegio, uno a nivel departamental y el otro a nivel municipal, por lo tanto solo era necesario seleccionar uno de ellos para indicar la ubicación del colegio. Pero como los 2 tenían una frecuencia que cumplía con el criterio, se decidió crear 2 vista minables en donde una tuviera una granularidad de departamento en

ID	Atributo	# de veces que se seleccionó
58	FAMI_COD_EDUCA_MADRE	19
4	COLE_CODIGO_MCPIO	17
41	ESTU_EDAD	17
63	FAMI_ING_FMILIAR_MENSUAL	17
6	COLE_INST_JORNADA	16
52	ESTU_RESIDE_MPIO	16
7	COLE_INST_VLR_PENSION	15
59	FAMI_COD_EDUCA_PADRE	15
60	FAMI_COD_OCUP_MADRE	15
51	ESTU_RESIDE_DEPTO	12
3	COLE_DEPTO_COLEGIO	8
38	ESTU_DISC_SORDO_SIN_INTERPRETE	8
43	ESTU_ETNIA	7
61	FAMI_COD_OCUP_PADRE	7
5	COLE_IDIOMA_COLEGIO	6
47	ESTU_GENERO	6
49	ESTU_CARR_RAZON_DESEA	6
48	ESTU_IES_RAZON_DESEADA	5
36	ESTU_DISC_MOTRIZ	4
37	ESTU_DISC_SORDO_CON_INTERPRETE	4
1	COLE_CALENDARIO_COLEGIO	3
35	ESTU_DISC_INVIDENTE	3
2	COLE_CARACTER_COLEGIO	2
68	FAMI_NUM_PERS_GRUP_FAM	2
55	ESTU_TRABAJA	1

Tabla 4.3: Cantidad de veces que los atributos fueron seleccionados como buenos predictores por los algoritmos de selección de atributos, en el área de biología.

la ubicación del colegio y la otra una granularidad de municipio en la ubicación del colegio.

Así, siguiendo una lógica similar a la aplicada al área de biología, se crearon las vistas minables de las demás áreas académicas.

En las 14 áreas académicas se intentó seguir las mismas reglas de manera general, pero en algunas áreas como por ejemplo medio ambiente, eran pocos los atributos que cumplían con estar seleccionado en al menos 12 combinaciones, en el caso de medio ambiente, solo 3 atributos cumplían esto.

También para el caso de algunos atributos, como FAMI_COD_OCUP_PADRE, no tuvieron buena frecuencia de selección, pero FAMI_COD_OCUP_MADRE si lo hizo, teniendo en cuenta

que estos indican la misma información de los padres, en algunas vistas minables se agregó la ocupación del padre como atributo a pesar de que su frecuencia de selección no fuera alta. La tabla 4.4 muestra las vistas minables creadas para cada una de las áreas académicas.

Vistas minables	
Nombre	Atributos
Biologia9	6, 7, 41, 51, 58, 59, 60, 61, 63
Biologia10	4, 6, 7, 41, 52, 58, 59, 60, 61, 63
Filosofia7	1, 4, 7, 41, 52, 58, 59
Fisica11	1, 4, 7, 41, 47, 52, 58, 59, 60, 61, 63
Fisica211	1, 6, 7, 41, 47, 52, 58, 59, 60, 61, 63
Ingles11	4, 5, 6, 7, 41, 52, 58, 59, 60, 61, 63
Lenguaje10	4, 7, 41, 43, 52, 58, 59, 60, 61, 63
Lenguaje210	4, 6, 7, 41, 43, 58, 59, 60, 61, 63
Matematica10	4, 6, 7, 41, 47, 52, 58, 59, 60, 61
Matematica210	1, 3, 6, 7, 41, 47, 58, 59, 60, 61
MedioAmbiente5	2, 4, 52, 58, 60
MedioAmbiente7	2, 4, 52, 58, 59, 60, 61
ProBiologia6	1, 4, 7, 58, 60, 63
ProBiologia8	1, 4, 7, 58, 59, 60, 61, 63
ProLenguaje7	2, 4, 7, 41, 58, 60, 63
ProLenguaje9	2, 4, 7, 41, 58, 59, 60, 61, 63
ProMatematica7	4, 6, 7, 41, 58, 60, 63
ProMatematica9	4, 6, 7, 41, 58, 59, 60, 61, 63
ProSociales7	1, 4, 7, 41, 58, 60, 63
ProSociales9	1, 4, 7, 41, 58, 59, 60, 61, 63
Quimica10	4, 6, 7, 41, 52, 58, 59, 60, 61, 63
Sociales7	4, 7, 41, 52, 58, 60, 63
Sociales9	4, 7, 41, 52, 58, 59, 60, 61, 63
ViolenciaSociedad11	4, 6, 7, 41, 52, 55, 58, 59, 60, 61, 63

Tabla 4.4: Lista de las vistas minables creadas.

Teniendo lista la estructura para construir las vistas minables, se procedió a la carga de estas. Cada vista minable se creó con 1.000, 10.000 y 100.000 registros.

Las vistas minables se cargaron utilizando el lenguaje de programación Python. Se utilizó el generador de números aleatorios para seleccionar los registros en la base de datos de manera que se aseguró que no existía ninguna correlación entre los registros cargados. También se aseguró que estos registros fueran cargados sin ninguno de sus atributos con valor nulo, esto se hizo con el fin de que los algoritmos de clasificación tuvieran una mejor confiabilidad.

En el caso de las profundizaciones, en donde la cantidad de registros es menor, ya que la

presentación de estas pruebas es opcional, no se pudieron crear vistas minables de 100.000 registros, porque fue difícil lograr que todos los atributos coincidieran en no ser nulos, por eso en las profundizaciones los grupos fueron de 1.000, 10.000 y 50.000.
Con las vistas minables cargadas, se procedió a la evaluación de estos en los algoritmos de construcción de clasificadores.

4.3 Construcción de clasificadores

Utilizando Weka, se procedió a generar distintos clasificadores, teniendo en cuenta los trabajos revisados y que sirvieron de apoyo para este, los algoritmos seleccionados para evaluar las vistas minables fueron: generador de árboles de decisión C4.5[11], K vecinos más cercanos[12], Naive Bayes[13] y la red neuronal RBF (Radial Basis Function)[14].
Weka ofrece la posibilidad de ejecutar cada uno de estos algoritmos con una colección de datos de entrenamiento y evaluar su nivel de confianza con un archivo de evaluación.
Los archivos de entrenamiento fueron aquellos que se crearon siguiendo la estructura de las vistas minables presentadas en la anterior sección. En Weka es posible dividir estos archivos, para que un porcentaje sea de entrenamiento y el otro porcentaje sea de evaluación. Pero en este trabajo se decidió crear los archivos de evaluación a partir de un conjunto de datos completamente nuevo y desconocido para el conjunto de datos de entrenamiento, esto se logró utilizando los resultados de la prueba en el año 2012.
Como se explicó inicialmente, al momento de empezar este trabajo las bases de datos disponibles eran las de los años 2000 hasta 2011, pero al momento de empezar la creación de los clasificadores, la base de datos del año 2012 estaba disponible para su descarga y se pudo utilizar como un conjunto de datos desconocido para el conjunto de datos inicial.
Para cada vista minable se creó un archivo que tenía la misma estructura pero que se cargaba con registros de la base de datos de 2012, estos archivos de evaluación se crearon con una cantidad de 1000 registros. Es decir que por cada clasificador creado, se evaluaba su precisión y exhaustividad con 1000 registros de los cuales se conocía la clase, pero al clasificador se le presentaba como desconocida, para que la predijera.
Con las evaluaciones de precisión y exhaustividad se pudo visualizar el nivel de confianza que ofrecía cada clasificador, la precisión fue el factor en base al cual se tomó la decisión de

[11] `http://weka.sourceforge.net/doc.stable/weka/classifiers/trees/J48.html`
[12] `http://weka.sourceforge.net/doc.stable/weka/classifiers/lazy/IBk.html`
[13] `http://weka.sourceforge.net/doc.stable/weka/classifiers/bayes/NaiveBayes.html`
[14] `http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/RBFNetwork.html`

cuales clasificadores iban a ser utilizados en la construcción de la interfaz de consultas.
El en anexo A se encuentran las gráficas que muestran los valores de precisión y exhaustividad obtenidos al ejecutar cada una de las vistas minables en cada uno de los algoritmos seleccionados. Para el caso del algoritmo de los k vecinos más cercanos, se realizaron 2 ejecuciones. Una con un valor de k=1 y otra con el valor de k=2, con el fin de encontrar como afecta la selección del valor de k al desempeño del algoritmo.

4.4 Selección de los clasificadores

Teniendo en cuenta los resultados de precisión y exhaustividad que obtuvo cada uno de los clasificadores con cada una de las vistas minables presentadas por las áreas académicas, se eligieron los mejores resultados por cada una.
La elección de los mejores clasificadores en cada vista minable se realizó comparando los valores de precisión y exhaustividad. El criterio más importante fue la precisión, en caso de existir similitud entre 2 o más clasificadores en este valor, entonces se procedía a observar cuál de ellos presentaba una mejor exhaustividad.
Teniendo en cuenta estos criterios y los resultados relacionados en las gráficas del anexo A, se seleccionaron los atributos, la tabla 4.5 presenta esta selección.

Área académica	**Vista minable**	**Instancias de entrenamiento**	**Algoritmo del clasificador**	**Precisión & Exhaustividad**
Biología	Biologia9	100.000	Naive Bayes	0.418 & 0.450
Ciencias Sociales	Sociales7	100.000	Redes RBF	0.426 & 0.469
Filosofía	Filosofia7	1.000	Naive Bayes	0.375 & 0.374
Física	Fisica11	10.000	Redes RBF	0.418 & 0.408
Ingles	Ingles11	10.000	KNN K=1	0.663 & 0.659
Lenguaje	Lenguaje210	1.000	Naive Bayes	0.520 & 0.555
Matemática	Matematica10	10.000	KNN K=2	0.339 & 0.345
Medio Ambiente	MedioAmbiente7	10.000	KNN K=1	0.768 & 0.245
Profundización en Biología	ProBiologia6	50.000	Redes RBF	0.406 & 0.262
Profundización en Lenguaje	ProLenguaje7	10.000	KNN K=1	0.405 & 0.247
Profundización en Matemática	ProMatematica9	50.000	RBF	0.604 & 0.251
Profundización en Ciencias Sociales	ProSociales7	10.000	Naive Bayes	0.554 & 0.214
Química	Quimica10	10.000	KNN K=2	0.544 & 0.533
Violencia y Sociedad	Violencia11	10.000	Redes RBF	0.347 & 0.374

Tabla 4.5: Clasificadores seleccionados para la construcción de la interfaz de consultas.

Capítulo 5

Construcción de la Interfaz de Consultas

Después de haber seleccionado cuales serían los clasificadores que se utilizarían para realizar las predicciones en cada una de las áreas académicas, se debió tomar la decisión de cuál sería la mejor manera de presentar una interfaz de consultas, para que un usuario pudiera consultar cual sería el posible puntaje que obtendría un evaluado en cada una de las áreas académicas.

El primer aspecto a tener en cuenta era la interpretación de los clasificadores construidos. Weka permite guardar, en un archivo codificado, la estructura de un clasificador entrenado, para posteriormente utilizarlos con instancias de pruebas. Pero estos archivos codificados solo pueden ser cargados por las librerías de Weka. Es decir, que para poder utilizar los clasificadores creados, en la aplicación se debían utilizar estas librerías, por lo tanto el lenguaje base de la aplicación debía ser Java[1].

El segundo aspecto que se tuvo en cuenta para la construcción de la interfaz, fue que analizando las vistas minables seleccionadas y realizando una intersección entre los atributos de los que estaba compuesta cada una, se tenía un total de 18 parámetros a preguntar. Esta cantidad de parámetros podrían ser fácilmente indagados en un formulación en HTML, además de que, una aplicación de escritorio en donde se tenga solo un formulario para obtener una respuesta es poco llamativo y se podría obtener un mayor acceso de usuarios si esta interfaz está construida sobre un entorno web.

Las ventajas que se obtienen con aplicaciones web son por ejemplo: que pueden ser accedidas siempre, sin importar en que dispositivos el usuario se encuentre: computador, laptop, tableta

[1] `http://www.java.com`

o teléfono inteligente. Se ejecutan sin importar el sistema operativo del usuario, no requieren instalación, entre otras.
Así, conociendo los 2 aspectos más importantes que debía cumplir la aplicación, se decidió utilizar la plataforma de programación J2EE de Java, la cual permite la creación de Servlets, los cuales son archivos de código Java que se ejecutan dentro de un contenedor de Servlets [29].
Los contenedores de Servlets capturan las peticiones que un usuario realiza sobre un servicio web y se encarga de procesar esta solicitud, enviársela al Servlet que procesa los datos y retorna una respuesta que de nuevo es procesa por el contenedor y entregada al navegador para que el usuario pueda visualizar los resultados de su petición.
Para la interfaz de navegación del usuario se utilizó la tecnología AJAX (Asynchronous JavaScript And XML). AJAX oficialmente es definido como:

> "AJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes." [30].

AJAX permite que las consultas que se realizan desde una interfaz HTML hacia un servidor se realicen de manera asíncrona, evitando así la recarga constante de la página, cada vez que se desea consultar algo. Esto habilita al usuario a realizar diferentes tareas al tiempo, sin perder la posibilidad de continuar interactuando con la página.

5.1 Arquitectura

La arquitectura usada para la construcción de la aplicación, fue una arquitectura Cliente-Servidor. Del lado del cliente se tiene la capa de presentación que se carga por medio de un navegador web, esta presentación está construida usando HTML y una hoja de estilo CSS3. La conexión con el servidor de Servlets se realiza a través de JavaScritp usando la tecnología AJAX y la librería JQuery[2] para manipular el comportamiento de los componentes gráficos de la presentación, como el intercambio de div's y estado de inhabilidad de los botones.
En el servidor se encuentra el Servlet PrediXaber, que se encarga de recibir los datos ingresados por el usuario y construir con ellos los archivos arff a ser ejecutados en los clasificadores. Estos archivos arff son creados agregando los datos de entrada a una plantilla predefinida, que tiene la estructura de datos que concuerda con cada uno de los clasificadores construidos. Utilizando el archivo arff creado, se procede a cargar cada uno de los clasificadores usando las librerías de Weka, se ejecuta la clasificación por cada una de las áreas académicas y se

[2] `http://jquery.com/`

construye una respuesta que es enviada, de nuevo, a través de AJAX. Esta respuesta se recibe y es cargada en la página HTML para que el usuario pueda observar los resultados de la consulta.
La figura 5.1 muestra gráficamente el diseño de la arquitectura de la aplicación.

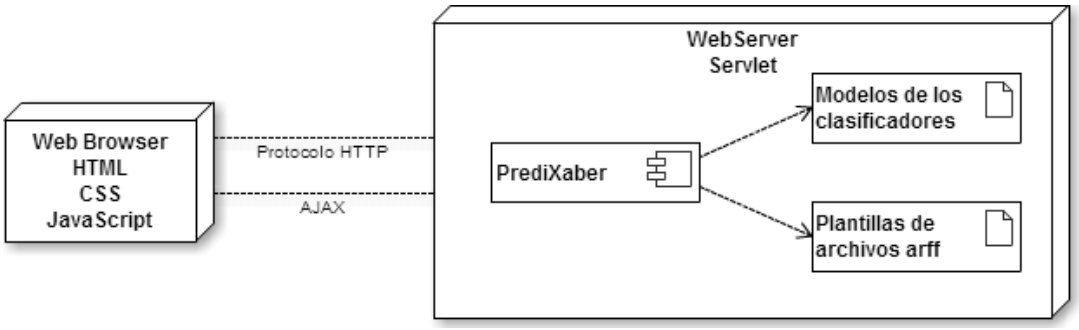

Figura 5.1: Diseño arquitectónico de la aplicación.

5.2 Flujo de navegación

El flujo de navegación normal que se debe seguir en la aplicación, es cargar la página principal donde se muestra el formulario con los campos requeridos para realizar la consulta. Después de ingresar la información correspondiente al evaluado, se procede a enviar este formulario para que sea procesado en el servidor. La respuesta es retornada desde el servidor y desplegada en la pantalla para que el usuario pueda conocer los resultados de su consulta.

Figura 5.2: Flujo normal de navegación que se sigue en la aplicación.

5.3 Codificación

El proceso de codificación de la aplicación, se realizó utilizando los lenguajes programación Java y JavaScript, en Java se hizo uso de la librería Weka y en JavaScript de JQuery. Se utilizó HTML para definir los componentes de la página principal presentada al usuario y CSS3 para darle estilo a esos componentes.
En el ambiente de desarrollo se utilizó como IDE de desarrollo Eclipse[3] y el contenedor de Servlets Tomcat[4].
Los parámetros que se envían al Servlet son procesados a través de un método POST para que la información sea enviada de manera codificada y no sea visible en la URL de la consulta.
En el Servlet construido heredando la clase predefinida en java HttpServlet[5] se construyó el método doPost() para procesar los datos recibidos y entregar una respuesta.
Dentro del Servlet la librería Weka se utilizó para la carga y ejecución de los clasificadores. Para cargar los archivos que contenían la información de los clasificadores se usó la clase SerializedClassifier, después de cargar la estructura, esta debía ser instanciada en un objeto del tipo Classifier para poder ejecutar el archivo arff. Se crea un objeto del tipo Instances que recibe como parámetro de construcción el BufferedReader que contiene el texto del archivo arff. Y por último, este objeto Instances es evaluado en el clasificador utilizando el método classifyInstance(), así se obtiene un valor para la clase a clasificar y este valor es retornado por medio del método doPost().
El código de la aplicación es sencillo, ya que solo se encarga de ejecutar las consultas usando los clasificadores anteriormente creados.

5.4 Pruebas de usabilidad

La usabilidad, en el software, se entiende como la facilidad que tiene un usuario de interactuar con la interfaz de las aplicaciones.
Los aspectos más importantes a evaluar en un diseño web son: curva de aprendizaje para el uso de la interfaz, eficacia de la interfaz, facilidad para memorizar, los errores que puede cometer el usuario y la satisfacción del usuario.
Para evaluar estos aspectos se hizo uso de la herramienta para evaluar usabilidad UsabilityHub[6]. Esta herramienta permita la creación de test que los usuarios pueden generar y

[3] `http://www.eclipse.org`
[4] `http://tomcat.apache.org/`
[5] `http://tomcat.apache.org/tomcat-7.0-doc/servletapi/javax/servlet/http/HttpServlet.html`
[6] `https://usabilityhub.com/`

compartir públicamente con las personas que ingresan a la página a responder estos test.
Se hizo uso de 2 tipos de test. El test de Five Second, el cual le presenta al evaluador una imagen durante 5 segundos, después de ese tiempo realiza una pregunta que el evaluador debe responder. Y el test Click, el cual presenta una imagen acompañada de una pregunta en donde se le indica al evaluador que presione click en el lugar donde el cree que se ejecutaría cierta acción.
En el test de Five Second se presentó una imagen con el inicio de la página y se le pregunto al usuario sobre cual creía que era el objetivo de la página, después de leer la presentación. El 85 % de los encuestados, respondió que su finalidad era predecir resultados en la prueba saber 11°, el 15 % restantes, respondieron no tener clara la finalidad de la página.
En los test de Click se presentaron 3 imágenes, cada una con una pregunta sobre una acción a ejecutar. Las imágenes 5.3, 5.4 y 5.5 muestra las preguntas realizadas y los resultados obtenidos.

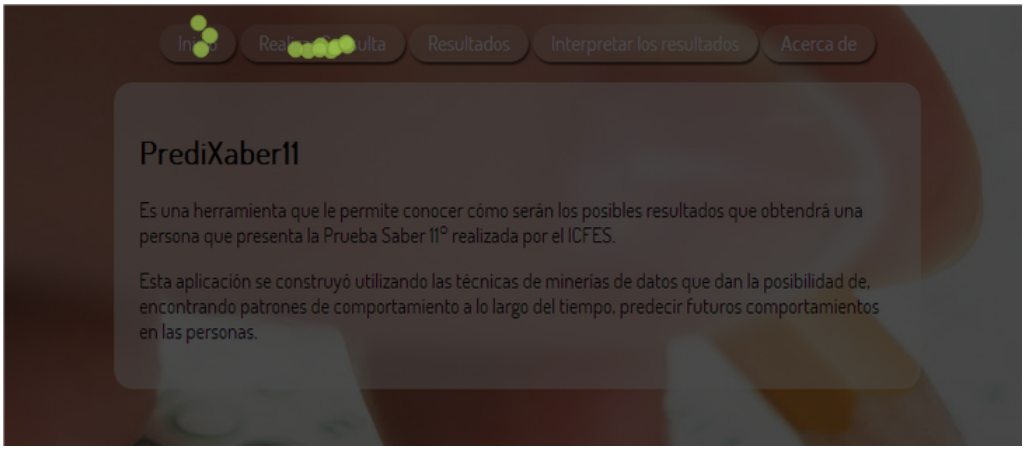

Figura 5.3: Acción: Mira las opciones en los links y selecciona aquella que creas que te permitirá ejecutar una consulta.

5.5 Pruebas de código

Para verificar la funcionalidad del código se generaron pruebas unitarias utilizando 2 frameworks diferentes pero conectables. Uno para evaluar que el flujo de navegación dentro de la aplicación retorne los resultados esperados y otro para evaluar que los métodos construidos dentro del Servlet funcionen correctamente al procesar la información recibida. Para construir las pruebas del flujo de navegación se utilizó el framework de automatización de pruebas

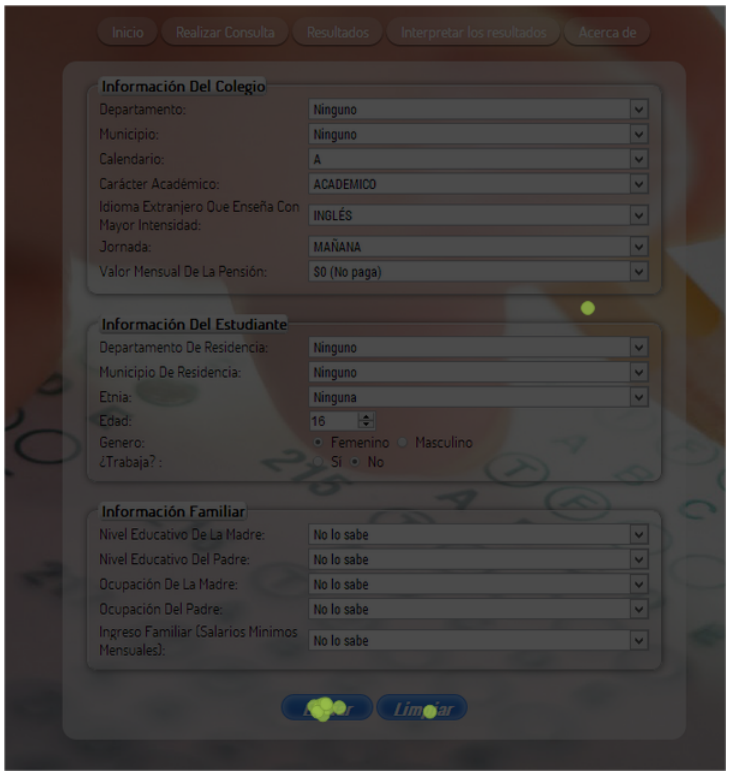

Figura 5.4: Acción: Donde harías click para enviar el formulario y recibir una respuesta.

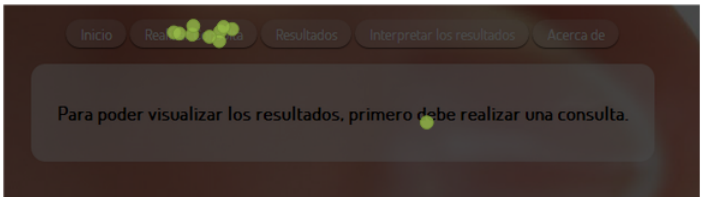

Figura 5.5: Acción: Donde harías click después de leer el mensaje.

Selenium[7]. Selenium permite grabar y reproducir pruebas a partir de un plugin que puede ser instalado en el navegador Firefox[8]. Con este plugin, el usuario inicia una grabación de pasos a seguir dentro del aplicativo web, entre cada paso ejecutado, el usuario puede agregar verificaciones de la respuesta o las respuesta esperadas al ejecutar ese paso. Selenium también permite la ejecución de estas grabaciones, así cada vez que se desee verificar si los resultados esperados de una secuencia de pasos están siendo correctamente procesados, se ejecuta el script creado con anterioridad y se verifica este comportamiento. Los scripts creados por Selenium pueden ser exportados a diferentes lenguajes de programación para ser ejecutados

[7] `http://docs.seleniumhq.org/`
[8] `http://www.mozilla.org/en-US/firefox/new/`

por otros frameworks de pruebas. Para el caso del presente trabajo, se decidió exportar el script creado al lenguaje de programación Java utilizando el framework de pruebas JUnit[9]. Con JUnit también se construyeron tests para evaluar el comportamiento de los métodos definidos en el Servlet utilizado en el servidor para procesar las consultas del usuario. Se crearon 16 métodos de test en total que permiten verificar tanto el comportamiento de la interfaz de usuario y del código generado dentro de las clases.

5.6 Pruebas de confiabilidad

En esta prueba se quería comprobar la calidad de las predicciones realizadas por los clasificadores. Pero no individualmente, como ya se realizó, sino en conjunto ejecutando la solicitud al Servlet que realiza el procesamiento de los datos.
Para ejecutar esta prueba se utilizaron 100.000 registros de las bases de datos del año 2012, la cual ya se advirtió que no fue utilizada en la construcción de los clasificadores. Se creó un script en Python con el cual se ejecutaron las solicitudes al Servlet. A cada registro se le aplico la transformación de datos correspondiente.
Al recibir la respuesta de la solicitud se comparaban los resultados de las áreas académicas para encontrar en cuantas se realizaba correctamente la predicción.
La tabla 5.1 muestra los resultados de cuantas áreas académicas coincidieron. El total de áreas académicas en las que es evaluada una persona son 9, 8 de núcleo común y 1 de elección del evaluado.

Cantidad de coincidencias	**Cantidad de registros**
9	98
8	881
7	3840
6	10116
5	17952
4	22166
3	20713
2	14598
1	8602
0	1034

Tabla 5.1: Resultados de la prueba de confiabilidad realizada a la aplicación.

Se puede observar que cerca del 33 % de los registros evaluados (32.887) tienen al menos 5

[9] `http://junit.org/`

coincidencias, que son más de la mitad de los puntajes a predecir. Se percibe que al momento de realizar consultas en conjunto sobre los clasificadores se pierde la calidad que estos mostraban de manera individual. Pero también se puede observar que las predicciones se comportan como una función gaussiana en donde la mayoría de los resultados están agrupados en predicciones con el 50 % de confianza.

Capítulo 6

Conclusiones y Trabajos Futuros

6.1 Conclusiones

- Se logró aplicar el proceso de minería de datos a la información suministrada por el ICFES, logrando construir clasificadores que permitieron la elaboración de una interfaz de consultas donde los usuarios podrán conocer previamente sus resultados.

- La suite de algoritmos de minería de datos Weka, es de gran utilidad en los procesos de descubrimiento del conocimiento en bases de datos. Su gran variedad de métodos que permiten realizar desde la selección de atributos, hasta algoritmos de clasificación construidos con distintas técnicas, hacen de esta herramienta un apoyo fundamental y que debería ser usada cuando se desarrollen proyectos de este tipo.

- En el caso de estudio, la creación de los clasificadores, requirió de gran ayuda de parte del desarrollador, para la selección de los más apropiados acorde a los atributos predictores. Se notó que al aplicar cambios sencillos en la selección de los atributos predictores, los clasificadores mostraban cambios drásticos en sus predicciones. Los algoritmos de construcción de clasificadores son sensibles al cambio de su estructura de entrenamiento. Con agregar o retirar un atributo en el conjunto de entrenamiento, la calidad de las predicciones variaba significativamente.

- El marco metodológico seguido en durante el proyecto [5], da una guía clara y útil a la hora de desarrollar proyectos de minería de datos, sus pasos son perfectamente explicados y se ajustaron precisamente a cada uno de los objetivos planteados y contribuyeron al cumplimiento de cada uno de estos.

- No se pudo encontrar un clasificador mejor o peor que otro, a excepción del algoritmo C4.5 el cual no genero resultados confiables. Todos los clasificadores tuvieron rendimientos confiables, pero algunos obtuvieron mejor rendimiento en ciertas áreas académicas. Esto se debe a que los algoritmos de clasificación tienen propiedades, en las que dependiendo el formato de los atributos predictores, generan una mayor calidad al momento de realizar las predicciones.

- El proceso de desarrollo del proyecto agrupó el mayor esfuerzo en la construcción de los clasificadores. La construcción de la interfaz de consulta fue sencilla, además de que se optó por un entorno web, en el que la codificación de las interfaces es simple y no requiere del uso de máquinas con requerimientos específicos.

- A pesar de que al evaluar la confiabilidad de la aplicación, la cantidad de clasificaciones que cumplieron con predecir correctamente todas las áreas académicas fue realmente baja, se obtuvieron resultados positivos al menos para el 50 % de las áreas académicas, además de que individualmente la calidad de las predicciones de los clasificadores estuvo por encima del 70 % de precisión.

- La interfaz de consultas que se construyó, constó del uso de las tecnologías actuales en el desarrollo web. Como lo son componentes gráficos de HTML5 y CSS3, además de la aplicación de AJAX que permite al usuario interactuar de una manera más confortable con la interfaz de consultas.

- El tiempo que tarda un clasificador en procesar una consulta y retornar una respuesta no presentó grandes diferencias entre los distintos algoritmos utilizados. Principalmente en la construcción de los clasificadores, el algoritmo C4.5 fue el que más tiempo demoró en generar el clasificador. Pero en el momento de resolver consultas, todos los algoritmos tomaron tiempos muy similares.

- Los criterios de Precisión y Exhaustividad, que fueron tomados para evaluar la calidad de los resultados, presentaron en la mayoría de los clasificadores un comportamiento muy similar entre su valor. Pero principalmente se destacó que los algoritmos presentaban un poco de mayor exhaustividad que precisión. Esto no afectó la decisión de elegir los algoritmos que obtuvieran una mayor precisión en cada una de las pruebas aplicadas.

  Se destaca también que la precisión alcanzo niveles mayores en los clasificadores que predecían áreas académicas de las pruebas del componente flexible. Esto se puede dar

a que en el caso de estas pruebas, cada estudiante tiene la opción de elegir el área en la que se sienta mejor capacitado, generando esto en que esta área los puntajes sean de un promedio más alto.

6.2 Trabajos futuros

- El ICFES además de poner a disposición las bases de datos de los resultados en la prueba Saber 11°, también permite el acceso a los resultados en las pruebas Saber 3°, Saber 5°, Saber 9°y SaberPro. Siguiendo una propuesta metodológica similar a la aplicada en este trabajo, se pueden construir clasificadores que permitan predecir puntajes en estas pruebas.

- Además de clasificadores, la minería de datos también permite la creación de otro tipo de utilidades que ayudan a encontrar patrones de comportamiento, por ejemplo los algoritmos de agrupamiento y los de asociación. Con las bases de datos homogeneizadas en este trabajo, se puede proceder a aplicar estos algoritmos para hallar reglas de comportamiento en los datos.

- Las instituciones académicas pueden tener la iniciativa de desarrollar proyectos similares a este, en donde los datos sean más específicos a las particularidades de sus estudiantes. Así, con una mayor precisión en la información recolectada, las predicciones tendrán una mejor calidad y se podrán aplicar con más efectividad refuerzos a los estudiantes para mejorar sus puntajes en la prueba.

Referencias

[1] Periódico El Colombiano, "Leve mejoría en pruebas Saber 11". [artículo en Internet]. http://www.elcolombiano.com/BancoConocimiento/L/leve_mejoria_en_pruebas_saber_11/leve_mejoria_en_pruebas_saber_11.asp [Consulta: 24 agosto de 2012].

[2] Periódico El Tiempo, "El 45 % de los colegios presentó bajo rendimiento en pruebas Saber 11". [artículo en Internet]. http://www.eltiempo.com/vida-de-hoy/educacion/ARTICULO-WEB-NEW_NOTA_INTERIOR-8384822.html [Consulta: 24 agosto de 2012].

[3] Revista Dinero, "Las pruebas del Icfes no son el único indicador". [artículo en Internet] http://m.dinero.com/edicion-impresa/caratula/articulo/las-pruebas-del-icfes-no-unico-indicador/139069 [Consulta: 14 marzo de 2012].

[4] Departamento Del Valle Del Cauca, Secretaria De Educación. Informe Ejecutivo Análisis Pruebas Saber 5º, 9º Y 11º. Santiago de Cali, Febrero 15 de 2012.

[5] Hernández Orallo José, Ramírez Quintana Ma. José, Ferri, Ramírez César, Introducción a la minería de datos, Person Educación, S.A. Madrid 2004, ISBN: 978-84-205-4091-7.

[6] José C. Riquelme, Roberto Ruiz, Karina Gilbert, "Minería de Datos: Conceptos y Tendencias," *Revista Iberoamericana de Inteligencia Artificial*, No. 29 (2006), pp. 11-18.

[7] ICFES, Presentación. [artículo en Internet] http://www2.icfes.gov.co/informacion-institucional/informacion-general [Consulta: 4 junio de 2012].

[8] C.S.R. Prabhu, Data Warehousing Concepts, Techniques Products and Applications, Prentice-Hall, India 2006, ISBN: 81-203-2068-9.

[9] Timarán Pereira, Ricardo. Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de la minería de datos.Revista Científica Guillermo de Ockham, vol. 8, núm. 1, enero-junio, 2010, pp. 121-130.

[10] Álvaro Jiménez Galindo, Hugo Álvarez García. Minería de Datos en la Educación. Universidad Carlos III de Madrid.

[11] Dr Ray Hoare, Using CHAID for classication problems, en New Zealand Statistical Association 2004 conference, Wellington.

[12] Sergio Valero Orea, Alejandro Salvador Vargas, Marcela García Alonso. Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo de los k vecinos más cercanos. Universidad Tecnológica de Izúcar de Matamoros, Izúcar de Matamoros, Puebla, México. Recursos digitales para la educación y la cultura volumen Kaambal. ISBN Volumen: 978-607-95446-1-4.

[13] Erika Rodallegas Ramos, Areli Torres González, Beatriz B. Gaona Couto, Erick Gastelloú Hernández, Rafael A. Lez ama Morales, Sergio Valero Orea. Modelo predictivo para la determinación de causas de reprobación mediante Minería de Datos. Universidad Tecnológica de Izúcar de Matamoros, Mexico. Recursos digitales para la educación y la cultura volumen Kaambal. ISBN Volumen: 978-607-95446-1-4.

[14] Elena Gervilla García, Rafael Jiménez López, Juan José Montaño Moreno, Albert Sesé Abad, Berta Cajal Blasco, Alfonso Palmer Pol. La metodología del Data Mining. Una aplicación al consumo de alcohol en adolescentes. Área de Metodología de las Ciencias del Comportamiento. Departamento de Psicología. Universitat de les Illes Balears. Adicciones, 2009, Vol. 21 Núm. 1, Págs. 65-80.

[15] T.Jyothirmayi et. al., An Algorithm for Better Decision Tree, International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010, 2827-2830.

[16] Téllez A., Extracción de Información con Algoritmos de Clasificación [Tesis de Maestría]. Tonantzintla, Puebla, México. Instituto Nacional de Astrofísica, Óptica y Electrónica. 2005.

[17] Christopher D., Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008, ISBN: 978-0-521-86571-5.

[18] Ross Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers. San Mateo, CA 1993, ISBN: 1-55860-238-0.

[19] Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining Practical Machine Learning Tools and Techniques - 3rd ed., Morgan Kaufmann Publishers, 2011, ISBN: 978-0-12-374856-0.

[20] Tom Dietterich, Michael Kearns, Yishay Mansour, Applying the Weak Learning Framework to Understand and Improve C4.5 en In Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 1996, ISBN: 155860-419-7.

[21] Jon Louis Bentley, Multidimensional Binary Search Trees Used for Associative Searching, Communications of the ACM Vol. 18, No. 09, 1975, 509-517.

[22] Wu et al, Top 10 Algorithms in Data Mining, Knowledge and Information Systems Vol. 14, No. 01, 2007, 1-37.

[23] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning Vol. 06, 1991, 37-66.

[24] Mitchell T.M., Machine Learning, McGraw-Hill, 1997, ISBN: 0070428077.

[25] George H. John, Pat Langley, Estimating Continuous Distributions in Bayesian Classifiers, Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers. San Mateo, CA 1995, 338-345, ISBN: 1-55860-385-9.

[26] Guoqiang Peter Zhang, Neural Networks for Classification: A Survey, IEEE Transactions on systems, man, and cybernetics, Part c: Applications and reviews, vol. 30, no. 4, november 2000, 451-462.

[27] Mohamad H. Hassoun, Fundamentals of Artificial Neural Networks, Massachusetts Institute of Technology, 1995, ISBN: 0262514672

[28] J. Park, I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation Vol. 03, 1991, 246-257.

[29] Benjamin Aumaille, J2EE Desarrollo de aplicaciones Web, Ediciones ENI, 2002, ISBN: 2-7460-1912-4.

[30] Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett, Professional Ajax, 2nd Edition, Wiley Publishing, 2007, ISBN: 978-0-470-10949-6.