

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencia Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Modalidad: Proyecto de Investigación

Minería de datos aplicada a información sobre publicaciones científicas

Trimestre: 2017 Primavera

Alumno: Adrián Campa Vilchis

Matricula: 207332696

Correo Electrónico: al207332696@azc.uam.mx

Asesor: Doctor José Alejandro Reyes Ortiz

Departamento de Sistemas

Correo Electrónico: jaro@correo.azc.uam.mx

4 de septiembre del 2017

Yo, Asesor José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



José Alejandro Reyes Ortiz

Yo, Alumno Adrian Campa Vilchis, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Adrian Campa Vilchis

Resumen

Existen una gran cantidad de publicaciones científicas de las ciencias de la computación en diferentes sitios web como DBLP [8], esto lleva dificultades al momento de consultar una publicación científica de un tema específico o seguimiento de un patrón en común.

En este Proyecto Terminal de Investigación “Minería de Datos Aplicadas a Información sobre Publicaciones Científicas, se efectuó una clasificación realizada por medio de ACM CSS (Association for Computing Machinery Computing Classification Systems) donde contiene temas o ramas de diferentes áreas de la ciencia de la computación y una organización de una gran cantidad de información de un archivo XML insertada en una base de datos para poder evaluar patrones por medio de consultas implementadas en lenguaje JAVA, también se desarrollo un diseño de vista en JSP para que el usuario pueda interactuar y obtener la información estructuradas en tablas según sus peticiones.

Tabla de Contenido

1. Introducción.....	1
2. Antecedentes	1
2.1 Proyectos terminales e integración	1
2.1.1 Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información [2].	1
2.1.2 Representación Semántica y Extracción de Información sobre Publicaciones en Expedientes Curriculares[3]......	1
2.1.3 Descubrimiento de Relaciones de Datos Abiertos de Perfiles Académicos y Profesionales mediante una Ontología [4].	1
2.2 Artículos científicos.....	2
2.2.1 Introduction to Data Base [5].	2
2.2.2 Top Cat: Data Mining for Topic Identification in a Text Corpus [6]......	2
2.3 Tesis.....	2
2.3.1 Algoritmos TDIT aplicados a la minería de datos inteligentes [7]......	2
3. Justificación	3
4. Objetivos	3
4.1 Objetivo General.....	3
4.2 Objetivos específicos	3
5. Marco Teórico.....	4
5.1 Dataming (minería de datos).....	4
5.2 Clasificación de artículos.....	4
5.3 Patrones de Diseño	4
5.4 CCS (Computing Classification Systems)	4
5.5 ACM (Association for Computing Machinery)	5
5.6 Tema Hardware de CSS	5
5.7 Tema de Computing Systems Organization de CSS (organización de sistemas informáticos)	5
5.8 Tema Networks de CSS (Redes)	5
5.9 Tema Software de CSS	5
5.10 Tema Theory of Computing de CSS (Teoría de la Computación)	5
5.11 Tema Mathematics of Computing de CSS (Matemática de la Informática).....	5
5.12 Tema Information Systems de CSS (Sistemas de Información).....	5
5.13 Tema Security and Privacy de CSS (Seguridad y Privacidad)	5
5.14 Tema Human-Centered Computing de CSS (Interacción Humanos Computadora)	5

5.15 Tema Computing Methodologies de CSS (Metodología de la computación)	5
5.16 Tema Applied Computing de CSS (Informática Aplicada)	6
5.17 Tema Social and Professional Topics de CSS (Temas Sociales y Profesionales)	6
5.18 Parser Dom	6
5.19 JSP (Java Server Pages)	6
5.20 Java Servlet.....	6
5.21 XML	6
5.22 Netbeans y Eclipse.....	6
5.23 Servidor GlassFish	6
5.24 API Java Database Connectivity (JDBC)	6
6. Desarrollo del Proyecto	7
6.1 Base de Datos.....	7
6.2 Módulo de integración	10
6.3 Módulo de Minería de Datos.....	19
6.4 Módulo de Evaluación de Patrones	22
6.5 Módulo de Presentación	24
7 Resultados	28
7.1 Pruebas para la base de datos	28
7.2 Pruebas para el Módulo de Integración.....	31
7.3 Pruebas para el Módulo de Minería de Datos	34
7.4 Pruebas para el Módulo de Evaluación de Patrones.....	38
7.5 Pruebas para el Módulo de Presentación.....	42
8 Análisis y Discusión de Resultados.....	44
8.1 Análisis de resultados de la Base de Datos	44
8.2 Análisis de resultados del Módulo de Integración.....	46
8.3 Análisis de resultados del Módulo de Minería de Datos	49
8.4 Análisis de resultados del Módulo de Evolución de Patrones	50
8.5 Análisis de resultados del Módulo de Presentación.....	52
9 Conclusiones.....	53
10 Referencias Bibliográficas	54
Apéndices	56

Índice de Figuras

Figura 1. Primer Bosquejo de las Tablas Entidad-Relación.....	7
Figura 2. Ejemplo de Artículo de Publicaciones Científicas del archivo DBLP.xml	8
Figura 3. Tablas Entidad-Relación del Proyecto Terminal	9
Figura 4. Tabla articulo con signos de puntuación.....	10
Figura 5. Tabla autor con signos de puntuación.....	11
Figura 6. Tabla autor llenada correctamente	12
Figura 7. Tabla de artículo llenada correctamente.....	12
Figura 8. Tabla de artículo llenada correctamente	13
Figura 9. Tabla de artículo llenada correctamente	13
Figura 10. Relación entre la tabla artículo y la tabla autor.	14
Figura 11. Consulta 4 efectuada en Mysql Worbench.	15
Figura 12. Consulta 5 efectuada en Myqsl Worbench con los datos del articulo	15
Figura 13. Consulta 5 efectuada en Myqsl Worbench con los datos del autor	16
Figura 14. Consulta 6 efectuada en Msqsl Worbench mostrando los datos de los autores	17
Figura 15. Consulta 6 efectuada en Mysql Worbench con la información de las publicaciones científicas	17
Figura 16. Consulta 7 efectuada en Mysql Worbench	18
Figura 17. Tabla acm temas insertado correctamente	19
Figura 18. Tabla articulo con las modificaciones mencionadas anteriormente.....	20
Figura 19. Título de ejemplo para la clasificación	20
Figura 20. Ejemplo de cómo clasificar.	21
Figura 21. Procedimiento para la evaluación de patrones.....	23
Figura 22. Vista Principal del Proyecto Terminal.....	24
Figura 23. Menú de Consultas del Proyecto Terminal.....	25
Figura 24. Vista de Consulta para la Clasificación.	26
Figura 25. Vista de consulta para los Publicaciones Científicas	26
Figura 26. Vista para mostrar los datos de las consultas.....	27
Figura 27. Publicación Científica donde se modificaron los autores	28
Figura 28. Resultado de las publicaciones científicas modificadas por autor	28
Figura 29. Publicación Científica donde no contiene el nodo cadena para la prueba	29
Figura 30. Resultado de la publicación científica no contiene el nodo ee	29

Figura 31. Publicación Científica donde no contiene el nodo numero para la prueba	30
Figura 32. Resultado de la publicación científica no contiene el nodo número.....	30
Figura 33. Publicación científica en donde se muestra un carácter especial en el nodo autor	31
Figura 34. Realización de la consulta de prueba.....	31
Figura 35. Visualización de la consulta	32
Figura 36. Realización de la consulta con cadenas con signos de puntuación.....	32
Figura 37. Visualización de la consulta sin encontrar resultado.	33
Figura 38. Primera Consulta de prueba para la clasificación	34
Figura 39. Visualización de resultados de la primera consulta	34
Figura 40. Verificación por medio de Mysql Workbench de la consulta efectuada anteriormente ...	35
Figura 41. Segunda Consulta de prueba para la clasificación.....	36
Figura 42. Visualización de resultados de la segunda consulta	36
Figura 43. Verificación por medio de Mysql Workbench de la consulta efectuada anteriormente ...	37
Figura 44. Primera Consulta para la prueba de evaluación de patrones por titulo.....	38
Figura 45. Visualización de resultados de la primera consulta	38
Figura 46. Segunda Consulta para la prueba de evaluación de patrones por nombre de autor	39
Figura 47. Visualización de resultados de la segunda consulta	40
Figura 48. Segunda Consulta para la prueba de evaluación de patrones por número de ID	41
Figura 49. Visualización de resultados de la tercera consulta.....	41
Figura 50. Consulta realizada para la prueba del módulo de presentación	42
Figura 51. Visualización del resultado de la consulta	43
Figura 52. Consulta de la publicación científica que no contiene la cadena ee	45
Figura 53. Consulta de la publicación científica que no contiene la cadena numero.....	45
Figura 54. Respuesta del servidor GlassFish de la consulta de prueba sin signo de puntuación	46
Figura 55. Respuesta del servidor Glassfish de la consulta de prueba con signo de puntuación	47
Figura 56. Respuesta del servidor Glassfish de la consulta con los datos corregidos.....	48
Figura 57. Consulta implementada en lenguaje Java para la clasificación	49
Figura 58. Relación entre la tabla acm_temas con articulo.....	49
Figura 59. Evaluación de consultas por medio de titulo	50
Figura 60. Evaluación de consultas por medio de numero de id.....	51
Figura 61. Evaluación de consultas por medio de autor	51
Figura 62. Evaluación de consultas por medio de nombre del tema	52

1. Introducción

En la actualidad, se genera demasiada información, por lo que se necesita tenerla organizada y centralizada. Esto ocasiona un problema en manejarla de manera eficiente; para solucionar este problema se tuvo la necesidad de mejorar las Bases de Datos Estructuradas y Sistemas de Gestión de Bases de Datos (DBMS).

Debido a la enorme cantidad de datos almacenados que hay alrededor del mundo, es necesario analizar y buscar ciertos patrones para poder organizarlos de una manera comprensible y efectiva para que nos ayude a una toma de decisión adecuada.

Un ejemplo serían las publicaciones científicas; de acuerdo a la página oficial de Conacyt [1] que se realizan alrededor de 322 publicaciones en México, por lo cual es esencial categorizar para su fácil estudio.

2. Antecedentes

2.1 Proyectos terminales e integración

2.1.1 Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información [2].

El proyecto consiste en diseñar una aplicación en minería de datos que integre diferentes estrategias para corregir datos ausentes y/o incoherentes en una base de datos.

La similitud con esta propuesta consiste en emplear diferentes técnicas de minería de datos para tratamiento de información.

La diferencia con esta propuesta: se obtendrá la extracción de información relevante para ver la relación entre ellas.

2.1.2 Representación Semántica y Extracción de Información sobre Publicaciones en Expedientes Curriculares[3].

Por medio de investigación de ontología se extraerá información sobre publicación a partir de expedientes curriculares en español utilizando técnicas de minería de texto.

La similitud con esta propuesta es: se extrajo información a partir de una fuente de información.

La diferencia con esta propuesta: se realizó con publicaciones científicas y no con publicaciones en expediente curriculares.

2.1.3 Descubrimiento de Relaciones de Datos Abiertos de Perfiles Académicos y Profesionales mediante una Ontología [4].

En esta propuesta permitirá descubrir relaciones entre datos de los académicos y de los profesionales mediante ontologías y reglas que analiza el contenido de las fuentes de datos.

La similitud con esta propuesta: se analizó contenido de fuente de datos. La diferencia con esta propuesta: no se utilizará ontología en el desarrollo.

2.2 Artículos científicos

2.2.1 Introduction to Data Base [5].

Es un artículo informativo donde nos introducen a las minerías de datos y nos da un panorama de donde se obtiene la información para desarrollar las bases de datos para el uso del procedimiento.

La similitud con esta propuesta es para un conjunto masivo de datos obtenidos de las publicaciones. La diferencia con esta propuesta es: solo se obtuvo de la información sobre una fuente específica (publicaciones científicas)

2.2.2 Top Cat: Data Mining for Topic Identification in a Text Corpus [6].

En este artículo nos introduce a técnicas que se usa para identificar los temas que se repiten en los artículos de texto.

La similitud con esta propuesta es: se toma como referencia en estas técnicas para poder encontrar las relaciones entre las publicaciones.

La diferencia con esta propuesta: no se elaborará en artículos de noticias si no en publicaciones científicas.

2.3 Tesis

2.3.1 Algoritmos TDIT aplicados a la minería de datos inteligentes [7].

Esta tesis aborda el problema de inducir árboles de decisiones por medio de un proceso de aprendizaje automático (Top Down Induction Tree, TDIT) en minería de datos, que se usa para modelarlas clasificaciones de los datos mediante árboles de decisiones.

La similitud con esta propuesta es: se utiliza como referencia los árboles de decisión es para la clasificación de los datos.

La diferencia con esta propuesta es: no se realizará tan general como en la tesis si no particularmente en una fuente de datos.

3. Justificación

El profesor de la Universidad Autónoma Metropolitana de la Unidad Azcapotzalco necesita apoyo para clasificar ciertas publicaciones científicas para su fácil estudio; por eso mismo se propone implementar por medio de técnicas de minería de datos para distinguir las similitudes entre las distintas publicaciones y obtener la información más relevante de éstas.

4. Objetivos

En este apartado se mostraran los objetivos generales y los objetivos específicos que se siguieron para realizar el proyecto terminal de investigación.

4.1 Objetivo General

- ✓ Implementar un sistema de minería de datos para la organización para las clasificaciones relevantes en publicaciones de revistas científicas

4.2 Objetivos específicos

- Diseñar e implementar la base de datos de las publicaciones científicas.
- Diseñar e implementar módulo de integración con los datos estructurados.
- Implementar las técnicas de clasificación y agrupamiento.
- Evaluar los patrones con las técnicas de clasificación y agrupamiento.
- Representar los datos por una interfaz por medio de tablas.

5. Marco Teórico

En este proyecto terminal de investigación se tomó como bases los antecedentes y se investigó nuevos conceptos e información para poder alcanzar los objetivos.

De las investigaciones efectuadas se encontró la forma de clasificar las publicaciones científicas se usó como una técnica de Minería de Datos un documento que contiene temas o ramas de la ciencia de la computación, este documento es ACM CCS (Association For Computing Machinery Computing Classification Systems) [9] de la versión del 2012 en donde en su taxonomía contiene 14 temas para poder clasificar en este proyecto solo se usaran 12 temas cuales son: Hardware, Computing Systems Organization, Networks, Software, Theory of Computing, Mathematics of Computing, Information Systems, Security and Privacy, Human-Centered Computing, Computing Methodologies, Applied Computing y Social and Professional Topics los que no se tomaron en cuenta (General and Reference y Proper nouns: People, Technologies and Companies) ya que se hace referencia a los temas anteriores.

También se realizó una investigación sobre como analizar y transformar los nodos del archivo XML a cadenas (String) para que puedan ser insertadas en la base de datos se encontró el Parser adecuado para realizar la tarea (Dom).

Se usaron vista creada en JSP (Java Server Pages) con lenguaje de HTML realizadas en la plataforma Netbeans. También se usó Java Servlet para poder conectar las peticiones con las consultas realizadas en el lenguaje JAVA y las vistas creada en JSP.

Se utilizó la plataforma Netbeans en vez de usar la plataforma Eclipse como se propuso en la propuesta ya que se tenía que realizar vista y Netbeans se pude desarrollar mejor que Eclipse donde se utilizó el servido GlassFish para la visualización de las vistas web.

La conexión hacia la base de datos se realizó con el api JDBC Java Database Connectivity para realizar la conexión desde JAVA hacia la base de datos.

5.1 Dataming (minería de datos)

Es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos de manera automática y semiautomática con el objetivo de encontrar patrones repetitivos, tendencia o reglas que permite la explicación de los comportamientos de los datos. [10]

5.2 Clasificación de artículos

Es la búsqueda de un todo de todas aquellas cosas (en este caso artículos) que guarden o que compartan un tipo de relación para así agruparlos para buscar un determinado elemento sea más fácil de buscar [11]

5.3 Patrones de Diseño

Son aquellos que definen la solución para resolver problemas similares, facilitando además la reutilización del código [12]

5.4 CCS (Computing Classification Systems)

Es un sistema de clasificado computacional diseñado por ACM es poli-jerárquica con términos de conceptos múltiples [9]

5.5 ACM (Association for Computing Machinery)

Es una organización a nivel mundial que reúne a todos los profesionales, estudiantes, investigadores y a cada persona interesada en ciencias de la computación facilita varios recursos para toda su comunidad con el fin de que la computación crezca como ciencia y como profesión [20]

5.6 Tema Hardware de CSS

Es un tema o rama individual del CSS con conceptos encontrados bajo circuito de computadoras, estructura de memoria, estructura lógica, entre otros [9]

5.7 Tema de Computing Systems Organization de CSS (organización de sistemas informáticos)

Es un tema o rama individual del CSS dividido en cuatro conceptos sistemas basados en aplicaciones, sistemas embebidos sistema multicore y sistemas en tiempo real como los sistemas distribuidos [9]

5.8 Tema Networks de CSS (Redes)

Esta nueva rama o tema de alto nivel contiene una mezcla de términos nuevos y conceptos existentes extraídos de las redes de comunicación informática [9]

5.9 Tema Software de CSS

Esta nueva rama o tema de alto nivel contiene una mezcla de términos nuevos y conceptos existentes extraídos de las redes de comunicación informática [9]

5.10 Tema Theory of Computing de CSS (Teoría de la Computación)

Es un tema o rama individual del CSS contiene conceptos relacionados con el diseño y análisis de algoritmos como las lenguas formales y la lógica matemática [9]

5.11 Tema Mathematics of Computing de CSS (Matemática de la Informática)

Es un tema o rama individual del CSS incluyen conceptos bajo Algoritmos numéricos y problemas [9]

5.12 Tema Information Systems de CSS (Sistemas de Información)

Es un tema o rama individual del CSS donde reflejan sistemas de información como las bases de datos términos relacionados con la Web y los dispositivos multimedia [9]

5.13 Tema Security and Privacy de CSS (Seguridad y Privacidad)

Esta nueva rama incorpora conceptos Políticas de privacidad y políticas de seguridad [9]

5.14 Tema Human-Centered Computing de CSS (Interacción Humanos Computadora)

Es un tema o rama individual del CSS contiene términos relacionados con interfaces, computación personal como La interacción persona-ordenador, visualización de información, computación ubicua y móvil [9]

5.15 Tema Computing Methodologies de CSS (Metodología de la computación)

Esta rama contiene Inteligencia Artificial, Aplicaciones y sistemas expertos como la Computación Gráfica [9]

5.16 Tema Applied Computing de CSS (Informática Aplicada)

Es un tema o rama individual del CSS que contiene el concepto de Sistemas de información empresarial como el procesamiento de datos administrativos, los ordenadores en otros sistemas y uso de computadoras en la educación [9]

5.17 Tema Social and Professional Topics de CSS (Temas Sociales y Profesionales)

Es un tema o rama individual del CSS donde incluye la industria de la informática, historia de la informática, computadoras y sociedad, profesión informática y Gestión de sistemas informáticos y de información [9]

5.18 Parser Dom

Es el Modelo de Objetos de Documento que define una interfaz que permite a los programas acceder y actualizar el estilo, la estructura y el contenido de los documentos XML [19].

5.19 JSP (Java Server Pages)

Es una tecnología orientada a crear páginas web con programación en Java, crear aplicaciones web que se ejecuten en varios servidores web, de múltiples plataformas. Están compuestas de código HTML/XML mezclado para programar scripts de servidor en sintaxis Java [13]

5.20 Java Servlet

Es una tecnología utilizada para crear aplicaciones web que amplía las capacidades de los servidores y responde a la petición entrante. Puede responder a cualquier tipo de solicitudes. Es un componente web que se implementa en el servidor para crear una página web dinámica [15]

5.21 XML

Es un meta-lenguaje que permite definir etiquetas personalizadas para descripción y organización de datos. Por lo tanto, un archivo XML puede incluir cualquier flujo de datos basado en texto: un artículo de una revista. [16]

5.22 Netbeans y Eclipse

Son entornos de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. [18]

5.23 Servidor GlassFish

Es un servidor de aplicaciones que implementa la plataforma JavaEE5 (una forma estándar de desarrollar Aplicaciones Java Empresariales, es decir, que puedan ser utilizadas en más de un servidor sin importar el fabricante) por lo que soporta las últimas versiones de tecnologías como: JSP, Servlets, Enlaces XML, entre otros. [17]

5.24 API Java Database Connectivity (JDBC)

Proporciona acceso universal a los datos desde el lenguaje de programación Java, puede acceder virtualmente a cualquier fuente de datos, desde bases de datos relacionales a hojas de cálculo y archivos planos. También proporciona una base común en la que se pueden construir herramientas e interfaces alternativas. [14]

6. Desarrollo del Proyecto

6.1 Base de Datos

En la Descripción Técnica de la propuesta se especifica una base de datos donde se desarrollo un bosquejo de tabla y su implementación en el lenguaje SQL, así como los datos estructurados y organizados de las publicaciones científicas del archivo artículos.xml

Al principio se identificó las entidades de un artículo denominado “Descubrimiento y representación de conocimiento sobre datos científicos y académicos” [8]. , en donde nos explica cómo obtener información relevante por medio de ontologías sobre las publicaciones científicas que se pueden encontrar en diferentes recursos web, en este artículo se tomó de base el modelo ontológico donde se representa la información de las publicaciones científicas para la elaboración previa de las tablas entidad-relación de la bases de datos que se fue modificando en el desarrollo del proyecto terminal.

En la **Figura 1** se muestra el primer Bosquejo de las Tablas Entidad-Relación obtenida de la información del artículo, en donde se especifican las llaves foráneas, primarias, carnalidad y normalidad.

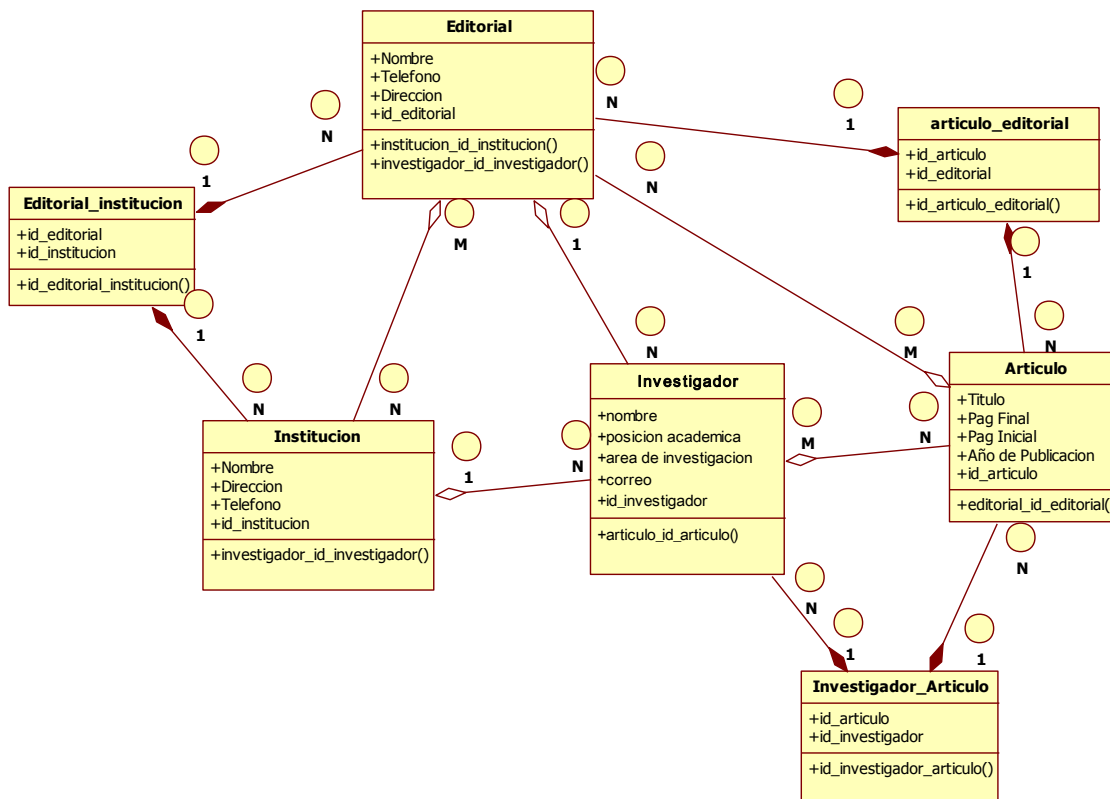


Figura 1. Primer Bosquejo de las Tablas Entidad-Relación

Mientras fue desarrollando la base de datos para el proyecto “Minería de datos aplicada a información sobre publicaciones científicas” se fue modificando las tablas entidad-relación por medio de la información de un archivo DLBP.xml en donde es un archivo obtenido de un sitio web (DBLP) [8] que contiene información bibliográfica de revistas sobre el área de las ciencias de la computación.

En la **Figura 2**, se muestra un ejemplo de cómo está formado el archivo DBLP.xml

```
<!DOCTYPE dblp SYSTEM "dblp.dtd">
<dblp>
<article mdate="2011-01-11" key="journals/acta/Saxena96">
<author>Sanjeev Saxena</author>
<title>Parallel Integer Sorting and Simulation Amongst CRCW Models.</title>
<pages>607-619</pages>
<year>1996</year>
<volume>33</volume>
<journal>Acta Inf.</journal>
<number>7</number>
<url>db/journals/acta/acta33.html#Saxena96</url>
<ee>http://dx.doi.org/10.1007/BF03036466</ee>
</article>
<article mdate="2011-01-11" key="journals/acta/Simon83">
<author>Hans-Ulrich Simon</author>
<title>Pattern Matching in Trees and Nets.</title>
<pages>227-249</pages>
<year>1983</year>
<volume>20</volume>
<journal>Acta Inf.</journal>
<url>db/journals/acta/acta20.html#Simon83</url>
<ee>http://dx.doi.org/10.1007/BF01257084</ee>
</article>
<article mdate="2011-01-11" key="journals/acta/Goodman883">
<author>Nathan Goodman</author>
<author>Oded Shmueli</author>
<title>NP-complete Problems Simplified on Tree Schemas.</title>
<pages>171-178</pages>
<year>1983</year>
```

Figura 2. Ejemplo de Artículo de Publicaciones Científicas del archivo DBLP.xml

Donde se modificaron los atributos, la carnalidad, la normalidad, como las llaves foráneas y primarias, así como creando una tabla de autores para mejorar la manipulación de los datos.

Por último se creó una tabla para almacenar los temas para clasificar los artículos científicos ya no se elaboró por medio de árboles de decisiones como se propuso en la propuesta si no se hizo por medio de un sistema de clasificación computacional ACM (Computing Classification Systems) [9] en donde contiene temas individuales en términos generales (Software, Hardware, Interacción Humano Computadora, entre otras) para poder clasificar en un enfoque general los artículos de publicaciones científicas sobre el área de las ciencias de la computación.

En la **Figura 3** se muestran las tablas de entidad relación modificadas para diseñar la base de datos que se usó para este proyecto terminal de investigación.

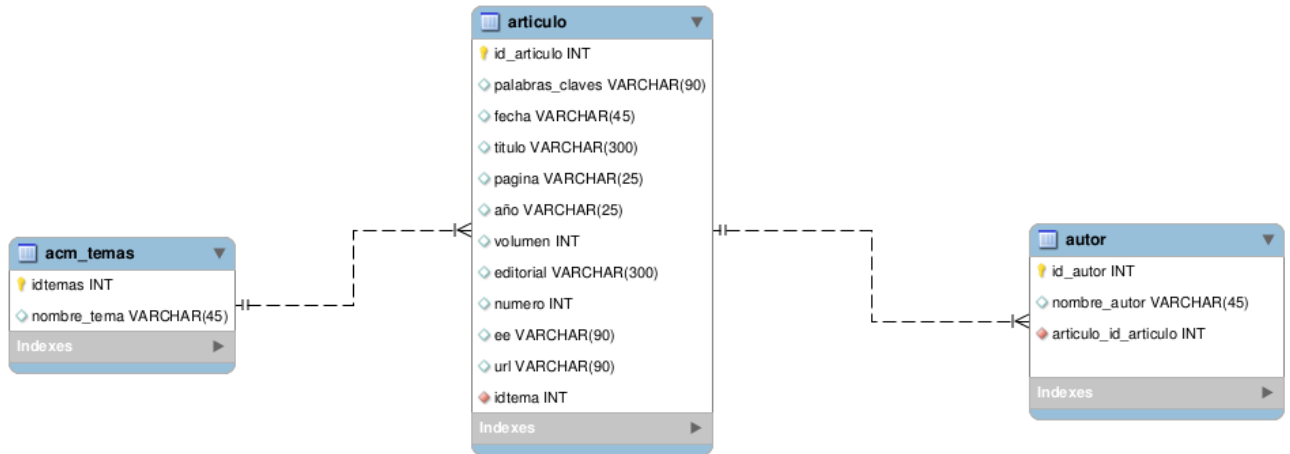


Figura 3. Tablas Entidad-Relación del Proyecto Terminal

Las tablas entidad-relación se implementó por medio del lenguaje SQL en Worbench Mysql 6 o posterior, en el **Apéndice 1** se muestra el código de diseño de las tablas.

Al concluir con el diseño de las tablas y de la base de datos, se implementó la conexión hacia la base de datos por medio del lenguaje JAVA utilizando la API JDBC “mysql-connector-java-5.1.16-bin.jar”.

En el **Apéndice 2** se muestra el programa de implementación de conexión a la base de datos en Worbench MYSQL desde JAVA

A continuación de implemento los VO, articuloVO que hacen la referencia a las columnas de la tabla articulo **Apéndice 3** y autor VO que hacen la referencia a las columnas de la tabla autor **Apéndice 4** para poder realizar la inserción correctamente en las tablas.

Posteriormente se diseñaron los DAO para la inserción hacia la base de datos por medio de la QUERY INSERT INTO desde el lenguaje JAVA.

En el **Apéndice 5** se muestra el programa del DAO para inserción de la tabla artículo denominado insertar_articulo.

En el **Apéndice 6** se muestra el programa del DAO para inserción de la tabla autor denominado insertar_autor.

Después de implementar los respectivos DAO y VO, se realizó el relleno de tablas por medio de un diseño de un Parser DOM en el Lenguaje JAVA (para la transformación del archivo XML a cadenas de tipo String para que su inserción fuera un éxito en las tablas de la base de datos) donde se utilizaron 300 artículos (artículos.xml) del archivo DBLP.xml que contiene un aproximado 3000 artículos para realizar las pruebas.

6.2 Módulo de integración

En este módulo se desarrolló en lo datos estructurados y organizados de la base de datos consultas necesarias para la verificación de los datos que estén incoherentes e incompletos.

En la **Consulta 1** y **Consulta 2** se realizaron para averiguar si los datos fueron insertados correctamente en la base de datos.

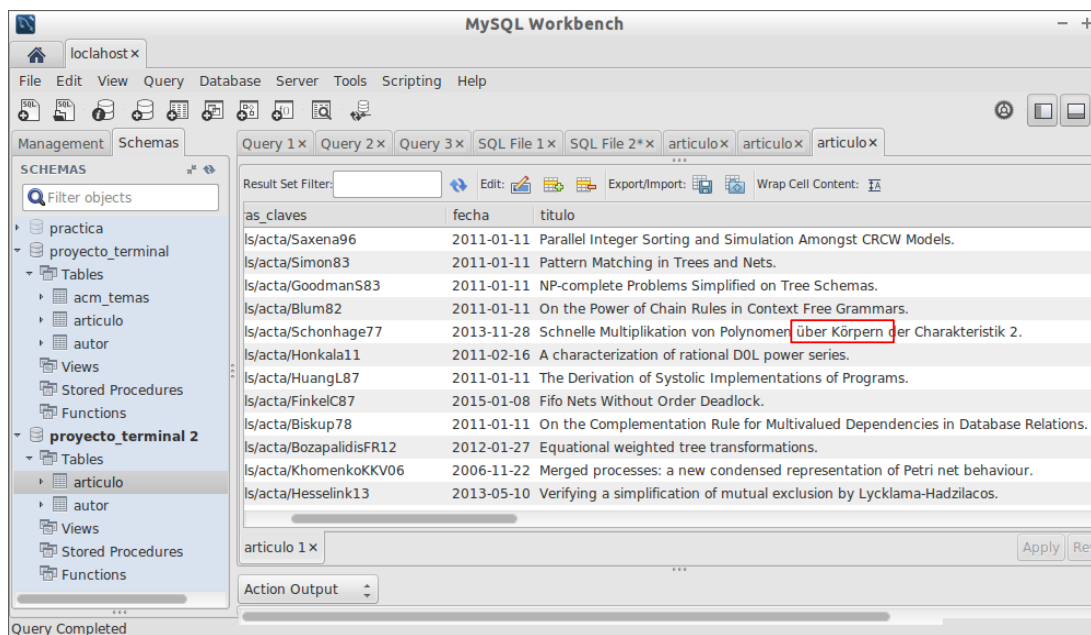
```
SELECT *FROM Proyecto_Terminal.autor;
```

Consulta 1. Selección de la tabla autor

```
SELECT *FROM Proyecto_Terminal.articulo;
```

Consulta 2. Selección de la tabla articulo

En las **Figura 4** y **Figura 5** se muestran los datos insertados en las tablas con el detalle que se insertan con signos de puntuación tanto en los artículos como en los autores



as_claves	fecha	titulo
ls/acta/Saxena96	2011-01-11	Parallel Integer Sorting and Simulation Amongst CRCW Models.
ls/acta/Simon83	2011-01-11	Pattern Matching in Trees and Nets.
ls/acta/GoodmanS83	2011-01-11	NP-complete Problems Simplified on Tree Schemas.
ls/acta/Blum82	2011-01-11	On the Power of Chain Rules in Context Free Grammars.
ls/acta/Schonhage77	2013-11-28	Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2.
ls/acta/Honkala11	2011-02-16	A characterization of rational DOL power series.
ls/acta/HuangL87	2011-01-11	The Derivation of Systolic Implementations of Programs.
ls/acta/FinkelC87	2015-01-08	Fifo Nets Without Order Deadlock.
ls/acta/Biskup78	2011-01-11	On the Complementation Rule for Multivalued Dependencies in Database Relations.
ls/acta/BozapalidisFR12	2012-01-27	Equational weighted tree transformations.
ls/acta/KhomenkoKKV06	2006-11-22	Merged processes: a new condensed representation of Petri net behaviour.
ls/acta/Hesselink13	2013-05-10	Verifying a simplification of mutual exclusion by Lycklama-Hadzilacos.

Figura 4. Tabla articulo con signos de puntuación

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'practica', 'proyecto_terminal', and 'proyecto_terminal 2'. Under 'proyecto_terminal', the 'autor' table is selected. The main window displays the 'autor' table with the following data:

#	id_autor	nombre_autor	articulo_id_articulo
1	1	Sanjeev Saxena	1
2	2	Hans-Ulrich Simon	2
3	3	Nathan Goodman	3
4	4	Oded Shmueli	3
5	5	Norbert Blum	4
6	6	Arnold Schönbage	5
7	7	Juha Honkala	6
8	8	Chua-Huang Huang	7
9	9	Christian Lengauer	7
10	10	Alain Finkel	8
11	11	Annie Choquet	8
12	12	Joachim Biskup	9
13	13	Symeon Bozapalidis	10

Figura 5. Tabla autor con signos de puntuación

Se modificó el Parser DOM con el código, **Apéndice 7**, para poder quitar los signos de puntuación y poder resolver los problemas de las consultas programadas en JAVA se explicará con más detalle en las pruebas de módulo de integración. En el **Apéndice 8** se muestra el programa del Parser DOM modificado con la especificación mencionada anteriormente

En la **Figura 6** se muestra la **Consulta 1** efectuada mostrando que si se insertó los datos correctamente en la tabla autor de la base de datos proyecto terminal con la información de las publicaciones científicas que contiene el archivo artículos.xml

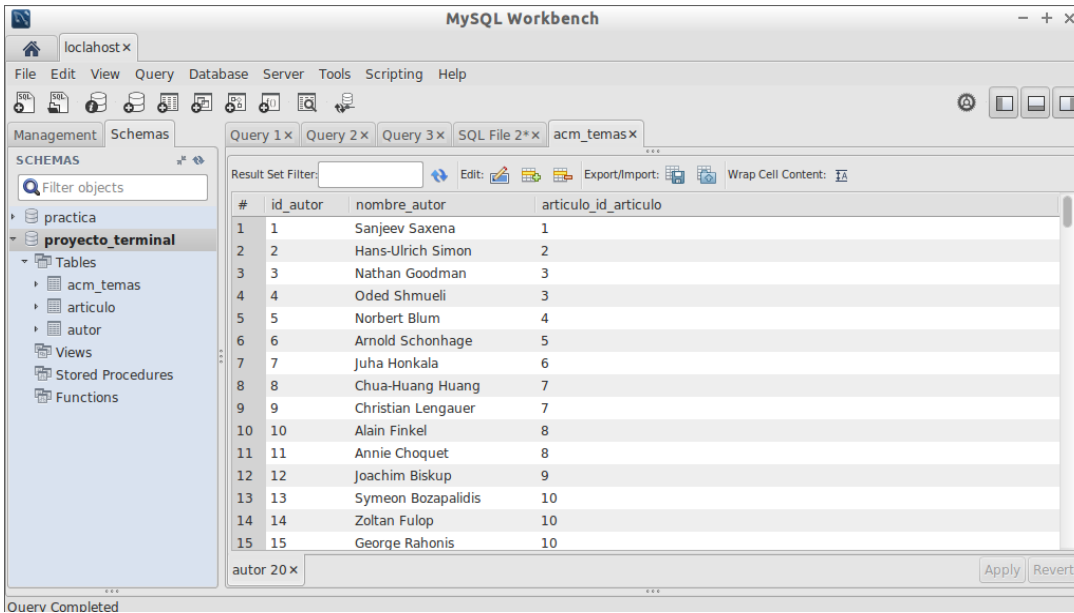


Figura 6. Tabla autor llenada correctamente

En las **Figura 7**, **Figura 8** y **Figura 9** se muestra la **Consulta 2** efectuada mostrando que si se insertó los datos correctamente en la tabla artículo de la base de datos proyecto terminal con la información de las publicaciones científicas que contiene el archivo artículos.xml

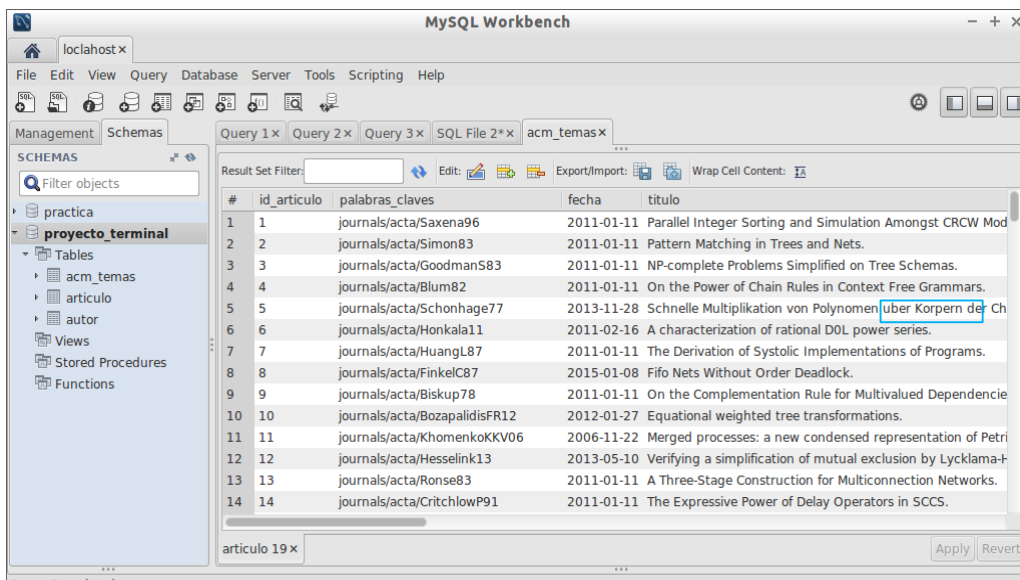


Figura 7. Tabla de artículo llenada correctamente.

MySQL Workbench interface showing a table named 'articulo' with the following data:

pagina	año	volumen	editorial	numero	ee
607-619	1996	33	Acta Inf.	7	http://dx.doi.org/10.1007/BF00289416
227-248	1983	20	Acta Inf.	0	http://dx.doi.org/10.1007/BF00289414
171-178	1983	20	Acta Inf.	0	http://dx.doi.org/10.1007/BF00264161
425-433	1982	17	Acta Inf.	0	http://dx.doi.org/10.1007/BF00289470
395-398	1977	7	Acta Inf.	0	http://dx.doi.org/10.1007/s00236-010-0128-1
19-24	2011	48	Acta Inf.	1	http://dx.doi.org/10.1007/BF00282618
595-632	1987	24	Acta Inf.	6	http://dx.doi.org/10.1007/BF00268843
15-36	1988	25	Acta Inf.	1	http://dx.doi.org/10.1007/BF00264322
297-305	1978	10	Acta Inf.	0	http://dx.doi.org/10.1007/s00236-011-0148-5
29-52	2012	49	Acta Inf.	1	http://dx.doi.org/10.1007/s00236-006-0023-y
307-330	2006	43	Acta Inf.	5	http://dx.doi.org/10.1007/s00236-013-0178-2
199-228	2013	50	Acta Inf.	3	http://dx.doi.org/10.1007/BF00289416
197-206	1983	20	Acta Inf.	0	http://dx.doi.org/10.1007/BF01178582
447-452	1991	28	Acta Inf.	5	http://dx.doi.org/10.1007/BF01178582

Figura 8. Tabla de artículo llenada correctamente

MySQL Workbench interface showing a table named 'articulo' with the following data:

ro	ee	url
	http://dx.doi.org/10.1007/BF03036466	db/journals/acta/acta33.html#Saxena96
	http://dx.doi.org/10.1007/BF01257084	db/journals/acta/acta20.html#Simon83
	http://dx.doi.org/10.1007/BF00289414	db/journals/acta/acta20.html#Goodman5
	http://dx.doi.org/10.1007/BF00264161	db/journals/acta/acta17.html#Blum82
	http://dx.doi.org/10.1007/BF00289470	db/journals/acta/acta7.html#Schonhage
	http://dx.doi.org/10.1007/s00236-010-0128-1	db/journals/acta/acta48.html#Honkala11
	http://dx.doi.org/10.1007/BF00282618	db/journals/acta/acta24.html#HuangL87
	http://dx.doi.org/10.1007/BF00268843	db/journals/acta/acta25.html#FinkelC87
	http://dx.doi.org/10.1007/BF00264322	db/journals/acta/acta10.html#Biskup78
	http://dx.doi.org/10.1007/s00236-011-0148-5	db/journals/acta/acta49.html#Bozapalidi
	http://dx.doi.org/10.1007/s00236-006-0023-y	db/journals/acta/acta43.html#Khomenkc
	http://dx.doi.org/10.1007/s00236-013-0178-2	db/journals/acta/acta50.html#Hesselink1
	http://dx.doi.org/10.1007/BF00289416	db/journals/acta/acta20.html#RonseB3
	http://dx.doi.org/10.1007/BF01178582	db/journals/acta/acta28.html#CritchlowP

Figura 9. Tabla de artículo llenada correctamente

Después de realizar las consultas en la **Figura 10** se verifica que tanto las tablas artículo como la de autor tuvieron la relación con la llave primaria de la tabla artículo (id artículo) y su llave foránea de la tabla autor (artículo id artículo) para demostrar que la inserción fueron hechas simultáneamente.

	autor	artículo
	Hans-Ulrich Simon (2)	1 1 journals/acta/Saxena96
	Nathan Goodman (3)	2 2 journals/acta/Simon83
	Oded Shmueli (3)	3 3 journals/acta/Goodman583
	Norbert Blum (4)	4 4 journals/acta/Blum82
	Arnold Schonhage (5)	5 5 journals/acta/Schonhage77
	Juha Honkala (6)	6 6 journals/acta/Honkala11
	Chua-Huang Huang (7)	7 7 journals/acta/HuangL87
	Christian Lengauer (7)	8 8 journals/acta/FinkelC87
	Alain Finkel (8)	9 9 journals/acta/Biskup78
		10 10 journals/acta/BozapalidisFR12

Figura 10. Relación entre la tabla artículo y la tabla autor.

Después de verificar los datos insertado correctamente se realizó la **Consulta 3** para la unión de la tabla artículo con la tabla autor para comprobar posteriormente si las consultas que se programaron en JAVA se efectuaron correctamente

```
Select id_articulo, palabras_claves, fecha, titulo, pagina,
año,volumen,editorial,ee,url, nombre_autor from articulo, autor where
articulo.id_articulo=autor.articulo_id_articulo;
```

Consulta 3. Unión de la tabla autor y articulo por medio de Lenguaje SQL

Así se efectuaron la **Consulta 4**, **Consulta 5**, **Consulta 6** y **Consulta 7** como guía para la implementación de las consultas efectuadas en el lenguaje JAVA y para comprobación de los datos almacenados en las tablas, que a continuación se mencionaran.

```
SELECT * FROM autor where articulo_id_articulo = 45;
```

Consulta 4. Comprobación de relación entre autor y artículo.

En la **Figura 11** se muestra la relación correcta del artículo con número de id 45 tiene relacionado tres autores con su correspondiente número de id de la tabla autor.

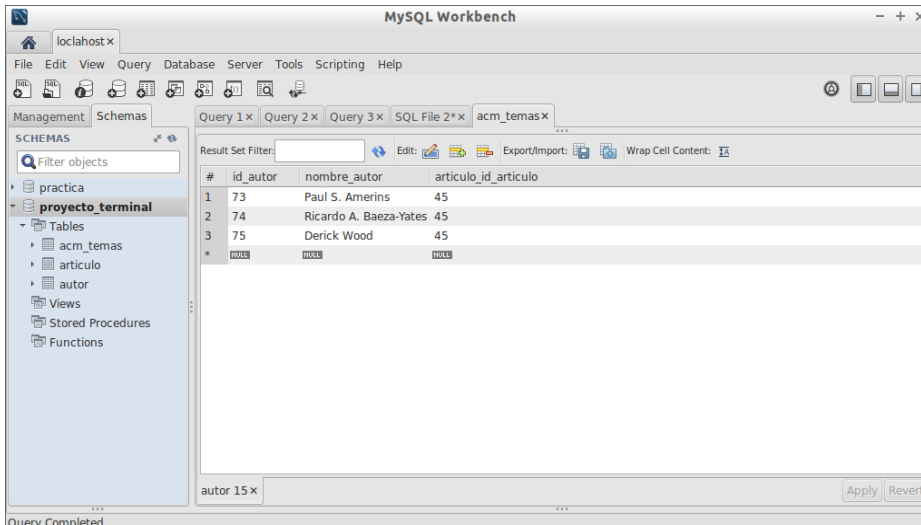


Figura 11. Consulta 4 efectuada en Mysql Worbench.

```
Select id_articulo, palabras_claves, fecha, titulo, pagina,
año,volumen,editorial,ee,url, nombre_autor from articulo, autor where
articulo.id_articulo=autor.articulo_id_articulo and id_articulo = 101;
```

Consulta 5. Consulta guía para la implementación de las consultas en el lenguaje JAVA

En las **Figuras 12 y Figura 13** se muestra la información del artículo con sus respectivos autores por la **Consulta 5** efectuada

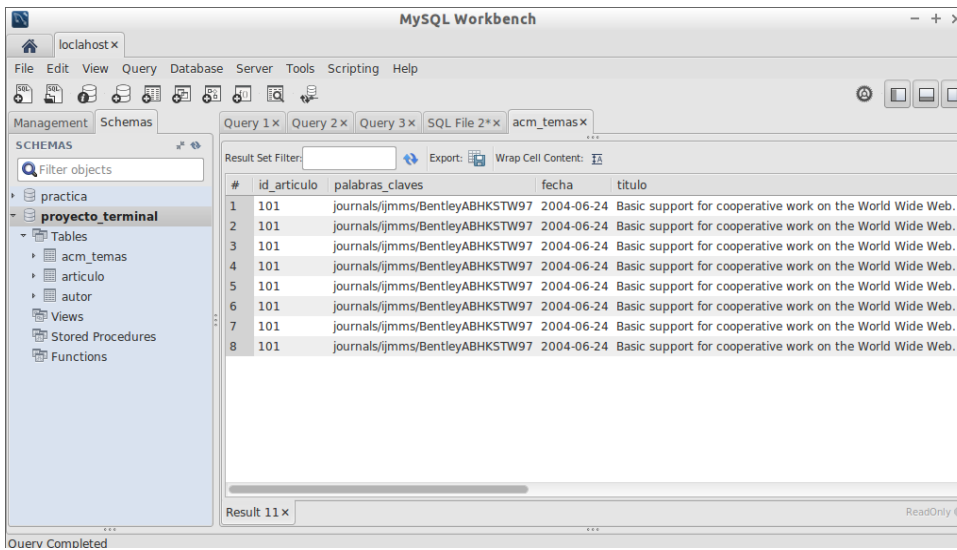


Figura 12. Consulta 5 efectuada en Myqsl Worbench con los datos del articulo

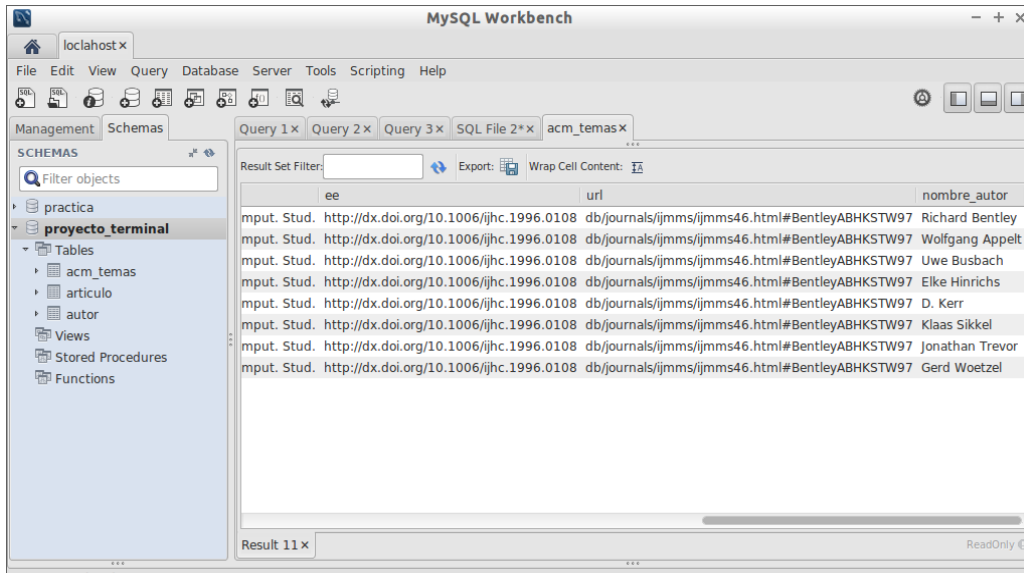


Figura 13. Consulta 5 efectuada en Myqsl Worbench con los datos del autor

Se analizó que las consultas por medio de atributos de tipo enteros se realizan rápido por medio de la **Consulta 4** y **Consulta 5**, pero al momento de analizar atributos de tipo String se complicaba ya que se tenía que poner el nombre de la cadena (título completo, nombre autor completo, fecha completa) para realizar la búsqueda.

Para corregir este problema se realizó **Consulta 6** por medio de la sentencia LIKE para que sea más fácil la búsqueda de información de atributos de tipo cadenas en las tablas de la bases de datos y así mejorando la sintaxis hacia las consulta que se implementaron en el lenguaje JAVA.

```
Select id_articulo, palabras_claves, fecha, titulo, pagina,
año, volumen, editorial, ee, url, nombre_autor from articulo, autor where
articulo.id_articulo=autor.articulo_id_articulo and nombre_autor like
's%';
```

Consulta 6. Consulta con LIKE para mejorar la búsqueda de información de tipo String

En la **Figura 14** y **Figura 15** se muestra los datos de los autores y de las publicaciones científicas que contiene el patrón especificado en la **Consulta 6**

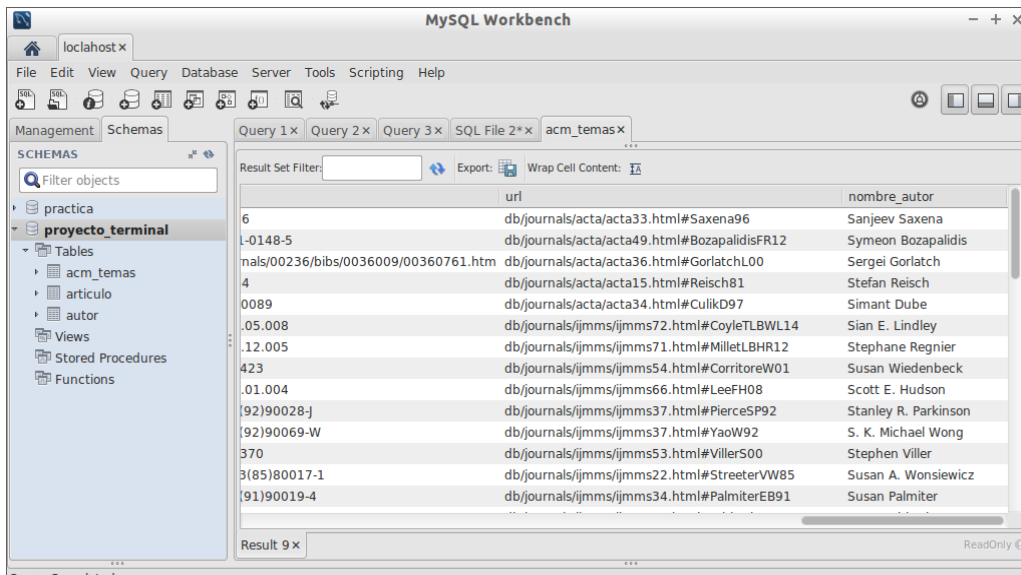


Figura 14. Consulta 6 efectuada en Msql Worbench mostrando los datos de los autores

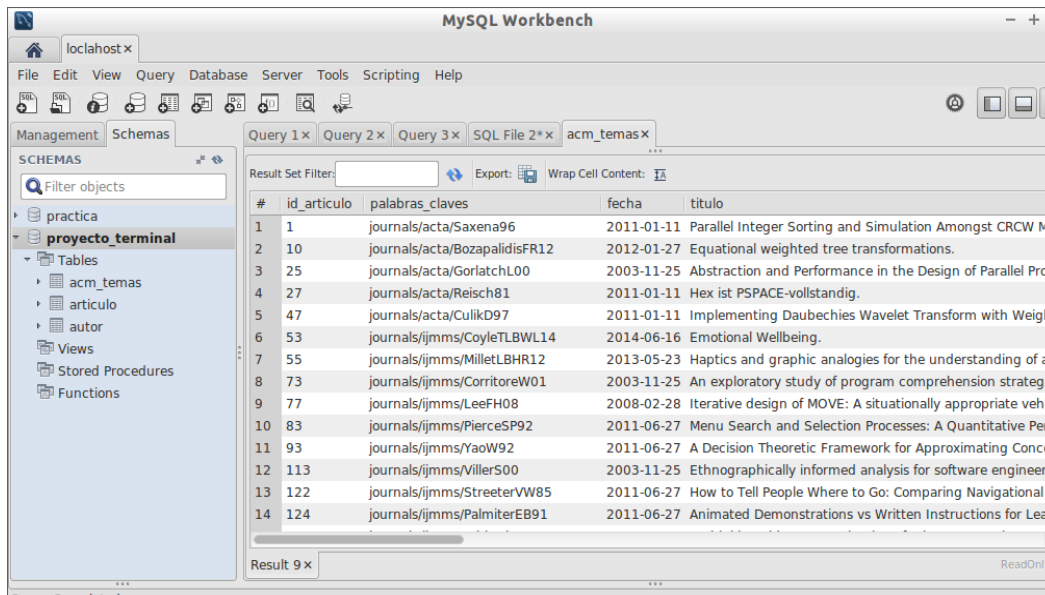


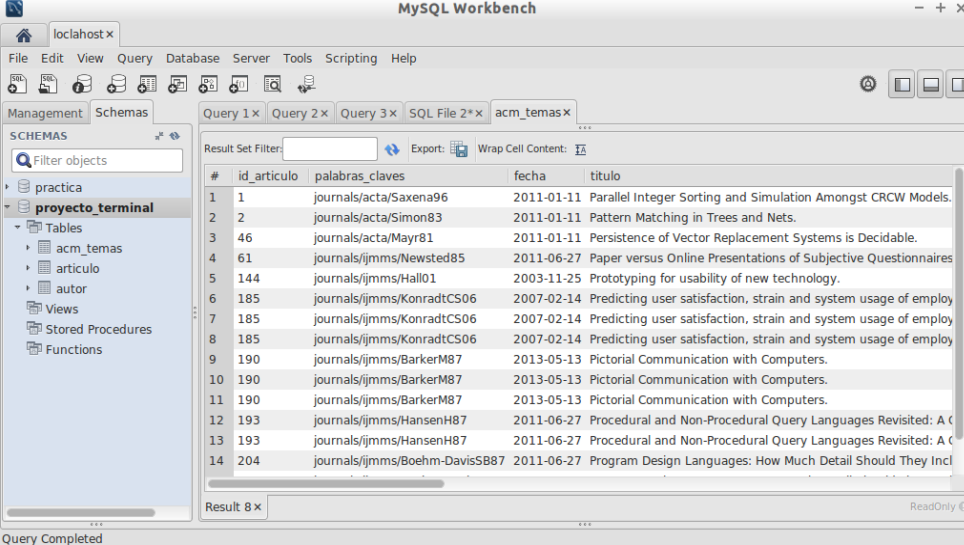
Figura 15. Consulta 6 efectuada en Mysql Worbench con la información de las publicaciones científicas

Se efectuó la **Consulta 7** como otro ejemplo con el título de las publicaciones (articulo).

```
Select id_articulo, palabras_claves, fecha, titulo, pagina,
año,volumen,editorial,ee,url, nombre_autor from articulo,autor where
articulo.id_articulo=autor.articulo_id_articulo and titulo like 'p%';
```

consulta 7. Consulta con Like con el título de las publicaciones científicas.

En la **Figura 16** se muestra se muestra los datos de los autores y de las publicaciones científicas que contiene el patrón especificado en la **Consulta 7**



#	id_articulo	palabras_claves	fecha	titulo
1	1	journals/acta/Saxena96	2011-01-11	Parallel Integer Sorting and Simulation Amongst CRCW Models.
2	2	journals/acta/Simon83	2011-01-11	Pattern Matching in Trees and Nets.
3	46	journals/acta/Mayr81	2011-01-11	Persistence of Vector Replacement Systems is Decidable.
4	61	journals/ijmms/Newsted85	2011-06-27	Paper versus Online Presentations of Subjective Questionnaires
5	144	journals/ijmms/Hall01	2003-11-25	Prototyping for usability of new technology.
6	185	journals/ijmms/KonradtCS06	2007-02-14	Predicting user satisfaction, strain and system usage of employ
7	185	journals/ijmms/KonradtCS06	2007-02-14	Predicting user satisfaction, strain and system usage of employ
8	185	journals/ijmms/KonradtCS06	2007-02-14	Predicting user satisfaction, strain and system usage of employ
9	190	journals/ijmms/BarkerM87	2013-05-13	Pictorial Communication with Computers.
10	190	journals/ijmms/BarkerM87	2013-05-13	Pictorial Communication with Computers.
11	190	journals/ijmms/BarkerM87	2013-05-13	Pictorial Communication with Computers.
12	193	journals/ijmms/HansenH87	2011-06-27	Procedural and Non-Procedural Query Languages Revisited: A C
13	193	journals/ijmms/HansenH87	2011-06-27	Procedural and Non-Procedural Query Languages Revisited: A C
14	204	journals/ijmms/Boehm-DavisSB87	2011-06-27	Program Design Languages: How Much Detail Should They Incl

Figura 16. Consulta 7 efectuada en Mysql Worbench

Con estas comprobaciones se puede verificar que los datos son insertados correctamente en las tablas de las bases de datos para que se puedan manipular los datos con éxitos para los siguientes módulos del proyecto.

6.3 Módulo de Minería de Datos

En este módulo se desarrolla la clasificación de los datos completos agrupados en la base de datos. En este módulo se especificó la clasificación por medio de árboles de decisiones para mejor desempeño del proyecto se utilizó la clasificación por medio del ACM (Association for Computing Machinery) Computing Classification Systems que es un sistema de diseño para la computación para etiquetar y clasificar las publicaciones subsecciones correspondiente a la informática utilizando 12 temas de la taxonomía de clasificación. Para desempeñar esta labor se tuvo que insertar semi automáticamente como se muestra en la **Consulta 8** los temas de ACM Computing Classification Systems en la tabla acm_temas que se diseñó en la base de datos.

```
insert into acm_tema values(1,'Hardware');
insert into acm_tema values(2,'Computer Systems Organization ');
insert into acm_tema values(3,'Networks');
insert into acm_tema values(4,'Software');
insert into acm_tema values(5,'Theory Of Computation');
insert into acm_tema values(6,'Mathematics Of Computing');
insert into acm_tema values(7,'Information Systems');
insert into acm_tema values(8,'Security And Privacy');
insert into acm_tema values(9,'Human Centered Computing');
insert into acm_tema values(10,'Computing Methodologies ');
insert into acm_tema values(11,'Applied Computing');
insert into acm_tema values(12,'Social And Professional Topics');
```

Consulta 8. Inserción de los temas a la tabla acm temas.

En la **Figura 17** se muestra la inserción correcta de los temas en la tabla acm_temas para clasificar las publicaciones científicas.

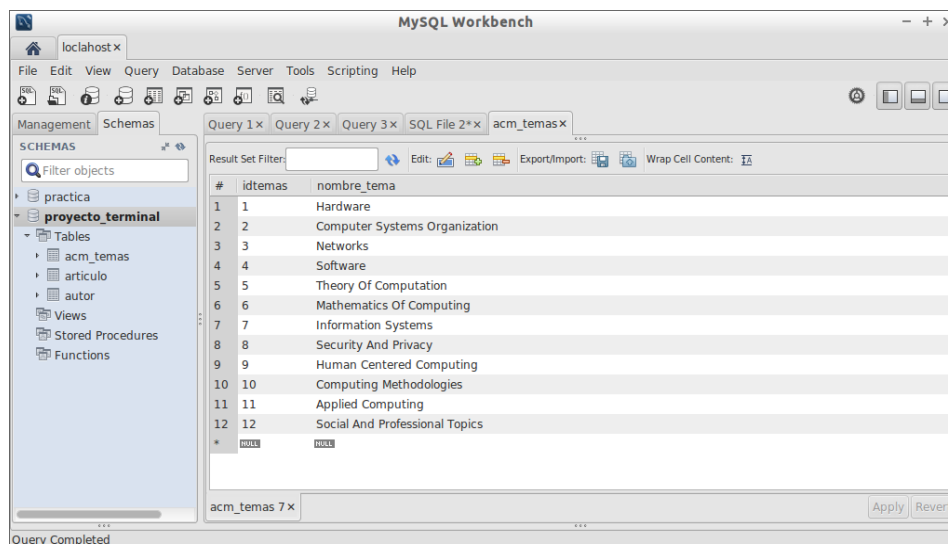


Figura 17. Tabla acm temas insertado correctamente

En la tabla de artículo en donde se insertaron los datos de las publicaciones científicas se insertó una nueva columna con nombre de id tema que se usara como referencia de llave foránea hacia la tabla de artículos para poder clasificar por tema según el número id de la tabla acm tema.

Después de crear la columna se llena semi automáticamente por medio de una consulta UPDATE o manualmente por medio de la tabla de la base de datos para poder referencias los temas con las publicaciones científicas (artículos).

En el **Apéndice 9** se muestra las consultas UPDATE que se efectuaron para la inserción de los números de id relacionando cada publicación científica (artículo) con su tema según la ACM CCS.

En la **Figura 18** se muestra la relación entre los numero de id de las publicaciones científicas con su respectivo tema asignado.

	url	idtema
L007/BF03036466	db/journals/acta/acta33.html#Saxena96	5
L007/BF01257084	db/journals/acta/acta20.html#Simon83	2
L007/BF00289414	db/journals/acta/acta20.html#Goodman583	3
L007/BF00264161	db/journals/acta/acta17.html#Blum82	5
L007/BF00289470	db/journals/acta/acta7.html#Schonhage77	5
L007/s00236-010-0128-1	db/journals/acta/acta48.html#Honkala11	6
L007/BF00282618	db/journals/acta/acta24.html#HuangL87	6
L007/BF00268843	db/journals/acta/acta25.html#FinkelC87	3
L007/BF00264322	db/journals/acta/acta10.html#Biskup78	7
L007/s00236-011-0148-5	db/journals/acta/acta49.html#BozapalidisFR12	6
L007/s00236-006-0023-y	db/journals/acta/acta43.html#KhomenkoKKV06	7
L007/s00236-013-0178-2	db/journals/acta/acta50.html#Hesselink13	4
L007/BF00289416	db/journals/acta/acta20.html#Ronse83	3
L007/BF01178582	db/journals/acta/acta28.html#CritchlowP91	8

Figura 18. Tabla articulo con las modificaciones mencionadas anteriormente.

La clasificación se desarrolló de la siguiente manera:

1. De los temas insertados en la tabla acm temas se les asigno un numero de id del 1 al 12 (ver. **Figura 17**)
2. Se investiga en los temas del ACM Computing Classification Systems (ACM CCS) y se busca palabras claves en los titulo de la publicación científica como se muestra en la **Figura 19** para poder relacionar cada publicación científica con su respectivo tema

Pattern Matching in Trees and **Nets.**

Figura 19. Título de ejemplo para la clasificación

En el título de la publicación científica (**Figura 19**) se buscó la siguiente palabra clave Nets que se significa redes en uno de los temas del ACM CCS es Network donde hace referencia a todo relacionado con redes de computadora por lo tanto en la **Figura 20** se muestra como la columna id tema de la tabla artículo se coloca el numero tres porque el tema Network su número de id (idtemas) asignado en la tabla acm_tema es el numero tres para poder hacer la relación

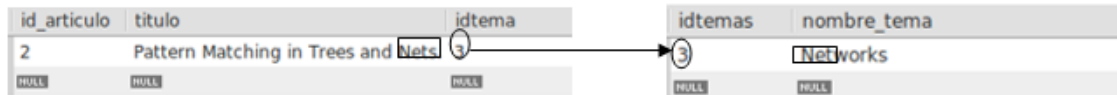


Figura 20. Ejemplo de cómo clasificar.

3. Se realiza la clasificación en las 300 publicaciones científicas del archivos.xml con el procedimiento del paso 2 para que posteriormente se evalúen por medio de consultas implementadas en JAVA.

6.4 Módulo de Evaluación de Patrones

En este módulo se evalúan los patrones con los datos obtenidos por los módulos anteriores implementar en el lenguaje de JAVA técnicas para clasificar.

En este módulo se evaluaron los patrones por medio de consultas diseñadas e implementadas en el lenguaje JAVA para mostrar mejor los datos organizados y clasificados en el módulo de presentación.

De todos los atributos especificado en el articulos.xml (palabra claves (key), fecha (mdate), autor (autor), title (titulo), pages (pagina), year (año), volume (volumen), journal (editorial), number (numero), url (url), ee (ee)) que es una parte obtenida del archivo dblp.xml.

Se escogió realizar las consultas con los atributos de fecha, autor, titulo, volumen, número y por el número de id (el número de id no se especifica en el archivo xml si no cuando se inserta los datos en la tabla articulo (publicaciones científicas)).

Se escogieron estos atributos ya que los contiene en diferentes publicaciones científicas y se relaciona entre si, así se puede obtener información relevante.

Por ultimo también se evaluó por medio de consultas la clasificación por medio del ACM CCS.

Para realizar las consultas también se implementaron sus correspondientes VO, autorVO **Apéndice 10**, articuloVO **Apéndice 11**, acmtemasVO **Apéndice 12** para poder relacionar las columnas de las tablas artículo, autor y acmtemas con las vistas que se diseñaron en el módulo de presentación.

La primera consulta implementada en este caso fue la del número de id, posteriormente se implementó la consulta por medio de fecha, por título, por volumen y por número **Apéndice 13** así como por medio de la clasificación **Apéndice 15** y por autor **Apéndice 14**

Se implementaron como consultas anidadas, es decir, si buscamos el nombre del autor se realizó la consulta en la tabla autor, posteriormente se relacionó el número del id del autor con la consulta de la tabla articulo y acmtemas para poder obtener el diseño deseado.

En la **Consulta 9** se muestra un ejemplo de las consultas anidadas implementadas en el lenguaje JAVA

```
SELECT nombre_autor, articulo_id_articulo FROM autor WHERE nombre_autor
LIKE '%" + nombre_autor + "%';
```

```
SELECT nombre_tema, palabras_claves, fecha, titulo, " +
"pagina, año, volumen, editorial, numero, ee, url FROM
articulo, acm_temas WHERE articulo.idtema=acm_temas.idtemas AND
id_articulo="+rs.getInt("articulo_id_articulo")+";
```

Consulta 9. Diseño de la consulta implementado en el lenguaje JAVA

El procedimiento para la evaluación de patrones es el siguiente:

1. Se insertar un patrón a evaluar en la vista de consulta (por número de id, por autor, por fecha, por título, por volumen, por número y por nombre de tema)
2. Las peticiones de patrones a evaluar son dirigidas hacia un Servlet de consulta en donde hace la petición hacia la consulta implementada en el lenguaje JAVA.
3. Las evaluaciones de las consultas implementadas en el lenguaje JAVA son anidadas.
4. Después de la evaluación de las consultas se redirige al Servlet para que muestre los datos por medio de las vistas diseñadas (mostrar consulta).
5. Se muestra los datos de las consultas por medio de tablas ordenadas y organizadas

En la **Figura 21** se muestra gráficamente el procedimiento que se desarrolló para la evaluación de patrones en el proyecto terminal

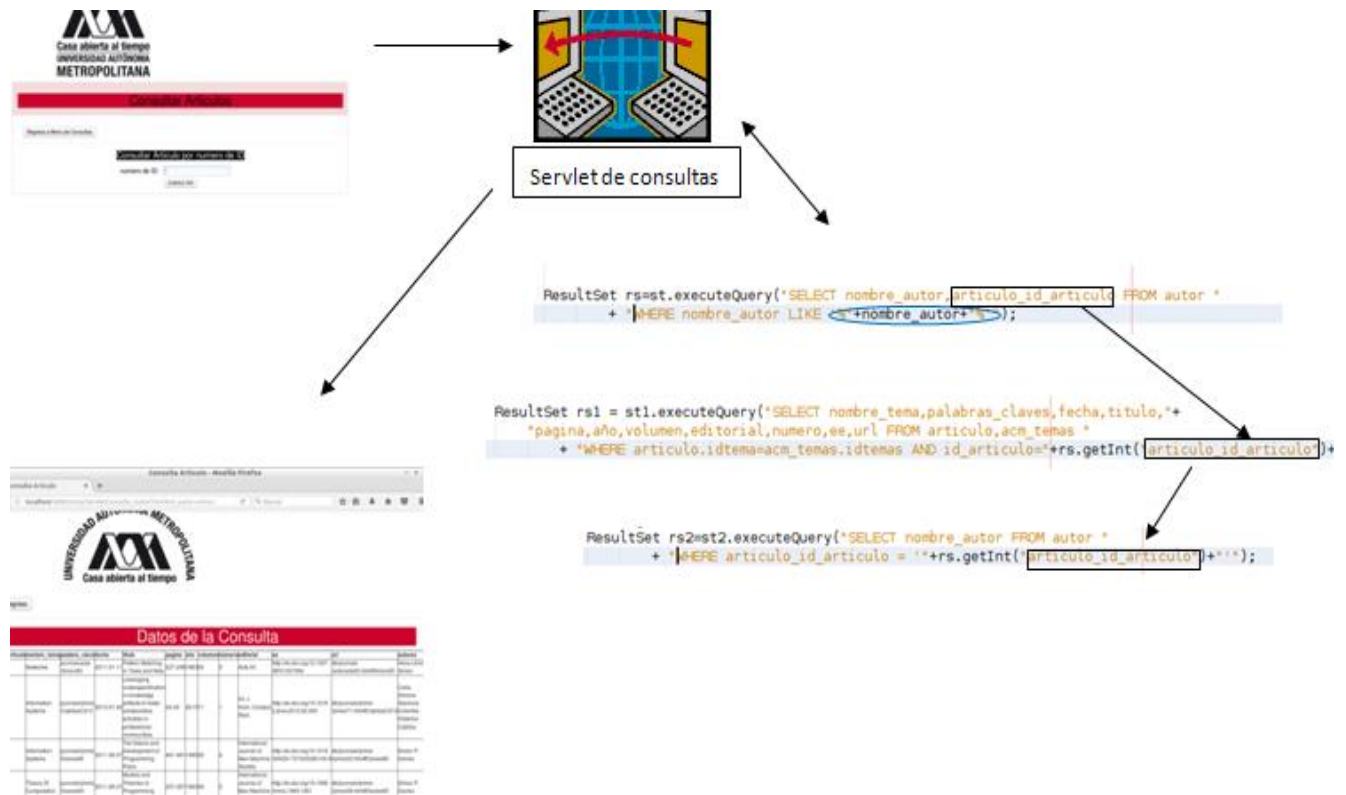


Figura 21. Procedimiento para la evaluación de patrones

Las consultas realizadas por autor, titulo, nombre del tema y fecha son evaluadas con la sentencia LIKE en donde se evalúa un patrón específico (por medio de una letra de alfabeto, una palabra específica o un número). Por lo tanto, en las consultas de número, volumen y número de id son evaluadas por medio de una consulta sencilla de enteros.

6.5 Módulo de Presentación

En este módulo se desarrolló tablas visuales en JSP con los datos evaluados en el módulo de evaluación de patrones para la visualización del usuario.

En este módulo se efectuaron en 5 pasos.

Paso 1: Se implementó una vista principal del proyecto terminal de investigación con el código **Apéndice 16**

En la **Figura 22** se muestra el diseño que contiene dos botones para la consulta del artículo y otro para la consulta de la clasificación.



Figura 22. Vista Principal del Proyecto Terminal

Paso 2: si se seleccionó el consultar artículo se diseñó un menú de consultas **Apéndice 17** en donde el usuario puede interactuar con diferentes botones para realizar la consulta deseada, si se seleccionó el consultar clasificación se diseñó una vista para insertar el patrón a buscar del nombre del tema **Apéndice 18**

En la **Figura 23** se muestra la vista con el diseño del menú de consultas mientras en la **Figura 24** se muestra la vista con el diseño de inicio para poder consultar por medio del tema de clasificación

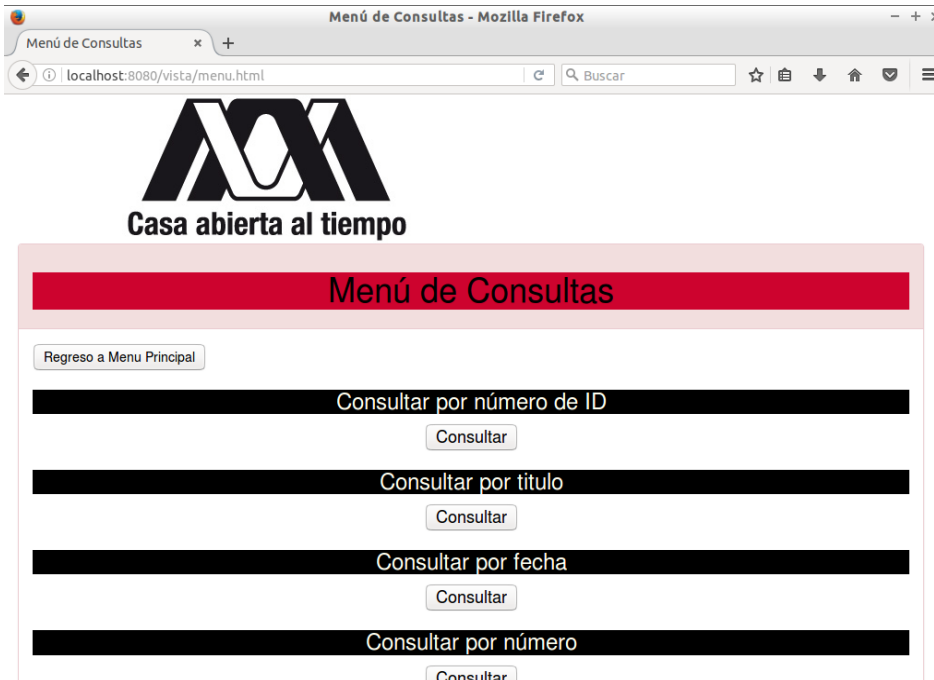


Figura 23. Menú de Consultas del Proyecto Terminal



Figura 24. Vista de Consulta para la Clasificación.

Paso 3: Se diseñaron varias vistas para cada uno de las selecciones del menú de consultas por medio de numero de id del articulo (publicación científica), por medio de título, por medio de volumen, por medio de fecha , por medio de numero y por medio de autor **Apéndice 19.**

En la **Figura 25** se muestra la vista de diseño para que el usuario insertara el patrón a consultar (fecha, título, autor, volumen, numero o número de id del articulo (publicación científica))



Figura 25. Vista de consulta para los Publicaciones Científicas

Paso 4: Se diseñaron Servlet por número de id de artículo (publicación científica), por título, por fecha, por autor, por numero y por volumen, así como el Servlet para la consulta de la clasificación, para poder unir las consultas implementadas en JAVA con las vistas JSP **Apéndice 20**.

Paso 5: Se diseñó vistas JSP para poder mostrar las consultas por medio de tablas con la información correspondiente, por medio de los atributos fecha, por título, por autor, por volumen, por número, por número de id de articulo (publicación científica) y por ultimo para la clasificación **Apéndice 21**.

En la **Figura 26** se muestra la vista diseñada para mostrar los datos de las consultas por medio de las tablas



Figura 26. Vista para mostrar los datos de las consultas

7 Resultados

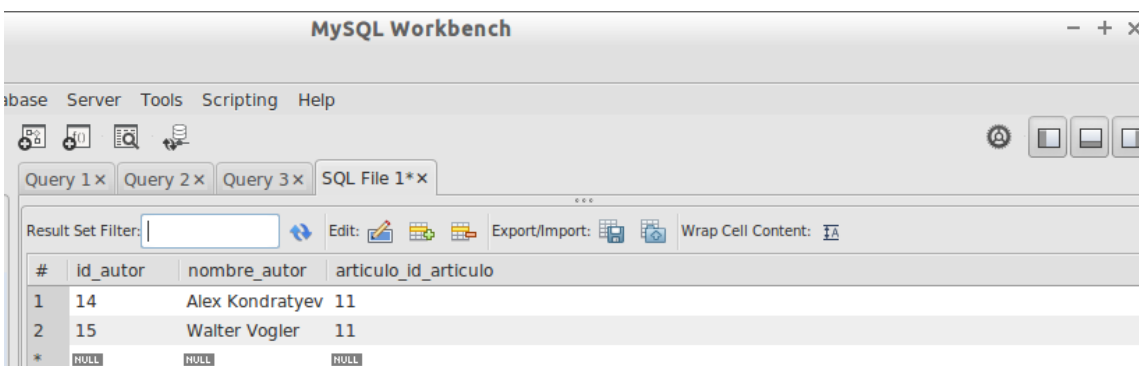
7.1 Pruebas para la base de datos

Prueba 1: Se realizó con la siguiente publicación científica en donde en la **Figura 27** se muestra el código XML en donde se quitaron dos autores Víctor Khomenko y Walter Vogler para realizar la prueba

```
<article mdate="2006-11-22" key="journals/acta/KhomenkoKKV06" >
  <author>Victor Khomenko</author>
  <author>Alex Kondratyev</author>
  <author>Maciej Koutny</author>
  <author>Walter Vogler</author>
  <title>Merged processes: a new condensed representation of Petri net behaviour.</t
  <pages>307- 330</pages>
  <year>2006</year>
  <volume>43</volume>
  <journal>Acta Inf.</journal>
  <number>5</number>
  <ee>http://dx.doi.org/10.1007/s00236-006-0023-y</ee>
  <url>db/journals/acta/acta43.html#KhomenkoKKV06</url>
</article>
```

Figura 27. Publicación Científica donde se modificaron los autores

En la **Figura 28** se muestra el resultado de la inserción de la **prueba 1** con la **Consulta 10** efectuada.



#	id_autor	nombre_autor	articulo_id_articulo
1	14	Alex Kondratyev	11
2	15	Walter Vogler	11
*	NULL	NULL	NULL

Figura 28. Resultado de las publicaciones científicas modificadas por autor

```
SELECT * FROM autor where articulo_id_articulo = 11
```

Consulta 10. Consulta realizada para mostrar el resultado

En esta prueba se efectuó para verificar el diseño del parser Dom para la inserción de los datos en las tablas de la base de datos, cuando el algoritmo no encuentra un autor en el código de XML no lo inserta en la tabla autor ya que en el algoritmo tiene una condición en donde se especifica que si encuentre el nodo `<author></author>` se inserta la información mientras no se encuentra ese nodo no lo inserta hasta que encuentra el nodo es donde insertar la información

Prueba 2: Se realizó con la siguiente publicación científica en donde en la **Figura 29** se muestra el código XML en donde no contiene el nodo <ee></ee>

```

<article mdate="2003-11-25" key="journals/ijmms/Black92">
  <author>Alison Black</author>
  <title>Book Review: "Envisioning Information, " by Edward Tufte.</title>
  (Reseña del libro: Información de visualización)
  <pages>387-388</pages>
  <year>1992</year>
  <volume>37</volume>
  <journal>International Journal of Man-Machine Studies</journal>
  <number>3</number>
  <url>db/journals/ijmms/ijmms37.html#Black92</url>
</article>

```

Figura 29. Publicación Científica donde no contiene el nodo cadena para la prueba

En la **Figura 30** se muestra el resultado de la inserción de la **prueba 2** con la **Consulta 11** efectuada.

pagina	año	volumen	editorial	numero	ee	url
e. 387-388	1992	37	International Journal of Man-Machine Studies	3	NULL	db/journals
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 30. Resultado de la publicación científica no contiene el nodo ee

```
SELECT * FROM articulo where id_articulo = 67
```

Consulta 11. Consulta realizada para mostrar el resultado

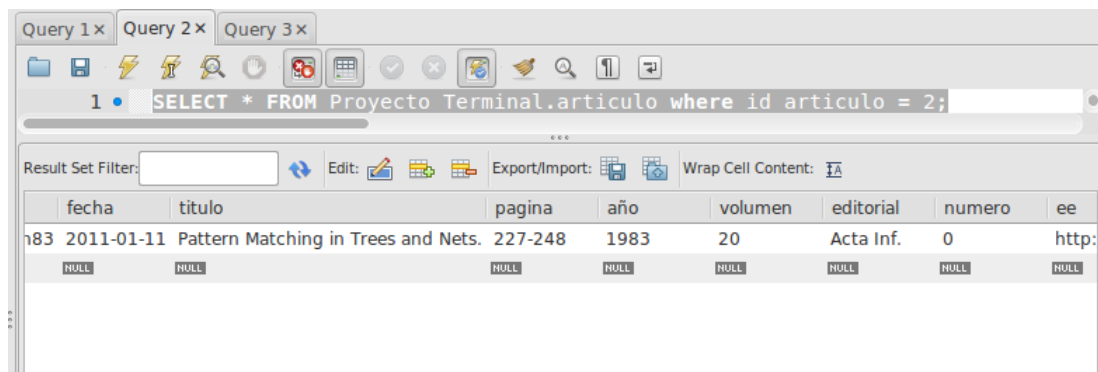
En esta prueba se realizó para verificar cuando la publicación científica no contenía un nodo específico por el diseño del XML en este caso fue el nodo <ee></ee>. En el algoritmo de parser Dom se diseñó para al momento de no encontrar un nodo de la publicación científica de XML de tipo String o cadena que se especificó en el algoritmo que son <ee></ee>, <title></title>, <pages></pages>, <journal></journal>, <url></url> así como el mdate y key se coloca un null para representar que la cadena está vacía

Prueba 3: Se realizó con la siguiente publicación científica en donde en la **Figura 31** se muestra el código XML en donde no contiene el nodo <number></number>

```
<article mdate="2011-01-11" key="journals/acta/Simon83">  
<author>Hans-Ulrich Simon</author>  
<title>Pattern Matching in Trees and Nets.</title> art_tema 2  
<pages>227-248</pages>  
<year>1983</year>  
<volume>20</volume>  
<journal>Acta Inf.</journal>  
<url>db/journals/acta/acta20.html#Simon83</url>  
<ee>http://dx.doi.org/10.1007/BF01257084</ee>  
</article>
```

Figura 31. Publicación Científica donde no contiene el nodo numero para la prueba

En la **Figura 32** se muestra el resultado de la inserción de la **prueba 3** con la **Consulta 12** efectuada



fecha	titulo	pagina	año	volumen	editorial	numero	ee
2011-01-11	Pattern Matching in Trees and Nets.	227-248	1983	20	Acta Inf.	0	http:
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 32. Resultado de la publicación científica no contiene el nodo número

```
SELECT * FROM articulo where id_articulo = 2
```

Consulta 12. Consulta realizada para mostrar el resultado

En esta prueba se realizó para verificar cuando una publicación científica no contiene un nodo específico por el diseño del XML en este caso fue el nodo <number></number> En el algoritmo de parser Dom se diseñó para al momento de no encontrar un nodo de la publicación científica de XML de tipo int o entero que se especificó en el algoritmo que son <volume></volume>, <number></number>, <year></year> se coloca un cero para representar que el entero está vacío

7.2 Pruebas para el Módulo de Integración

Prueba 1: Se realizó con la siguiente publicación científica en donde en la **Figura 33** se muestra el código XML donde se observa un carácter especial o con signos de puntuación en el nodo de `<author></author>` en donde solo se hizo la prueba con el nombre del autor

```
<article mdate="2013-11-28" key="journals/acta/Schonhage77">
<author>Arnold Sch&ouml;nhage</author>
<title>Schnelle Multiplikation von Polynomen &uuml;ber K&ouml;rpern der Charakteristik 2.</title>
<journal>Acta Inf.</journal>
<volume>7</volume>
<year>1977</year>
<pages>395-398</pages>
<url>db/journals/acta/acta7.html#Schonhage77</url>
<ee>http://dx.doi.org/10.1007/BF00289470</ee>
</article>
```

Figura 33. Publicación científica en donde se muestra un carácter especial en el nodo autor

En la **Figura 34**, **Figura 35**, se muestra las consultas y resultados efectuadas en la **prueba 1**



Figura 34. Realización de la consulta de prueba



Datos de la Consulta												
id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	lee	url	autores
5	Theory Of Computation	journals/acta/Schonhage77	2013-11-28	Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2.	395-398	1977	7	0	Acta Inf.	10.1007/BF00289470	http://dx.doi.org/db/journals/acta/acta7.html#Schonhage77	Arnold Schönhage

Figura 35. Visualización de la consulta

Se realizó la prueba para verificar la funcionalidad de la consulta implementada en JAVA por medio de una evaluación de patrón en la vista JSP del proyecto terminal en donde se evaluó solo el nombre del autor Arnold en donde al momento de evaluarlo se muestra el resultado información en la tabla de los datos de la consulta ya que la consulta implementada reconoce el patrón (Figura35)

Prueba 2: Se realizó la misma publicación científica de la Figura 33 en donde se realizó la prueba con el nombre completo del autor

En la Figura 36, Figura 37, se muestra las consultas y resultados efectuadas en la prueba 2



Figura 36. Realización de la consulta con cadenas con signos de puntuación



Figura 37. Visualización de la consulta sin encontrar resultado.

Se realizó la prueba para verificar la funcionalidad de la consulta implementada en JAVA por medio de una evaluación de patrón en la vista JSP del proyecto terminal en donde se evaluó en nombre completo de Arnold Schönbage en donde al momento de evaluarlo no se mostró ningún resultado información en la tabla de los datos de la consulta ya que la consulta implementada no reconoce el patrón con caracteres especiales o signo de puntuación (**Figura37**)

En la **Figura 40** se muestra una verificación para mostrar si la información de la consulta empleada en la vista JSP es correcta

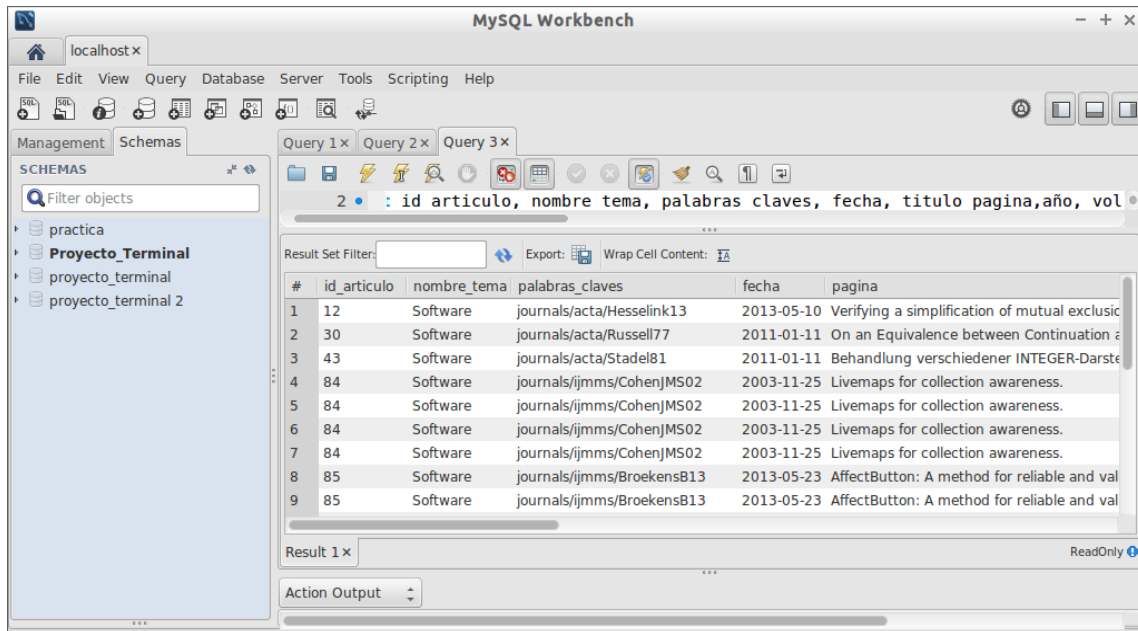


Figura 40. Verificación por medio de Mysql Worbench de la consulta efectuada anteriormente

En esta prueba se efectuó para verificar si la clasificación fue elaborada correctamente si se evalúa en un patrón en este caso Software y que contenga el tema que se está consultando.

Prueba 2: En esta prueba se evaluaron patrón Security and Privacy para verificar que si se realizó la clasificación correctamente

En la **Figura 41** y la **Figura 42** se realiza la evaluación del patrón y el resultado en la vista datos de la consulta

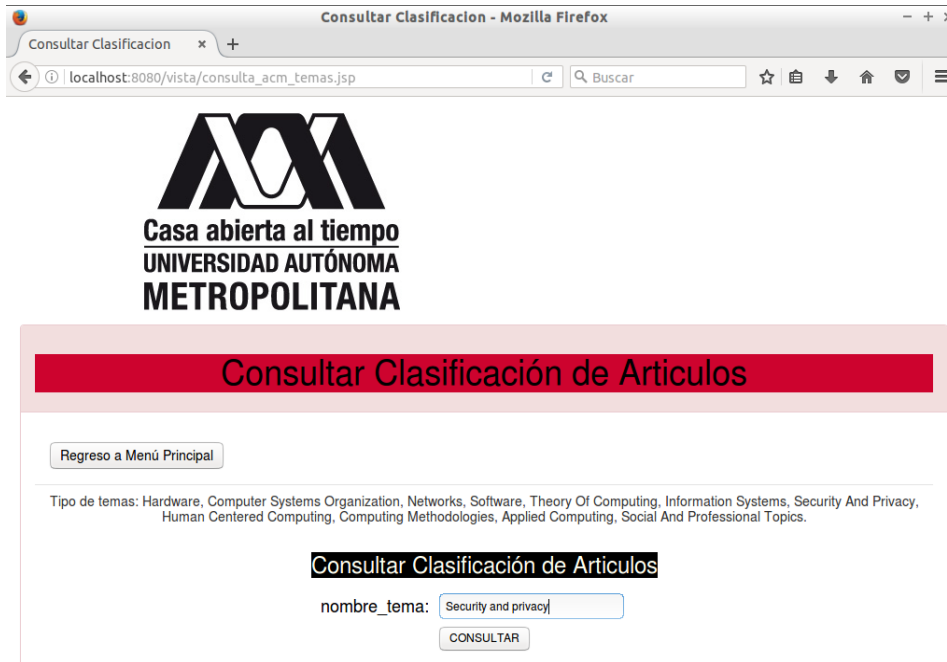


Figura 41. Segunda Consulta de prueba para la clasificación.

Datos de la Consulta											
id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	ee	url
14	Security And Privacy	journals/acta/CritchlowP91	2011-01-11	The Expressive Power of Delay Operators in SCCS.	447-452	1991	28	5	Acta Inf.	http://dx.doi.org/10.1007/BF01178582	db/journals/acta/acta28.html#CritchowP91
111	Security And Privacy	journals/ijmms/GomezW91	2011-06-27	TO: A Language for Specifying Programming Problems.	633-656	1991	35	5	International Journal of Man-Machine Studies	http://dx.doi.org/10.1016/S0020-7373(05)80181-6	db/journals/ijmms/ijmms35.html#GomezW91
112	Security And Privacy	journals/ijmms/FlechaisS09	2009-09-10	Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-Science.	281-296	2009	67	4	Int. J. Hum.-Comput Stud.	http://dx.doi.org/10.1016/j.jhcs.2007.10.002	db/journals/ijmms/ijmms67.html#FlechaisS09
116	Security And Privacy	journals/ijmms/BahnerHM08	2009-02-12	Misuse of automated decision aids: Complacency, automation bias and the impact of training experience.	688-699	2008	66	9	Int. J. Hum.-Comput Stud.	http://dx.doi.org/10.1016/j.jhcs.2008.06.001	db/journals/ijmms/ijmms66.html#BahnerHM08

Figura 42. Visualización de resultados de la segunda consulta

En la **Figura 43** se muestra una verificación para mostrar si la información de la consulta empleada en la vista JSP es correcta

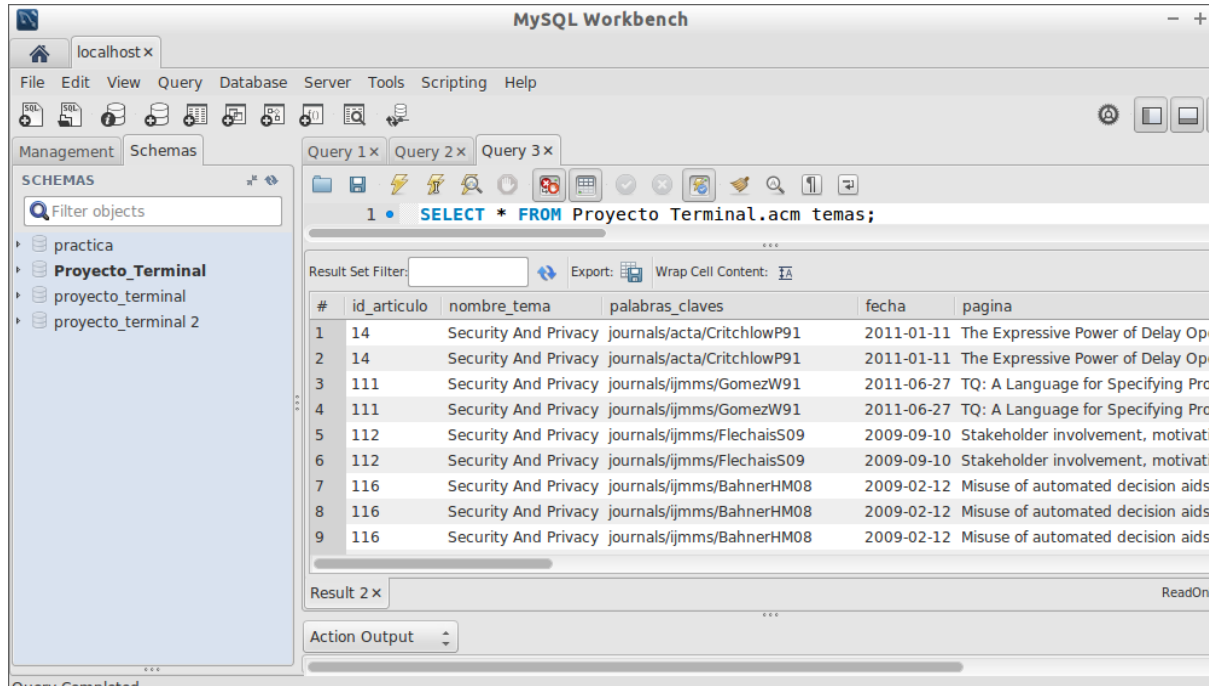


Figura 43. Verificación por medio de Mysql Worbench de la consulta efectuada anteriormente

En esta prueba se efectuó para verificar si la clasificación fue elaborada correctamente si se evalúa en un patrón en este caso Software y que contenga el tema que se está consultando.

7.4 Pruebas para el Módulo de Evaluación de Patrones

Prueba 1: Se realiza la prueba con la evaluación del patrón Pattern que es una parte de un titulo

En la **Figura 44** y la **Figura 45** se realiza la evaluación del patrón y el resultado en la vista datos de la consulta



Figura 44. Primera Consulta para la prueba de evaluación de patrones por titulo

id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	doi	url	autores
2	Networks	journals/acta Simon83	2011-01-11	Pattern Matching in Trees and Nets.	227-248	1983	20	0	Acta Inf.	http://dx.doi.org/10.1007/BF01257084	dbjournals/acta/acta20.html#Simon83	Hans-Ulrich Simon
231	Information Systems	journals/ijmms SarkelaTMLN09	2011-06-16	The movement patterns and the experiential components of virtual environments.	787-798	2009	67	9	Int. J. Hum.-Comput. Stud.	http://dx.doi.org/10.1016/j.ijhcs.2009.05.003	dbjournals/ijmms/ijmms67.html#SarkelaTMLN09	Gole Nyman Mikko Laakso Patrick J. C. May Jari Takala Heikki Sarkela

Figura 45. Visualización de resultados de la primera consulta

En esta prueba se realizó para verificar la consulta implementada en JAVA por medio de la vista JSP, se puede visualizar que la consulta fue efectuada correctamente por medio de la sentencia Like que evalúa la palabra, letra o patrón específico dando el resultado cuando encuentra ese patrón que evaluó en este caso fue Pattern, obteniendo dos resultados correctos **Pattern Matching in Tree and Nets** y **The movement patterns and the experiential componenets of virtual environmenets** mostrado en la **Figura 45**

Prueba 2: Se realiza la prueba con la evaluación del patrón Simón que es un apellido de un autor

En la **Figura 46** y la **Figura 47** se realiza la evaluación del patrón y el resultado en la vista datos de la consulta



Figura 46. Segunda Consulta para la prueba de evaluación de patrones por nombre de autor

Datos de la Consulta

id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	ee	url	autores
2	Networks	journals/acta/Simon83	2011-01-11	Pattern Matching in Trees and Nets.	227-248	1983	20	0	Acta Inf.	http://dx.doi.org/10.1007/BF01257084	db/journals/acta/acta20.html#Simon83	Hans-Ulrich Simon
133	Information Systems	journals/ijmms/CabitzaCS13	2013-01-04	Leveraging underspecification in knowledge artifacts to foster collaborative activities in professional communities.	24-45	2013	71	1	Int. J. Hum.-Comput. Stud.	http://dx.doi.org/10.1016/j.ijhcs.2012.02.005	db/journals/ijmms/ijmms71.html#CabitzaCS13	Carla Simone Gianluca Colombo Federico Cabitza
143	Information Systems	journals/ijmms/Davies90	2011-06-27	The Nature and Development of Programming Plans.	461-481	1990	32	4	International Journal of Man-Machine Studies	http://dx.doi.org/10.1016/S0020-7373(05)80143-9	db/journals/ijmms/ijmms32.html#Davies90	Simon P. Davies
284	Theory Of Computation	journals/ijmms/Davies93	2011-06-27	Models and Theories of Programming Strategy.	237-267	1993	39	2	International Journal of Man-Machine Studies	http://dx.doi.org/10.1006/ijmms.1993.1061	db/journals/ijmms/ijmms39.html#Davies93	Simon P. Davies

Figura 47. Visualización de resultados de la segunda consulta

En esta prueba se realizó para verificar la consulta implementada en JAVA por medio de la vista JSP, se puede visualizar que la consulta fue efectuada correctamente por medio de la sentencia Like que evalúa la palabra, letra o patrón específico dando el resultado cuando encuentra ese patrón que evaluó en este caso **Simón** obteniendo los siguientes resultados correctos **Hans- Ulrich Simon, Carla Simone, Simon P. Davies** mostrado en la **Figura 47**

Prueba 3: Se realiza la prueba con la evaluación del patrón 23 que es número de id de una publicación científica

En la **Figura 48** y la **Figura 49** se realiza la evaluación del patrón y el resultado en la vista datos de la consulta



Figura 48. Segunda Consulta para la prueba de evaluación de patrones por número de ID



Figura 49. Visualización de resultados de la tercera consulta

En esta prueba se realizó para verificar la consulta implementada en JAVA por medio de la vista JSP, se puede visualizar que la consulta fue efectuada correctamente, se evaluó por medio de una consulta sin sentencia Like evaluando solo un entero en este caso un numero de id cual fue 23 obteniendo los resultados correctos mostrado en la **Figura 49**

7.5 Pruebas para el Módulo de Presentación

Prueba 1: Se realizó una consulta para por un patrón común solo una letra del alfabeto para que por medio de la consulta con la sentencia LIKE se pueda obtener la mayor información posible.

En la **Figura 50** y la **Figura 51** se realiza la evaluación del patrón y el resultado en la vista datos de la consulta



Figura 50. Consulta realizada para la prueba del módulo de presentación

Consulta Artículo - Mozilla Firefox

Consulta Artículo

localhost:8080/vista/ServletConsulta_ACM_Temas?nombre_tema=c

Buscar

Datos de la Consulta

Id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	ee	url
1	Theory Of Computation	journals/acta/Saxena96	2011-01-11	Parallel Integer Sorting and Simulation Amongst CRCW Models.	607-619	1996	33	7	Acta Inf.	http://dx.doi.org/10.1007/BF03036466	db/journals/acta/acta33.html#
3	Computer Systems Organization	journals/acta/GoodmanS83	2011-01-11	NP-complete Problems Simplified on Tree Schemas.	171-178	1983	20	0	Acta Inf.	http://dx.doi.org/10.1007/BF00289414	db/journals/acta/acta20.html#
4	Theory Of Computation	journals/acta/Blum82	2011-01-11	On the Power of Chain Rules in Context Free Grammars.	425-433	1982	17	0	Acta Inf.	http://dx.doi.org/10.1007/BF00264161	db/journals/acta/acta17.html#
5	Theory Of Computation	journals/acta/Schonhage77	2013-11-28	Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2.	395-398	1977	7	0	Acta Inf.	http://dx.doi.org/10.1007/BF00289470	db/journals/acta/acta7.html#S
6	Mathematics Of Computing	journals/acta/Honkala11	2011-02-16	A characterization of rational DOL power series.	19-24	2011	48	1	Acta Inf.	http://dx.doi.org/10.1007/s00236-010-0128-1	db/journals/acta/acta48.html#
7	Mathematics Of Computing	journals/acta/HuangL87	2011-01-11	The Derivation of Systolic Implementations of Programs.	595-632	1987	24	6	Acta Inf.	http://dx.doi.org/10.1007/BF00282618	db/journals/acta/acta24.html#
10	Mathematics Of Computing	journals/acta/BozapalidisFR12	2012-01-27	Equational weighted tree transformations.	29-52	2012	49	1	Acta Inf.	http://dx.doi.org/10.1007/s00236-011-0148-5	db/journals/acta/acta49.html#
14	Security And Privacy	journals/acta/CritchlowP91	2011-01-11	The Expressive Power of Delay Operators in SCCS.	447-452	1991	28	5	Acta Inf.	http://dx.doi.org/10.1007/BF01178582	db/journals/acta/acta28.html#
15	Theory Of Computation	journals/acta/Milner96	2011-01-11	Calculi for Interaction.	707-737	1996	33	8	Acta Inf.	http://dx.doi.org/10.1007/BF03036472	db/journals/acta/acta33.html#
16	Applied Computing	journals/acta/Darlington78	2011-01-11	A Synthesis of Several Sorting Algorithms.	1-30	1978	11	0	Acta Inf.	http://dx.doi.org/10.1007/BF00264597	db/journals/acta/acta11.html#
	Social And	journals/acta/		A Workload Model						http://dx.doi.org/10.1007/	db/journals/acta/

Figura 51. Visualización del resultado de la consulta

En esta prueba se especificó un patrón por medio una letra de alfabeto en este caso fue una c para poder obtener la mayor información que puede arrojar la consulta con una sentencia LIKE para verificar que la consulta implementada en JAVA evalúa correctamente, en este caso la evaluación y la obtención de la información fue la correcta mostrado en la **Figura 51**

8 Análisis y Discusión de Resultados

8.1 Análisis de resultados de la Base de Datos

Se realizaron dos pruebas en donde se quitaron autores de la publicación científica **Figura 27** se analizó que al momento de insertar los datos de las publicaciones científicas (artículos) en la bases de datos se inserta correctamente, pero al momento de revisar la tabla de los autores **Figura 28** se ve que solo se inserta dos autores de los cuatro que tenía originalmente la publicación científica del archivo artículos.xml, esto se debe a que el código de Parser Dom hay una sentencia en donde si no se encuentra un autor no se inserta, por lo tanto la prueba fue un éxito y la tablas fue llenada correctamente.

En la **Figura 29** se eligió una publicación científica (articulo) donde no tiene un atributo (nodo) ee y una publicación científica en donde no tiene un atributo (nodo) numero **Figura 31** para verificar la inserción, en este caso los atributos que no se encuentra en el contexto de la publicación científica por medio del código de Parser Dom que se diseñó para que colocar un null si no se encuentra una cadena y un cero si no se encuentra un entero para no perder la información al momento de realizar las consultas en las vistas.

En esta prueba se realizó para ver si se insertaron correctamente los nodos cadenas vacías **Figura 30** y nodos enteros vacíos **Figura 32** donde se muestran que la inserción fue efectuada correctamente.

Por último, se realizó una consulta en las vistas JSP del proyecto terminal para comprobar los resultados y verificar el análisis descrito anteriormente

En la **Figura 52** se puede observar el parámetro null del nodo ee fue interpretado como un dato de la publicación científica, así como en la **Figura 53** se muestra como el parámetro cero del nodo numero también se interpretó como un dato de su respectiva publicación científica mostrando correctamente los datos de las consultas.

Consulta Artículo - Mozilla Firefox

localhost:8080/vista/ServletConsulta_id_articulo?id_articulo=67



Regreso


Datos de la Consulta

id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	lee	url	autores
67	Applied Computing	journals/ijmms/Black92	2003-11-25	Book Review: "Envisioning Information," by Edward Tufte.	387-388	1992	37	3	International Journal of Man-Machine Studies	ee	db/journals/ijmms/ijmms37.html#Black92	Alison Black

Figura 52. Consulta de la publicación científica que no contiene la cadena ee

Consulta Artículo - Mozilla Firefox

localhost:8080/vista/ServletConsulta_id_articulo?id_articulo=2



Regreso

Datos de la Consulta

id_articulo	nombre_tema	palabra_clave	fecha	titulo	pagina	año	volumen	número	editorial	lee	url	autores
2	Networks	journals/acta/Simon83	2011-01-11	Pattern Matching in Trees and	227-248	1983	20	0	Acta Inf.	http://dx.doi.org/10.1007/BF01257084	db/journals/acta/acta20.html#Simon83	Hans-Ulrich Simon

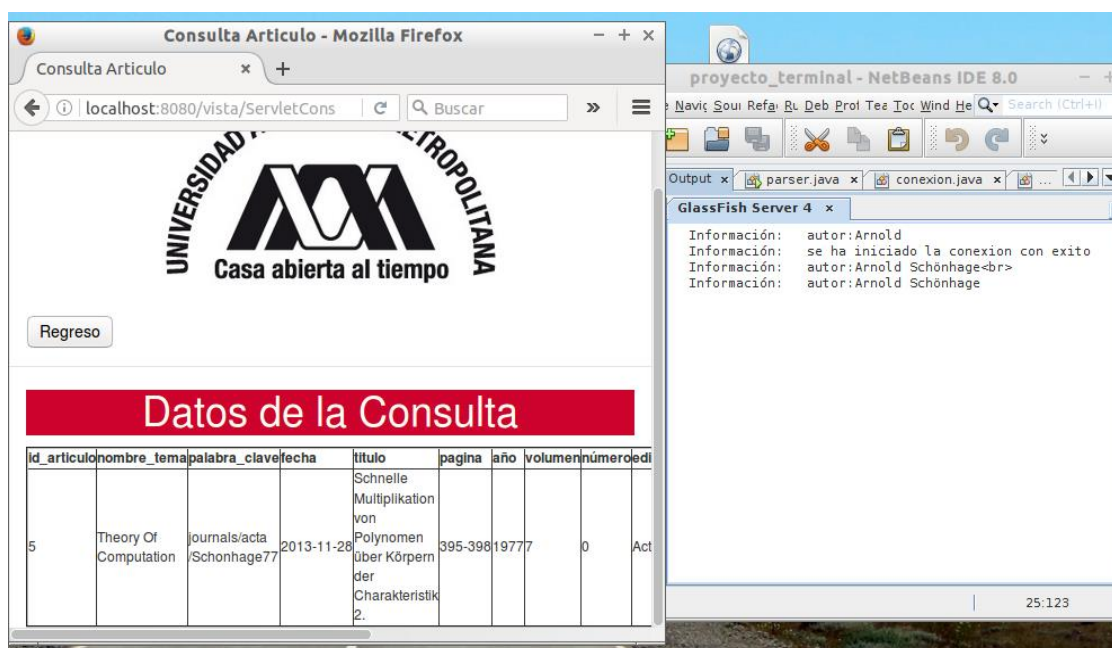
Figura 53. Consulta de la publicación científica que no contiene la cadena numero

8.2 Análisis de resultados del Módulo de Integración

En este módulo se realizaron pruebas con consultas con signos de puntuación **Figura 34** en donde el patrón a evaluar fue el nombre de un autor, se escribió el nombre del autor sin su apellido en este caso fue Arnold mostrando los datos de la consulta correctamente **Figura 35**, mientras que se realizó otra consulta por medio de su nombre completo Arnold Schönhage **Figura 36** donde no se mostró ningún dato de la consulta **Figura 37**.

Si se puede verificar en la **Figura 35** que el nombre del autor aparece en la columna nombre de autor y contiene el nombre completo del autor con todo signo de puntuación

Se analizó que las consultas implementadas en el lenguaje JAVA no reconocen las cadenas con signos de puntuación o caracteres especiales, aunque la consulta es efectuada correctamente por el servidor de Glassfish.



The screenshot shows a web browser window titled 'Consulta Artículo - Mozilla Firefox' displaying the results of a query. The page header includes the logo of 'UNIVERSIDAD METROPOLITANA' and the slogan 'Casa abierta al tiempo'. Below the header is a 'Regreso' button. The main content area is titled 'Datos de la Consulta' and contains a table with the following data:

id_articulo	nombre_tem	palabra_clave	fecha	titulo	pagina	año	volumen	número	edi
5	Theory Of Computation	journals/acta/Schönhage77	2013-11-28	Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2.	395-398	1977	7	0	Act

To the right of the browser window is the NetBeans IDE 8.0 interface, showing the 'GlassFish Server 4' output window with the following log messages:

```
Información: autor:Arnold
Información: se ha iniciado la conexión con éxito
Información: autor:Arnold Schönhage<br>
Información: autor:Arnold Schönhage
```

Figura 54. Respuesta del servidor GlassFish de la consulta de prueba sin signo de puntuación

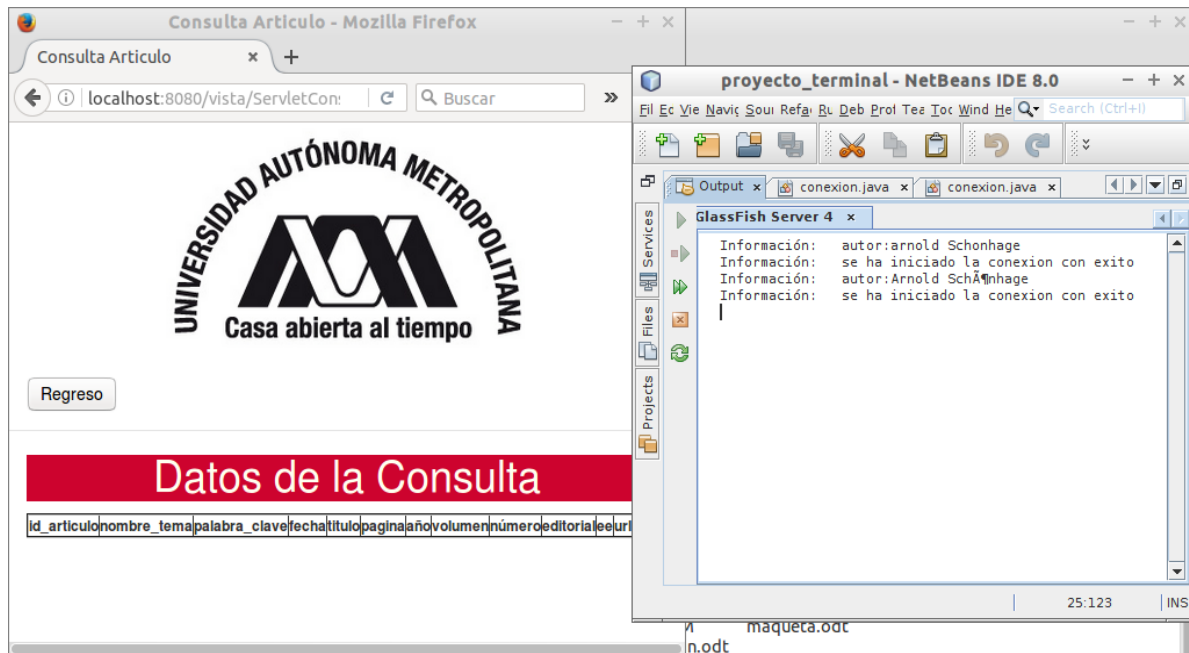


Figura 55. Respuesta del servidor Glassfish de la consulta de prueba con signo de puntuación

En la **Figura 54** se puede apreciar la consulta efectuada (autor: Arnold) después se inicia la conexión hacia la base de datos y posteriormente se obtiene la información de la consulta (autor:Arnold Schönhage
...) demostrando que se realizó correctamente la visualización en la tabla.

Por otro lado, en la **Figura 55** se realiza dos consultas (autor: Arnold Schönhage) y la de (autor: Arnold Schonhage) después se realiza la conexión hacia la base de datos pero posteriormente en ambas consultas no se efectúa ninguna otra acción después de la conexión no muestra la información como muestra en la consulta de la **Figura 54**, hay que notar el carácter especial contenido en la letra donde tiene el signo de puntuación como se mencionó anteriormente no reconoce el patrón a evaluar.

Por lo tanto, como se especificó en el módulo de integración para quitar esta problemática se utilizó el código de codificación para quitar los signos de puntuación de las tablas artículo y autor **Apéndice 7**, con este código empleado en el Parser Dom al momento de realizar consultas con signos de puntuación no se visualiza los datos de la consulta porque no se encontró el atributo en la base de datos.

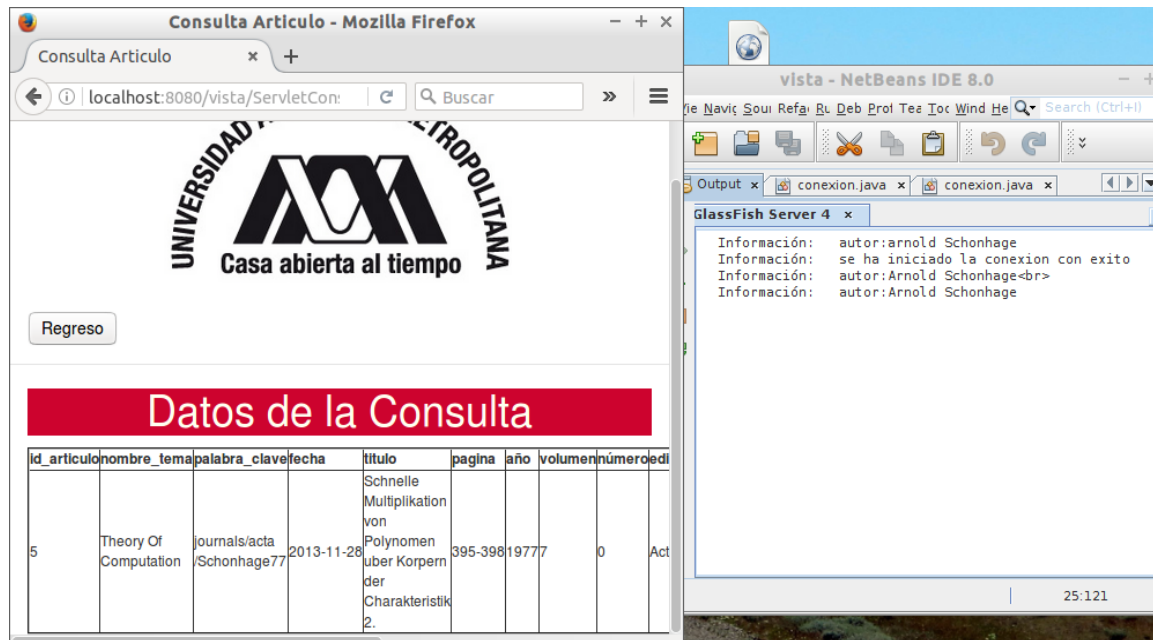


Figura 56. Respuesta del servidor Glassfish de la consulta con los datos corregidos

En la **Figura 56** se realizar de nuevo la consulta (autor: Arnold Schonhage) en donde se visualiza con éxito los datos de la consulta. Así mismo con las demás cadenas como el titulo se puede realizar consultas sin tener el conflicto de los signos de puntuación o caracteres especiales.

8.3 Análisis de resultados del Módulo de Minería de Datos

En este módulo se realizaron dos consultas como pruebas en este caso las consultas se realizan por medio del nombre del tema.

La primera consulta **Figura 38** se realiza por medio del nombre de tema Software que es el patrón a evaluar mientras en la segunda consulta **Figura 41** se evaluó como patrón Security and Privacy, en al momento de colocar el patrón a evaluar en la vista consulta de clasificación esta manda la petición a un Servlet de consultas en donde está ligado con la consulta implementada en el lenguaje JAVA

En la **Figura 57** se muestra la consulta implementada en el lenguaje JAVA

```
ResultSet rs=st.executeQuery("SELECT id_articulo,nombre_tema,palabras_claves,fecha,titulo, "+
    "pagina,año,volumen,editorial,numero,ee,url FROM articulo,acm_temas "
    + "WHERE articulo.idtema = acm_temas.idtemas AND nombre_tema like '%" + nombre_tema + "%'");
//articuloVO articulo1 = new articuloVO();
```

Figura 57. Consulta implementada en lenguaje Java para la clasificación

Después de analizar la consulta (nombre_tema) se dirige a la vista de mostrar consulta el procedimiento se explicará con más detalle en el análisis de resultado del módulo de evaluación de patrones.

En la **Figura 58** se muestra el análisis de la relación entre la tabla artículo que contiene las publicaciones científicas y la tabla de acm_temas para verificar la clasificación que cada publicación científica se va dirigida hacia un tema.

247	247	A Review and Analysis of the Usability of Data Management Environments.	7	7	Information Systems
248	248	The Effect of Language Design on Time-Sharing Operational Efficiency.	2	2	Computer Systems Organization
249	249	Analysing the Novice Analyst: Cognitive Models in Software Engineering.	4	4	Software
250	250	A User Enquiry Model for DSS Requirements Analysis: A Framework and Case Study	11	11	Applied Computing
251	251	EXCOVE and using videos in knowledge elicitation.	9	9	Human Centered Computing
252	252	Effects of "gender" of the computer on informational social influence: the moderati	12	12	Social And Professional Topics
253	253	A Knowledge-Based Approach to Computer-Aided Learning.	11	11	Applied Computing
254	254	Effects of time pressure and communication environment on team processes and o	2	2	Computer Systems Organization
255	255	Using Computers in a Natural Language Mode for Elementary Education.	11	11	Applied Computing
256	256	Neural network-based decision class analysis for building topological-level influence	3	3	Networks
257	257	An interactional perspective on group awareness: Alleviating the information-exchai	7	7	Information Systems
258	258	Affective interaction: How emotional agents affect users.	12	12	Social And Professional Topics
259	259	Development and Validation of a Reader-Based Documentation Measure.	11	11	Applied Computing
260	260	Attribute grammars as a robust technical basis for a human-computer interaction g	9	9	Human Centered Computing
261	261	Model-based communicative acts: human-computer collaboration in supervisory coi	9	9	Human Centered Computing
262	262	Stars, Polygons and Clusters; An Investigation of Polygon Displays.	6	6	Mathematics Of Computing
263	263	Security practitioners in context: Their activities and interactions with other stakeh	8	8	Security And Privacy

Figura 58. Relación entre la tabla acm_temas con articulo

La relación es correcta por medio de la llave foránea de la tabla articulo (idtema) hacia la llave

primaria de la tabla acm_temas (id temas) como se especificó en el módulo de minería de datos la clasificación se efectuó pro medio de los títulos de las publicaciones científicas.

8.4 Análisis de resultados del Módulo de Evolución de Patrones

En este módulo se realizaron tres pruebas por medio del número de id de la publicación científica, por su título y por autor. Se escogieron tres patrones a evaluar por medio del título se eligió la palabra Pattern, por autor el apellido del autor Simón y por número de id el número 23.

En el módulo de evaluación de patrones se demuestra el procedimiento a seguir **Figura 21**, en este caso al momento de colocar el patrón Pattern en la vista de consultar título este manda la petición al Servlet que manda el patrón a evaluar a la consulta y posteriormente manda la información a las vista que contiene las tablas, en la tabla de la prueba de módulo de evolución de patrones, en la **Figura 45** se puede apreciar que si encontró los resultados en los títulos que contiene el patrón o similar.

Si se nota en el código **Apéndice 15**, en la **Figura 59** se muestra la evaluación por medio de dos consultas una para evaluar el patrón en donde esta unidas la tabla articulo y la tabla acm_temas y luego se evalúa en la consulta de autor por medio de numero de id de la primera consulta, se hizo de esta manera para poder obtener y concatenar todos los autores que contiene la publicación científica de la tabla artículo.

```
ResultSet rs=st.executeQuery("SELECT id_articulo,nombre_tema,palabras_claves,fecha,titulo, "+
    "pagina,año,volumen,editorial,numero,ee,url FROM articulo,acm_temas "
    + "WHERE articulo.idtema = acm_temas.idtemas AND titulo LIKE '"+titulo+"%");

ResultSet rsl = st1.executeQuery("SELECT nombre_autor FROM autor"
    + "WHERE articulo_id_articulo = "+rs.getInt("id_articulo")+");

String autorN= "";
while(rsl.next()){
    autorN= rsl.getString("nombre_autor") + "<br>" + autorN;
}
```

Figura 59. Evaluación de consultas por medio de titulo

El mismo procedimiento con dos consultas se efectúa en la consulta por medio del número de id **Figura 48**, una consulta evalúa el número de id en este caso es 23 mientras que la otra consulta evalúa los autores que contiene la publicación científicas por medio del número del id de la primera consulta los resultados son correctos ya que despliega toda la información correcta de la publicación científica **Figura 49**.

En la **Figura 60** se muestra el procedimiento que realiza la consulta para evaluar el patrón del número de id 23

```

ResultSet rs=st.executeQuery("SELECT id_articulo, nombre_tema,palabras_claves,fecha,titulo, "+
"pagina,año,volumen,editorial,numero,es,url FROM articulo, acm_temas "
+ "WHERE articulo.idtema=acm_temas.idtemas AND id_articulo="+id_articulo+"");

ResultSet rs1 = st1.executeQuery("SELECT nombre_autor FROM autor "
+ "WHERE articulo_id_articulo = "+rs.getInt(id_articulo)+"");

String autorN= "";
while(rs1.next()){
autorN= rs1.getString("nombre_autor") + "<br>" + autorN;
}

```

Figura 60. Evaluación de consultas por medio de numero de id

En la última prueba consulta por medio del autor **Figura 46** se realiza con tres consulta, la primera consulta se evalúa el patrón en este caso el nombre del apellido Simón, mientras que la segunda consulta se evalúa los datos de la publicaciones científicas de la tabla articulo por medio del número de id de la tabla de autor y por último se vuelve a evaluar con la consulta de autores por el mismo número de id de la tabla autor para poder mostrar los autores que contiene la publicación científica los resultados con este procedimiento son correctos visualizados en la **Figura 47**.

En la **Figura 61** se muestra el procedimiento que realiza la consulta para evaluar el patrón del nombre de autor simón

```

ResultSet rs=st.executeQuery("SELECT nombre_autor, articulo_id_articulo FROM autor "
+ "WHERE nombre_autor LIKE "+nombre_autor+"");

ResultSet rs1 = st1.executeQuery("SELECT nombre_tema,palabras_claves,fecha,titulo, "+
"pagina,año,volumen,editorial,numero,es,url FROM articulo,acm_temas "
+ "WHERE articulo.idtema=acm_temas.idtemas AND id_articulo="+rs.getInt( articulo_id_articulo)+"");

ResultSet rs2=st2.executeQuery("SELECT nombre_autor FROM autor "
+ "WHERE articulo_id_articulo = "+rs.getInt( articulo_id_articulo)+"");

```

Figura 61. Evaluación de consultas por medio de autor

La consulta de autor es diseñada con tres consultas, mientras que las consultas por título, número de id, volumen, número, nombre del tema y fecha fueron realizadas con dos consultas.

8.5 Análisis de resultados del Módulo de Presentación

En este módulo se efectuó una consulta donde se evaluó un patrón sencillo una letra del alfabeto en este caso fue la letra c **Figura 48**, esto se efectuó para poder mostrar una mayor cantidad de datos en la tabla datos de la consulta. **Figura 49**.

En este caso se realiza la evaluación de patrones busca todas las palabras o referencia que contenga la letra c en el nombre del tema que se relaciona con las publicaciones científicas se realizó con dos consultas la primera consulta la evaluación del nombre del tema uniendo la tabla articulo con la tabla acm_temas y después se analiza con la consulta de los autores.

Figura 62 se muestra el procedimiento que realiza la consulta para evaluar el patrón en este caso fue una letra de alfabeto c como nombre de autor

```
ResultSet rs=st.executeQuery("SELECT id_articulo,nombre_tema,palabras_claves,fecha,titulo, "+
    "pagina,año,volumen,editorial,numero,ee,url FROM articulo,acm_temas "
    + "WHERE articulo.idtema = acm_temas.idtemas AND nombre_tema like '%nombre_tema+'");
//articulo() articulo() = new articulo();

ResultSet rsl = st1.executeQuery("SELECT nombre_autor FROM autor"
    + " WHERE articulo_id_articulo =" +rs.getInt("id_articulo")+"");
```

Figura 62. Evaluación de consultas por medio de nombre del tema

9 Conclusiones

En el proyecto de investigación se enfocó en extracción y representación de datos relevantes de las diferentes publicaciones científicas que se encontraron en el archivo XML

Sus aportaciones principales son las siguientes:

- 1) Extracción exitosa de información relevante del archivo XML por medio de un Parser Dom programado en lenguaje JAVA.
- 2) Llenado exitosamente de las tablas de la base de datos con los datos recopilados.
- 3) Clasificación efectuada correctamente según los temas propuestos por el ACM CSS 2012.
- 4) Diseño e implementación satisfactoriamente de las consultas en el lenguaje JAVA, así como la evaluación de patrones realizada con éxito.
- 5) Diseño de vistas adecuadas para la fácil interacción para el usuario.
- 6) Representación adecuada según las peticiones realizada por el usuario por medio de tablas en vista JSP.

Se realizaron suficientes pruebas tanto en la base de datos como en las consultas y en las vistas de representación para poder tener un mejor desempeño de la interfaz del proyecto.

Posiblemente se pueda reutilizar la base de datos para proyectos en un futuro.

10 Referencias Bibliográficas

- [1]"Fichas Publicas", Conacyt.gob.mx,2016.[Online]. Available: <http://www.conacyt.gob.mx/index.php/transparencia/transparencia-focalizada/fichas-publicas>. [Accessed:14-Sep-2016].
- [2]N. Guzmán Gonzales, "Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información", Licenciatura, Proyecto Terminal, Universidad Autónoma Metropolitana Azcapotzalco, 2011.
- [3]F. Guiño Maripérez, "Representación Semántica y Extracción de Información sobre Publicaciones en Expedientes Curriculares" Licenciatura, Proyecto Terminal, Universidad Autónoma Metropolitana Azcapotzalco, 2014.
- [4]G. Poma Hernández, "Descubrimiento de Relaciones de Datos Abiertos de Perfiles Académicos y Profesionales mediante una Ontología" Licenciatura, Propuesta de Proyecto Terminal, Universidad Autónoma Metropolitana Azcapotzalco, 2015
- [5]Osmar R. Zaine "Introduction to Data Base" CMPUT690 Principle of Knowledge Discovery in DataBases. University of Alberta, pp 1-15 , 1999
- [6] C. Clifton, R .Cooley and J. Rennie, "Top Cat: Data Mining for Topic Identification Text Corpus", IEEE Transactions on Knowledge and Data Engineering. Clifton et Al, Unites States, New York, pp. 949-964, 2004.
- [7]M.Servente and R.García Martínez, "Algoritmo TDIDT Aplicados a la Minería de Datos Inteligentes", Ingeniería en Informática, Facultad de Ingeniería Universidad de Buenos Aires, 2002
- [8] José A. Reyes Ortiz "Descubrimiento y representación de conocimiento sobre datos científicos y académicos", Departamento de Sistemas, Universidad Autónoma Metropolitana Unidad Azcapotzalco
- [9] "The 2012 ACM Computing Classification System – Introduction", General Approach to the 2012 classification Revision. <http://www.acm.org/about/class>
- [10] DataMing, http://www.sinnexus.com/business_intelligence/datamining.aspx
- [11] Definicion abc Clasificacion, <https://www.definicionabc.com/general/clasificacion.php>
- [12] Informatica pc, Patrones de Diseño, <https://informaticapc.com/patrones-de-diseno/>
- [13]Desarrollo, web JSP, <https://desarrolloweb.com/articulos/831.php>
- [14]Oracle, API JBDC, <http://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/>
- [15]Java Servlet, <https://www.javatpoint.com/servlet-tutorial>
- [16] Archivos XML, <http://www.mundolinux.info/que-es-xml.htm>
- [17] Servidor Glassfish, <https://bannysolano.wordpress.com/2009/08/23/%C2%BFque-es-glassfish/>

[18] Netbeans y Eclipse, <https://es.wikipedia.org/wiki/NetBeans>

[19]Parser Dom, https://www.tutorialspoint.com/java_xml/java_dom_parser.htm

[20]ACM, https://www.tutorialspoint.com/java_xml/java_dom_parser.htm

Apéndices

Apéndice 1. Este apéndice presenta el código fuente para la implementación de Tablas Entidad-Relación en SQL, Worbench MYSQL.

```
CREATE TABLE `acm_temas` (  
  `idtemas` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_tema` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`idtemas`)  
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;  
  
CREATE TABLE `articulo` (  
  `id_articulo` int(11) NOT NULL AUTO_INCREMENT,  
  `palabras_claves` varchar(90) DEFAULT NULL,  
  `fecha` varchar(45) DEFAULT NULL,  
  `titulo` varchar(300) DEFAULT NULL,  
  `pagina` varchar(25) DEFAULT NULL,  
  `año` int(11) DEFAULT NULL,  
  `volumen` varchar(25) DEFAULT NULL,  
  `editorial` varchar(300) DEFAULT NULL,  
  `numero` int(11) DEFAULT NULL,  
  `ee` varchar(90) DEFAULT NULL,  
  `url` varchar(90) DEFAULT NULL,  
  `idtema` int(11) NOT NULL,  
  PRIMARY KEY (`id_articulo`),  
  KEY `articulo_idtema_idx` (`idtema`)  
) ENGINE=InnoDB AUTO_INCREMENT=301 DEFAULT CHARSET=latin1;  
  
CREATE TABLE `autor` (  
  `id_autor` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_autor` varchar(45) DEFAULT NULL,  
  `articulo_id_articulo` int(11) NOT NULL,  
  PRIMARY KEY (`id_autor`),  
  KEY `fk_autor_articulo1_idx` (`articulo_id_articulo`),  
  CONSTRAINT `fk_autor_articulo1` FOREIGN KEY (`articulo_id_articulo`)  
REFERENCES `articulo` (`id_articulo`) ON DELETE NO ACTION ON UPDATE NO  
ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=689 DEFAULT CHARSET=latin1;
```

Apéndice 2. Conexión hacia la Base de Datos desde el lenguaje JAVA

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/proyecto_termina
l", "root","");
```

Apéndice 3. Programa articuloVO para referenciar las columnas de la tabla artículo

```
public articuloVO(String palabras_claves, String fecha, String titulo,
String pagina, int año,int volumen,
String editorial, int numero, String url1, String eel)
    this.palabras_claves = palabras_claves ;
    this.fecha = fecha;
    this.titulo = titulo;
    this.pagina = pagina;
    this.año= año;
    this.volumen = volumen;
    this.editorial= editorial;
    this.numero = numero;
    this.eel = eel;
    this.url1 = url1;
        public int getid_articulo()
        return id_artículo;
        public void setid_articulo(int id_articulo)
        this.id_articulo=id_articulo;
        public String getpalabras_claves()
        return palabras_claves;
        public void setpalabras_claves(String palabras_claves)
        this.palabras_claves = palabras_claves;
        public void setfecha(String fecha)
        this.fecha=fecha;
    public String getfecha()
        return fecha
    public String gettitulo()
        return titulo;
    public void settitulo(String titulo)
        this.titulo= titulo;
    public String getpagina()
        return pagina;
    public void setpagina(String pagina)
        this.pagina = pagina;
        public int getaño()
        return año;
    public void setaño( int año)
        this.año=año;
    public int getvolumen()
        return volumen;
    public void setvolumen(int volumen)
        this.volumen = volumen;
    public int getnumero()
        return numero;
```



```

        public void setnumero(int numero)
this.numero = numero;
        public String geteditorial()
            return editorial;
public void setresumen(String editorial)
            this.editorial = editorial;
public String getee()
            return ee1;
        public void setee(String ee1)
            this.ee1=ee1;
            public String geturl()
                return url1;
        public void seturl(String url1)
            this.url1=url1;
        public int getidgenerado()
return idgenerado;

```

Apéndice 4. Programa autorVO para referenciar las columnas de la tabla autor.

```

private String nombre_autor;
private int id_articulo;
private int id_autor;
public autorVO( String nombre_autor)
this.nombre_autor=nombre_autor;
public void autor()
        public int getid_autor()
            return id_autor;
        public void setid_autor(int id_autor)
this.id_autor=id_autor;
        public String getnombre_autor()
            return nombre_autor;}
public void setnombre_autor(String nombre_autor)
this.nombre_autor=nombre_autor;
        public int getid_articulo(int id)
            return id;}
        public void setid_articulo(int id)
            this.id_articulo=id;

```

Apéndice 5. Programa insertar_articulo para realizar la inserción hacia la tabla artículo.

```

        PreparedStatement pst =cn.prepareStatement ("INSERT INTO
articulo(palabras_claves, fecha, titulo, pagina, año, volumen, editorial, numero, ee, url)
values(?,?,?,?,?,?,?,?,?,?)", PreparedStatement.RETURN_GENERATED_KEYS);
//boolean agregado=false;

//pst.setInt            (1, art.getid_articulo());
pst.setString         (1, art.getpalabras_claves());
//pst.setString        (3, art.getautor());
pst.setString         (2, art.getfecha());
pst.setString         (3, art.gettitulo());
pst.setString         (4, art.getpagina());
pst.setInt            (5, art.getaño());
pst.setInt            (6, art.getvolumen());

```



```

        NodeList author =
articuloElement.getElementsByTagName("author");
        NodeList title =
articuloElement.getElementsByTagName("title");
        NodeList pages =
articuloElement.getElementsByTagName("pages");
        NodeList year = articuloElement.getElementsByTagName("year");
        NodeList volume =
articuloElement.getElementsByTagName("volume");
        NodeList journal =
articuloElement.getElementsByTagName("journal");
        NodeList number =
articuloElement.getElementsByTagName("number");
        NodeList url = articuloElement.getElementsByTagName("url");
NodeList ee = articuloElement.getElementsByTagName("ee");
NodeList sub= articuloElement.getElementsByTagName("sub");
        int numero=0;
        for (int n = 0; n <number.getLength(); n++)
            if (null ==
(number.item(n).getFirstChild().getTextContent()))
                numero =
Integer.parseInt(number.item(n).getFirstChild().getTextContent());
        String eel = null;
        for (int e = 0; e <ee.getLength(); e++) {
            eel = ee.item(e).getFirstChild().getTextContent();
            System.out.println("ee:" +
ee.item(e).getFirstChild().getTextContent());
            int volumen=0;
            for (int v = 0; v <volume.getLength(); v++)
                if (null ==
(volume.item(v).getFirstChild().getTextContent()))
                    volumen =
Integer.parseInt(number.item(v).getFirstChild().getTextContent());
                    volumen =
Integer.parseInt(volume.item(v).getFirstChild().getTextContent());
                int aÃ±o=0;
                for(int y=0;y<year.getLength();y++)
                    aÃ±o =
Integer.parseInt(year.item(y).getFirstChild().getTextContent());

                String output_titulo ="";
                String subtítulo = "";
                for(int su=0;su<sub.getLength();su++){
                    if(!sub.item(su).getFirstChild().getTextContent().isEmpty()){
                        subtítulo = sub.item(su).getFirstChild().getTextContent();
                    }
                    for(int t=0; t<title.getLength();t++){
                        titulo = title.item(t).getFirstChild().getTextContent();
                        // sub=title.item(t).getFirstChild().getTextContent();
                        System.out.println("titulo:" +
title.item(t).getFirstChild().getTextContent());
                        output_titulo = titulo+subtítulo;
                    }
                    String original =
"Ã©ÃªÃ·Ã¸Ã¹ÃºÃ»Ã¼Ã½Ã¾Ã¿ÃÀÃÁÃÂÃÃÃÄÃÅÃÆÃÇÃÈÃÉÃÊÃËÃÌÃÍÃÎÃÏÃÐÃÑÃÒÃÓÃÔÃÕÃÖÃ×ÃØÃÙÃÚÃÛÃÜÃÝÃÞÃßÃàÃáÃâÃãÃäÃåÃæÃçÃèÃéÃêÃëÃìÃíÃîÃïÃðÃñÃòÃóÃôÃõÃöÃ÷ÃøÃùÃúÃûÃüÃýÃþÃß";
                // Cadena de caracteres ASCII que reemplazarÃ;n los originales.

```

```

        String ascii =
"AAAAAAACEEEEEIIIIIDN00000UUUUYBaaaaaaaceeeeiiiiionooooouuuuyy";
for (int x=0; x<original.length(); x++) {
output_titulo = output_titulo.replace(original.charAt(x),
ascii.charAt(x));
        String paginas = null;
for(int p=0; p<pages.getLength();p++){
paginas= pages.item(p).getFirstChild().getTextContent();
        // fecha
=pages.item(p).getAttributes().getNamedItem("mdate").getNodeValue();
        System.out.println("pages:" +
pages.item(p).getFirstChild().getTextContent());

        String resumen =null;
for(int r=0 ; r<journal.getLength();r++){
        resumen=journal.item(r).getFirstChild().getTextContent();
        System.out.println("journal:" +
journal.item(r).getFirstChild().getTextContent());
        String url1=null;
for(int u=0; u<url.getLength();u++){
        url1= url.item(u).getFirstChild().getTextContent();
        System.out.println("url:" +
url.item(u).getFirstChild().getTextContent());

!articuloElement.getAttributes().getNamedItem("mdate").getNodeValue().isEmpty(){
int ACM_Temas_idACM_Temas=0;
String fecha=
articuloElement.getAttributes().getNamedItem("mdate").getNodeValue();
        String
palabras_claves=articuloElement.getAttributes().getNamedItem("key").getNodeValue();
        articuloVO art = new
articuloVO(palabras_claves,fecha,output_titulo, paginas, año, volumen,
resumen,numero, url1, eel /*,ACM_Temas_idACM_Temas*/);
System.out.println("articulo"+art);
insertar_articulo cn = new insertar_articulo();
int id=cn.insercion_articulo(art);
System.out.println("id recuperado:" + id);
for(int a=0;a<author.getLength();a++){
if(!author.item(a).getFirstChild().getTextContent().isEmpty()){
        String nombre_autor=
author.item(a).getFirstChild().getTextContent();
String output_nombre = nombre_autor;
        String original =
"Ã©Ã±Ã,ÃfÃ,,Ã...Ã+Ã+Ã^Ã%:ÃŠÃ<Ã€Ã¼ÃžÃ¼Ã¼Ã'Ã'Ã"Ã"Ã•Ã-
Ã~Ã™ÃšÃ »ÃœÃ¼ÃÝÃ ã;ÃçÃfÃçÃÿÃ!ÃŠÃ"Ã©ÃªÃ«Ã¬Ã-
Ã®Ã¯Ã°Ã±Ã²Ã³Ã´ÃµÃ¶Ã·Ã¸Ã¹ÃºÃ»Ã¼Ã½Ã¾";
        String ascii =
"AAAAAAACEEEEEIIIIIDN00000UUUUYBaaaaaaaceeeeiiiiionooooouuuuyy";
for (int x=0; x<original.length(); x++) {
output_nombre = output_nombre.replace(original.charAt(x),
ascii.charAt(x));

        System.out.println("cadena:" + output_nombre);
        System.out.println("cadena autor:" + nombre_autor);
autorVO au = new autorVO(output_nombre);

```

```

        System.out.println("autor"+au);
insertar_autor cna = new insertar_autor();
cna.insercion_autor(au,id);

```

Apéndice 9. Consultas UPDATE para la inserción de numero de id de los temas para relacionarlos con las publicaciones científicas (artículo) correspondiente.

```

UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='5' WHERE
`id_articulo`='1';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='2' WHERE
`id_articulo`='3';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='3' WHERE
`id_articulo`='2';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='5' WHERE
`id_articulo`='4';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='5' WHERE
`id_articulo`='5';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='6';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='7';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='3' WHERE
`id_articulo`='8';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='7' WHERE
`id_articulo`='9';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='10';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='7' WHERE
`id_articulo`='11';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='4' WHERE
`id_articulo`='12';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='3' WHERE
`id_articulo`='13';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='8' WHERE
`id_articulo`='14';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='5' WHERE
`id_articulo`='15';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='11' WHERE
`id_articulo`='16';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='12' WHERE
`id_articulo`='17';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='18';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='10' WHERE
`id_articulo`='19';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='20';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='7' WHERE
`id_articulo`='21';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='6' WHERE
`id_articulo`='22';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='10' WHERE
`id_articulo`='23';
UPDATE `Proyecto_Terminal`.`articulo` SET `idtema`='2' WHERE
`id_articulo`='24';

```

Apéndice 10. Código de autorVO implementado para las consultas en el lenguaje JAVA.

```
public class autorVO
private String nombre_autor;
private int id_autor;
private int articulo_id_articulo;
    public autorVO(String nombre_autor, int id_autor, int
articulo_id_articulo)
        this.nombre_autor = nombre_autor;
        this.id_autor = id_autor;
        this.articulo_id_articulo = articulo_id_articulo;
        public int getid_autor()
            return id_autor;
            public void setid_autor(int id_autor)
                this.id_autor=id_autor;
        public String getnombre_autor()
            return nombre_autor;
        public void setnombre_autor(String nombre_autor)
            this.nombre_autor=nombre_autor;
        public int getarticulo_id_articulo()
            return articulo_id_articulo;
        public void setarticulo_id_articulo(int articulo_id_articulo)
            this.articulo_id_articulo = articulo_id_articulo;
```

Apéndice 11. Código de articuloVO implementado para las consultas en el lenguaje JAVA.

```
    public articuloVO( int id_articulo,String palabras_claves, String
fecha, String titulo,
    String pagina, int año, int volumen, String editorial, int numero,
String url1, String eel,
    String autor, int articulo_id, String acm_temas, int articulo_idacm,int
acm_temas_idacm_temas)

        this.palabras_claves = palabras_claves ;
        this.id_articulo =id_articulo;
        this.fecha = fecha;
        this.titulo = titulo;
        this.pagina = pagina;
        this.año = año;
        this.volumen = volumen;
        this.editorial= editorial;
        this.numero = numero;
        this.eel = eel;
        this.url1 = url1;
        this.autor = autor;
        this.articulo_id =articulo_id;
        this.articulo_idacm =articulo_idacm;
        this.acm_temas=acm_temas;
        this.acm_temas_idacm_temas = acm_temas_idacm_temas;

        public int  getacm_temas_idacm_temas()
return acm_temas_idacm_temas;
        Public void setacm_temas_idacm_temas(int acm_temas_idacm_temas)
this.acm_temas_idacm_temas=acm_temas_idacm_temas;
```

```

    public int getarticulo_idacm()
        return articulo_idacm;
    public void setautor(int articulo_idacm)
        this.articulo_idacm=articulo_idacm;
public String getautor()
    return autor;
    public void setautor(String autor)
        this.autor=autor;
public int getarticulo_id()
    return articulo_id;
public void setarticulo_id(int articulo_id)
    this.articulo_id=articulo_id;
        public int getid_articulo()
            return id_articulo;
        public void setid_articulo(int id_articulo)
            this.id_articulo=id_articulo;
        public String getpalabras_claves()
            return palabras_claves;
    public void setpalabras_claves(String palabras_claves)
        this.palabras_claves = palabras_claves;
    public void setfecha(String fecha)
        this.fecha=fecha;
public String getfecha()
    return fecha;
    public String gettitulo()
        return titulo;
    public void settitulo(String titulo)
        this.titulo= titulo;
public String getpagina()
    return pagina;
        public void setpagina(String pagina)
            this.pagina = pagina;
    public int getaÑto()
        return aÑto;
    public void setaÑto(int aÑto)
        this.aÑto=aÑto;
public int getvolumen()
    return volumen;
    public void setvolumen(int volumen)
        this.volumen = volumen;
    public int getnumero()
        return numero;
    public void setnumero(int numero)
        this.numero = numero;
public String geteditorial()
    return editorial;
    public void seteditorial(String editorial)
        this.editorial = editorial;
public String getee()
    return ee;
    public void setee(String ee)
        this.ee=ee;
public String geturl()
    return url;
    public void seturl(String url)
        this.url=url;

```

```

    public int getidgenerado()
return idgenerado;
    public ArrayList<autorVO> getautorVOList()
        return autorLista;
    public void setautorVOList(ArrayList<autorVO> autorVOList)
        this.autorLista=autorVOList;
        public ArrayList<acmtemasVO> getacm_temasVOList()
            return acm_temaslista;
        public void setacm_temasVOList(ArrayList<acmtemasVO>
acm_temasVOList)
            this.acm_temaslista=acm_temasVOList;
    public String getacm()
        return acm_temas;
    public void setacm(String acm_temas)
        this.acm_temas=acm_temas;

```

Apéndice 12. Código de acmtemasVO implementado para las consultas en el lenguaje JAVA.

```

    private int idacm_temas;
    private String nombre_tema;
    public acmtemasVO()
        public acmtemasVO(int idacm_temas, String nombre_tema)
            this.idacm_temas = idacm_temas;
            this.nombre_tema = nombre_tema;
    public int getid_acm_temas()
        return idacm_temas;
    public void setid_acm_temas(int idacm_temas)
        this.idacm_temas = idacm_temas;
    public String getnombre_tema()
        return nombre_tema;
    public void setnombre_tema(String nombre_tema)
        this.nombre_tema = nombre_tema;

```

Apéndice 13. Consulta implementada en lenguaje JAVA para la evaluación de Patrones.

```

    ResultSet rs=st.executeQuery("SELECT id_articulo,
nombre_tema,palabras_claves,fecha,titulo,"+
        "pagina,año,volumen,editorial,numero,ee,url FROM
articulo, acm_temas WHERE articulo.idtema=acm_temas.idtemas AND
id_articulo='"+id_articulo+"'");
    while(rs.next()){
        System.out.println("ResultSet:" +rs);
        articuloVO articulo = new articuloVO();
        acm_tema=rs.getString("nombre_tema");
        articulo.setid_articulo(rs.getInt("id_articulo"));
        articulo.setpalabras_claves(rs.getString("palabras_claves"));
        articulo.setfecha(rs.getString("fecha"));
        articulo.settitulo(rs.getString("titulo"));
        articulo.setpagina(rs.getString("pagina"));
        articulo.setaño(rs.getInt("año"));
        articulo.setvolumen(rs.getInt("volumen"));

```



```

        articulo.seteditorial(rs.getString("editorial"));
        articulo.setnumero(rs.getInt("numero"));
        articulo.setee(rs.getString("ee"));
        articulo.seturl(rs.getString("url"));
        ResultSet rs1 = st1.executeQuery("SELECT nombre_autor
FROM autor WHERE articulo_id_articulo = "+rs.getInt("id_articulo")+""");
        String autorN= "";
        while(rs1.next()){
            autorN= rs1.getString("nombre_autor") + "<br>" + autorN;
        autor.setarticulo_id_articulo(rs.getInt("articulo_id_articulo"));
        articulo.setacm(acm_tema);
        articulo.setautor(autorN);
        articulo.setautorVOLList(autorVOLList);
        articulo.setacm_temasVOLList(acm_temasVOLList);
        bid.add(articulo);
        return bid;
    }
}

```

Apéndice 14. Consulta implementada en lenguaje JAVA para la evaluación de Patrones por medio del autor

```

        ResultSet rs=st.executeQuery("SELECT
nombre_autor,articulo_id_articulo FROM autor WHERE nombre_autor LIKE
'%" + nombre_autor + "%'");
        while(rs.next()){
            articuloVO articulo = new articuloVO();
            autor =rs.getString("nombre_autor");
            articulo_id =rs.getInt("articulo_id_articulo");
            ResultSet rs1 = st1.executeQuery("SELECT
nombre_tema,palabras_claves,fecha,titulo,"+
"pagina,año,volumen,editorial,numero,ee,url FROM
articulo,acm_temas WHERE articulo.idtema=acm_temas.idtemas AND
id_articulo="+rs.getInt("articulo_id_articulo")+""");
            while(rs1.next()){
                acm_tema= rs1.getString("nombre_tema");
                articulo.setpalabras_claves(rs1.getString("palabras_claves"));
                articulo.setfecha(rs1.getString("fecha"));
                articulo.settitulo(rs1.getString("titulo"));
                articulo.setpagina(rs1.getString("pagina"));
                articulo.setaño(rs1.getInt("año"));
                articulo.setvolumen(rs1.getInt("volumen"));
                articulo.seteditorial(rs1.getString("editorial"));
                articulo.setnumero(rs1.getInt("numero"));
                articulo.setee(rs1.getString("ee"));
                articulo.seturl(rs1.getString("url"));
                String autorN ="";
                ResultSet rs2=st2.executeQuery("SELECT nombre_autor FROM
autor WHERE articulo_id_articulo =
'" +rs.getInt("articulo_id_articulo")+ "'");
                while(rs2.next()){
                    autorN=rs2.getString("nombre_autor") + "<br>" + autorN;
                    articulo.setautor(autorN);
                    articulo.setarticulo_id(articulo_id);
                    articulo.setacm(acm_tema);
                }
            }
        }
    }
}

```

```

        articulo.setacm_temasVOLList(acm_temasVOLList);
        System.out.println("autor:"+autorN);
        System.out.println("autor:"+autor);
        articulo.setautorVOLList(autorVOLList);
        bautor.add(articulo);
    return bautor;

```

Apéndice 15. Consulta implementada en lenguaje JAVA para la evaluación de Patrones para la clasificación

```

    ResultSet rs=st.executeQuery("SELECT
id_articulo,nombre_tema,palabras_claves,fecha,titulo,"+
        "pagina,año,volumen,editorial,numero,ee,url FROM
articulo,acm_temas WHERE articulo.idtema = acm_temas.idtemas AND
nombre_tema like '%" + nombre_tema + "%'");
    while(rs.next()){
        articuloVO articulo= new articuloVO();
        acm_tema=rs.getString("nombre_tema");
        articulo.setid_articulo(rs.getInt("id_articulo"));
        articulo.setpalabras_claves(rs.getString("palabras_claves"));
        articulo.setfecha(rs.getString("fecha"));
        articulo.settitulo(rs.getString("titulo"));
        articulo.setpagina(rs.getString("pagina"));
        articulo.setaño(rs.getInt("año"));
        articulo.setvolumen(rs.getInt("volumen"));
        articulo.seteditorial(rs.getString("editorial"));
        articulo.setnumero(rs.getInt("numero"));
        articulo.setee(rs.getString("ee"));
        articulo.seturl(rs.getString("url"));
        ResultSet rs1 = st1.executeQuery("SELECT nombre_autor FROM
autor WHERE articulo_id_articulo ="+rs.getInt("id_articulo")+");
        String autorN="";
        while(rs1.next())
            autorN= rs1.getString("nombre_autor") + "<br>" + autorN;
        articulo.setautor(autorN);
        articulo.setautorVOLList(autorVOLList);
        articulo.setacm(acm_tema);
        articulo.setacm_temasVOLList(acm_temasVOLList);
        btema.add(articulo);
    return btema;

```

Apéndice 16. Código de vista principal del proyecto terminal en lenguaje de HTML

```
<title>Proyecto Terminal: Minería de datos aplicadas a
información sobre publicaciones científicas</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1,maximum-scale=1,user-scalable=no">
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="../css/bootstrap.min.css">
<div class="container">
<h1>Universidad Autónoma Metropolitana
Unidad Azcapotzalco</h1>

<div class="container">
<h1>Proyecto Terminal</h1>
<h2>Minería de datos aplicadas a información
sobre publicaciones científicas</h2>
<div class="container-fluid">
<section class="main row">
<aside class="col-xs-12 col-sm-4 col-md-3">
<p>Asesor: José Alejandro Reyes Ortiz </p>
<p>Nombre co-Asesora: Mireya Tovar Vidal
<p>Alumno: Adrian Campa Vilchis</p>
<p>Reseña: En este proyecto se realizó una
clasificación por medio de ACM
(Computing Classification System) y se efectuó
varias consultas de las publicaciones científicas.
<article class="col-xs-12 col-sm-8 col-md-9">
<form action="consulta_acm_temas.jsp" method="post">
<h3> Consulta de Clasificación de un Artículo
Científico</h3>
<h4> <input value="Consultar Clasificación"
type="submit"></h4>
<form action="menu.html" method="post">
<h3> Consulta de un Artículo Científico</h3>
<h4><input value="Consulta Artículo" type="submit"></h4>
```

Apéndice 17. Código de vista menú del proyecto terminal en lenguaje de HTML

```
<title>Menú de Consultas</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="../css/bootstrap.min.css">
<link rel="stylesheet" href="css/estilo.css">

<div class="container-fluid">
<div class="panel panel-danger">
<div class="panel-heading"><h1> Menú de Consultas </h1> </div>
<div class="panel-body">
<a href="http://localhost:8080/vista/index.html" > <input
type="submit" value="Regreso a Menu Principal" ></a>
```

```

<form action = "consulta_id_articulo.jsp" method="post">
  <h3>Consultar por n mero de ID</h3>
    <h4> <input value="Consultar" type="submit"></h4>
    <form action = "consulta_titulo.jsp" method="post">
  <h3>Consultar por titulo </h3>
  <h4><input value="Consultar" type="submit" ></h4>
    <form action = "consulta_fecha.jsp" method="post">
  <h3>Consultar por fecha</h3>
    <h4> <input value="Consultar" type="submit"></h4>
    <form action = "consulta_numero.jsp" method="post">
  <h3>Consultar por n mero</h3>
    <h4><input value="Consultar" type="submit"></h4>
    <form action = "consulta_volumen.jsp" method="post">
  <h3>Consultar por volumen</h3>
    <h4><input value="Consultar" type="submit"></h4>
    <form action = "consulta_autor.jsp" method="post">
  <h3>Consultar por autor</h3>
    <h4> <input value="Consultar" type="submit"></h4>

```

Ap ndice 18. C digo de vista principal de consulta por nombre de tema de la clasificaci n del proyecto terminal en lenguaje de HTML

```


  <div class="container-fluid">
    <div class="panel panel-danger">
      <div class="panel-heading"> <h1>Consultar Clasificaci n de
Articulos </h1>
      <div class=" panel-body">
        <div class ="modal-header">
          <a href="http://localhost:8080/vista/index.html" > <input
type="button" value="Regreso a Men  Principal"></a>
        <div align="center">
<form action="ServletConsulta_ACM_Temas" method="get">
  <td colspan="2" align="center"><h3>Consultar Clasificaci n de
Articulos</h3></td>
  Tipo de temas:
  Hardware,
  Computer Systems Organization,
  Networks,
  Software,
  Theory Of Computing,
  Information Systems,
  Security And Privacy,
  Human Centered Computing,
  Computing Methodologies,
  Applied Computing,
  Social And Professional Topics.
  <td> <h4>nombre_tema:</h4></td>
  <td><input type= "text" size="25" name="nombre_tema"></td>
<td colspan="2" align="center"><input type="submit"
value="CONSULTAR"></td>

```

Apéndice 19. Código de vista principal de consulta para las publicaciones científicas del proyecto terminal en lenguaje de HTML

```

  <div class="container-fluid">
<div class="panel panel-danger">
  <div class="panel-heading"> <h1>Consultar Articulos </h1> </div>
  <div class="panel-body">
    <div class="modal-header">
      <a href="http://localhost:8080/vista/menu.html" > <input
type="button" value="Regreso a MenÃ° de Consultas"></a>
<div align="center">
<form action="ServletConsulta_id_articulo" method="get">
  <td colspan="2" align="center"><h3>Consultar Articulo por numero de
ID</h3> </td>
  <td><h4>numero de ID:</h4></td>
  <td><input type="text" size="25" name="id_articulo"></td>
<td colspan="2" align="center"><input type="submit"
value="CONSULTAR"></td>
```

Apéndice 20. Servlet para la consulta en el lenguaje JAVA

```
@WebServlet("/ServletConsulta_Titulo")
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String titulo=request.getParameter("titulo");
    ArrayList<articuloVO>
btitulo=consulta_titulo.consultarArticuloPorTitulo(titulo);
    if(btitulo!=null){
        request.setAttribute("btitulo", btitulo);

request.getRequestDispatcher("mostrarconsulta_titulo.jsp").forward(reques
t, response);
```

Apéndice 21. Código de vista para mostrar consulta del proyecto terminal en lenguaje HTML.

```
<a href="http://localhost:8080/vista/consulta_fecha.jsp" > <input
type="button" value="Regreso"></a>
  <h1> Datos de la Consulta</h1>
  <%
    ArrayList<articuloVO>
bfecha=(ArrayList<articuloVO>)request.getAttribute("bfecha");
    ArrayList<autorVO> autorVOList = new ArrayList();
    ArrayList<acmtemasVO> acm_temasVOList = new ArrayList();
  %>

  <center><table border="2">
    <tr>
      <th>id_articulo</th>
      <th>nombre_temas</th>
      <th>palabra_clave</th>
      <th>fecha</th>
      <th>titulo</th>
```

```

        <th>pagina</th>
        <th>aÃ±o</th>
        <th>volumen</th>
        <th>nÃºmero</th>
        <th>editorial</th>
        <th>ee</th>
        <th>url</th>
        <th>autores</th>

</tr>
<%   for(articuloVO articulo : bfecha){ %>

<tr>
    <td><%= articulo.getId_articulo() %></td>
    <td><%= articulo.getacm()%></td>
    <td><%= articulo.getpalabras_claves() %></td>
    <td><%= articulo.getfecha() %></td>
    <td><%= articulo.gettitulo() %></td>
    <td><%= articulo.getpagina() %></td>
    <td><%= articulo.getaÃ±o() %></td>
    <td><%= articulo.getvolumen() %></td>
    <td><%= articulo.getnumero() %></td>
    <td><%= articulo.geteditorial() %></td>
    <td><%= articulo.getee() %></td>
    <td><%= articulo.geturl() %></td>
    <td><%= articulo.getautor() %></td>

    <%
        autorVList=articulo.getautorVList();
        for(int i=0;i<autorVList.size();i++){
            System.out.println(autorVList.get(i));
            %>
            <td><%=articulo.getautorVList() %></td>

            <%   }%>

            <%
//ArrayList<articuloVO> articuloVList= new ArrayList();
acm_temasVList=articulo.getacm_temasVList();
for(int j=0;j<autorVList.size();j++){
    System.out.println(acm_temasVList.get(j));
    %>
    <td><%=articulo.getacm_temasVList() %></td>

            <%   }%>
    
```