

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División Ciencias Básicas e Ingeniería

Reporte Final del Proyecto de Integración

Licenciatura en Ingeniería en Computación

Proyecto de Integración:
Proyecto tecnológico

Sistema para identificar la ubicación e intensidad de los síntomas de un paciente a partir de notas médicas

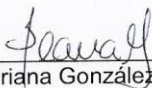
César Eduardo Juárez García
210204941

Asesor
Dra. Beatriz Adriana González Beltrán
Profesor Asociado

Co asesor
Dr. José Alejandro Reyes Ortiz
Profesor Asociado

Jueves 28 de abril del 2016

Yo, Beatriz Adriana González Beltrán, 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.



Beatriz Adriana González Beltrán

Yo, 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, César Eduardo Juárez García, doy mi autorización a la Coordinación de Servicios de Información de la universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el siguiente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



César Eduardo Juárez García

Resumen

Una aplicación es un programa de computadora que se utiliza como herramienta para una operación o tarea específica. A diferencia de otros programas como los sistemas operativos, los lenguajes de programación y otros, la aplicación tiene el único y principal fin de realizar una tarea específica, a menudo básica y de rápido y fácil uso para el usuario común.

La razón más frecuente para la creación de una aplicación informática es la necesidad de resolver un problema o de simplificar una operación compleja. En definitiva, una aplicación informática sirve para ahorrar tiempo y dinero al usuario y, por eso, constantemente nuevas aplicaciones son desarrolladas ya sea por usuarios básicos, avanzados o programadores.

El proyecto tecnológico realizado es una aplicación que permite obtener información relevante de notas médicas en archivos de texto, como el diagnóstico o síntoma del mismo, así como el lugar afectado y la intensidad de la molestia.

Esto se realiza de manera masiva, lo que permite un ahorro de tiempo significativo, pues basta con seleccionar una carpeta para que todos los archivos en ella sean analizados.

Tabla de contenido

Resumen	3
1. Introducción	6
2. Antecedentes.....	7
2.1 Referencias internas	7
Representación semántica y extracción de información sobre publicaciones en expedientes curriculares [1].	7
Sistema Web para identificar eventos y actores en textos periodísticos [2].	7
Representación semántica de información espacial y temporal a partir de textos periodísticos mediante reglas lingüísticas [3].	7
Mediciones de similitud semántica aplicados a resúmenes de artículos científicos [4].	8
2.2 Referencias externas	8
Extracción de información con Algoritmos de Clasificación [5].	8
Minería de datos: Conceptos y tendencias [6].	9
3. Justificación	9
4. Objetivos	10
4.1 Objetivo general	10
4.2 Objetivos específicos	10
5. Marco Teórico.....	10
5.1 5.1 Nota médica	10
5.2 5.2 Extracción de información	11
6. Desarrollo de proyecto.....	11
6.1 Diagrama de casos de uso	12
6.2 Diagrama de bloques	14
6.2.1 Módulo de extracción de síntomas.	14
6.2.2 Módulo de extracción de lugar e intensidad del síntoma.	15
6.2.3 Módulo de representación de datos extraídos.	15
6.3 Uso del sistema	16
7. Software y hardware necesarios	24
7.1 Software	24
7.2 Hardware	24
8. Resultados	24
9. Análisis y discusión de resultados.....	25
10. Conclusiones	25
11. Perspectivas del proyecto.....	26
12. Bibliografía.....	26
13. Apéndice A.....	27

Tabla de figuras

Ilustración 1 Diagrama de casos de uso	12
Ilustración 2 Arquitectura del sistema para la extracción y representación de síntomas a partir de notas médicas en texto libre en español.....	14
Ilustración 3 Resultado de los tres módulos descritos.....	15
Ilustración 4 Base de datos de la aplicación.....	16
Ilustración 5 Pantalla principal de la aplicación.	16
Ilustración 6 Ventana de búsqueda de carpeta.	17
Ilustración 7 Dirección donde se ubica la carpeta.....	17
Ilustración 8 Mensaje de confirmación: "Guardado en la base de datos".....	18
Ilustración 9 Notas guardadas en la base de datos.....	18
Ilustración 10 Ventana de resultados del proceso de análisis.	19
Ilustración 11 Agregar diagnóstico.....	20
Ilustración 12 Lista de diagnósticos agregados al diccionario.....	21
Ilustración 13 Agregar a lugares.	21
Ilustración 14 Lista de lugares agregada al diccionario.....	22
Ilustración 15 Agregar a intensidades.	22
Ilustración 16 Lista de intensidades agregada al diccionario.....	23
Ilustración 17 Cargar archivo de texto con un listado de síntomas.	24

1. Introducción

El cuerpo humano está constituido por varios órganos, estos se estructuran en varios sistemas o aparatos para desempeñar tareas determinadas. Si alguno de estos órganos llega a tener complicaciones, crea un malestar, el cual tiene que ser atendido y valorado por un médico para su tratamiento correspondiente. El médico expide una nota médica como resultado de la consulta del paciente.

El manejo de la información es algo integrado en la práctica clínica. Médicos y pacientes interactúan en una compleja matriz de información donde el médico es un "manipulador" de la información en el sentido de que la adquiere, la procesa, almacena, revisa y la aplica en relación a la historia y evolución del paciente.

Por otro lado, la extracción de información es una tarea del área de la lingüística computacional cuyo objetivo es extraer determinados tipos de información de un documento, los cuales están basados en dominios específicos debido a que extraen información relevante. Esta tarea tiene cierto grado de dificultad y requiere de la aplicación de algoritmos computacionales.

Con ayuda de algoritmos computacionales se facilitará la extracción y representación de información esencial al médico. Partiendo de las notas médicas, se extraerán varios términos de interés tales como: síntoma, lugar del síntoma e intensidad del mismo. Estos términos serán almacenados de manera persistente para su posterior recuperación y manipulación.

2. Antecedentes

2.1 Referencias internas

Representación semántica y extracción de información sobre publicaciones en expedientes curriculares [1].

El proyecto consiste en la clasificación de expedientes curriculares para después con ayuda de técnicas de minería de datos extraer la información sobre investigaciones a partir de los expedientes curriculares descritos en lenguaje natural en español y así facilitar la búsqueda de información relevante.

Al igual que en este proyecto se aplicará técnicas de minería de datos para la extracción de síntomas, lugar del síntoma e intensidad del mismo. La diferencia es que la técnica de minería de datos se aplicará en notas médicas en texto libre en español.

Sistema Web para identificar eventos y actores en textos periodísticos [2].

En este proyecto se diseñó un sistema de extracción de información a partir de textos periodísticos que existe en la red Internet. Este proyecto utiliza técnicas de minería de datos para la extracción de temas de interés. Este proyecto es útil a las personas interesadas en textos periodísticos dado que es más fácil localizar los temas de su interés.

La diferencia es que a partir de las notas médicas se llevará a cabo la técnica de minería de datos utilizando reconocimiento de entidades nombradas y expresiones regulares, para extraer términos de interés y facilitar la manipulación de la información de las notas médicas.

Representación semántica de información espacial y temporal a partir de textos periodísticos mediante reglas lingüísticas [3].

En este proyecto se diseñó e implementó un sistema que realiza la extracción de información en textos periodísticos nacionales mediante reglas lingüísticas facilitando el acceso de grandes colecciones de datos textuales y mejorando así la productividad en las tareas de análisis y síntesis de información.

La diferencia con este proyecto se implementará un sistema para la recuperación de información, partiendo de notas médicas en texto libre y así ayudar a tener una mejor toma de decisiones clínicas por parte de los médicos.

Mediciones de similitud semántica aplicados a resúmenes de artículos científicos [4].

Consiste en buscar artículos de investigación similares que sean de gran ayuda a los autores para su trabajo. En el proyecto se diseñó e implemento una herramienta de software que compare los resúmenes de varios artículos de investigación por medio de medición de similitud semántica con el objetivo de obtener una tabla indicando el grado de similitud que se ha descubierto entre los resúmenes de los artículos científicos, con esto el investigador podrá seleccionar aquellos artículos que más se parezcan para tomarlos como referencia a su trabajo.

La diferencia con este proyecto es que se creará una lista con nombres de síntomas y otra más con nombres de las partes del cuerpo humano donde se origina el síntoma. Para después realizar la comparación con las notas médicas y sea más fácil localizar la información que se desea extraer.

2.2 Referencias externas

Extracción de información con Algoritmos de Clasificación [5].

En esta tesis se dan las bases teóricas para la extracción de información, nos muestra los algoritmos que se deben de utilizar, entre otras cosas que prácticamente te llevan de la mano para crear un sistema de extracción de datos eficiente. El cual será de gran ayuda para la elaboración de este proyecto que llevaremos a cabo con la implementación de módulos para la extracción de información.

Minería de datos: Conceptos y tendencias [6].

En este artículo, nos habla de diferentes cuestiones acerca de la introducción a la minería de datos como son: definición, ejemplificación de problemas que se pueden resolver con minería de datos, las tareas de la minería de datos, técnicas usadas y finalmente retos y tendencias en minería de datos.

3. Justificación

Hoy en día, la tecnología ha revolucionado la manera de extraer información con ayuda de diferentes herramientas computacionales. Estas herramientas son útiles para brindar una solución en un hospital donde se maneja un gran número de notas médicas que a medida que pasan los días van aumentando gradualmente. Para ello, con este proyecto se requiere implementar un algoritmo computacional que nos ayude a extraer cierta información de las notas y almacenarlas en una base de datos.

Considerando la cantidad de pacientes que atiende cada médico en un hospital, podemos imaginar el gran número de notas médicas que son generados día a día. Asumiendo lo anterior y tomando en cuenta la cantidad de notas médicos que produce cada persona a lo largo de toda su vida. Podemos darnos cuenta que extrayendo información primordial de cada una de las notas médicas, al observar los síntomas obtenidos de un paciente en un lapso de tiempo, puede ayudar a tener una mejor toma de decisiones clínicas por parte de los médicos y así dar una mejor atención a los pacientes.

4. Objetivos

4.1 Objetivo general

Diseñar e implementar un sistema para identificar la ubicación y la intensidad de los síntomas de un paciente a partir de notas médicas.

4.2 Objetivos específicos

- Analizar y diseñar un módulo para la extracción de nombres de síntomas a partir de notas médicas utilizando reconocimiento de entidades nombradas (NER).
- Analizar y diseñar un módulo para la extracción del lugar de síntoma y la intensidad del mismo con ayuda de técnicas de reconocimiento de entidades nombradas y expresiones regulares.
- Implementar un sistema computacional de procesamiento y representación de texto para la extracción de síntomas, lugar e intensidad del mismo.
- Diseñar y ejecutar un plan experimental para la validación del sistema en la tarea de extracción de información.

5. Marco Teórico

5.1 5.1 Nota médica

Una nota médica es el documento legal en el cual, el médico describe detalladamente la consulta realizada al paciente.

El médico debe identificar la causa principal por la que el paciente está siendo atendido. Así mismo deberá llenar la nota médica, de lo contrario, el sistema no podrá determinar las causas por las que el paciente asiste a consulta.

Para tal acción, el sistema contiene una búsqueda especializada de diagnósticos con el Catálogo Internacional de Enfermedades (CIE-10), la cual permite localizar de una manera ágil el diagnóstico identificado.

En la nota médica puede incluirse más de una enfermedad presentada por el paciente, pero sólo una debe ser identificada como “Principal”, ya que esto

alimentará reportes de Vigilancia Epidemiológica y la impresión automática en la Hoja Diaria de consulta externa.

5.2 5.2 Extracción de información

La gran cantidad de textos existentes en formato electrónico en los últimos tiempos, hace imposible que una persona pueda leer, comprender y sintetizar toda la información contenida en ellos.

Para gestionar esta información, se aplican estrategias tales como la Extracción de Información. En grandes conjuntos de textos, se intenta encontrar y extraer información que satisfaga las necesidades de información de un usuario.

El objetivo de la Extracción de Información es construir sistemas que encuentren y relacionen información relevante. Que es relevante se determina mediante guías predefinidas de un cierto dominio, especificadas con la mayor precisión posible.

Prácticamente, consiste en localizar las porciones de texto que contengan información relevante para una necesidad concreta de un usuario/s y proporcionar dicha información de forma adecuada para su proceso (de forma manual o automática).

6. Desarrollo de proyecto

Este proyecto llamado: “Sistema para identificar la ubicación e intensidad de los síntomas de un paciente a partir de notas médicas”, es una aplicación que permite analizar y extraer información relevante de notas médicas, a través de un algoritmo computacional. Esta aplicación representa un proyecto tecnológico dirigido por profesores del Área de Investigación en Sistemas de Información Inteligentes de la División de Ciencias Básicas e Ingeniería de la Universidad Autónoma Metropolitana, quienes tienen interés en proponer soluciones computacionales en el ámbito de la salud.

6.1 Diagrama de casos de uso

La figura 1 muestra dos actores, el usuario como actor principal y una Base de Datos como actor secundario. El actor principal puede interactuar con las funcionalidades de la interfaz: seleccionar ubicación de los archivos de texto, cargar archivos a la base de datos, analizar los archivos, consultar la base de datos, agregar síntoma, agregar lugar donde ocurre el síntoma y agregar la intensidad del mismo. Mientras que el actor secundario solo puede interactuar con el sistema a través del almacenamiento de los archivos y el almacenamiento de la información extraída de dichos archivos.

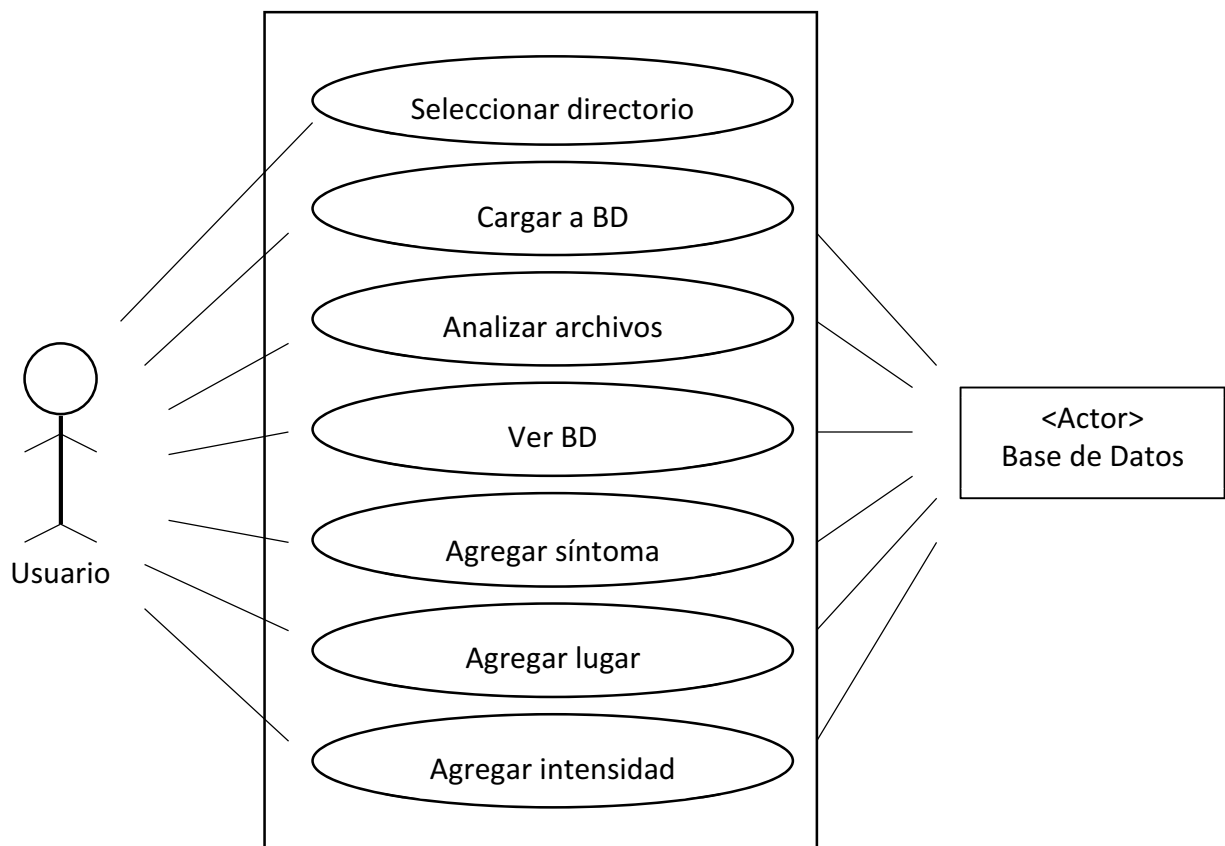


Ilustración 1 Diagrama de casos de uso

A continuación se presenta el caso de uso “Analizar nota médica”, el cual servirá de ejemplo para la manipulación de la interfaz: “Sistema de análisis de notas médicas”.

Caso de uso: Analizar nota médica

Resumen	Este caso de uso permite al usuario analizar notas médicas.
Actor	Principal Usuario
Precondiciones	Haber iniciado el programa.
Descripción	<p>Flujo principal</p> <ol style="list-style-type: none"> 1. El programa muestra la interfaz. 2. El usuario selecciona el directorio donde se encuentran las notas médicas. 3. El usuario cargar las notas a la base de datos. 4. El usuario presiona la opción de analizar. 5. La interfaz muestra la base de datos. 6. El usuario puede agregar palabras a los diccionarios. <p>Flujo alternativo</p> <p>Paso 4</p> <ol style="list-style-type: none"> 4.1 El usuario selecciona la opción “ver análisis”. 4.2 El usuario selecciona la opción “ver BD”. <p>Paso 6</p> <ol style="list-style-type: none"> 6.1 El usuario selecciona la opción de “agregar síntoma”. 6.2 El usuario selecciona la opción de “agregar lugar”. 6.3 El usuario selecciona la opción de “agregar intensidad”.
Postcondiciones	El análisis queda guardado en la base de datos y las listas de los diccionarios se actualizan.

6.2 Diagrama de bloques

En la Figura 2 se muestran los módulos que forman parte del proyecto.

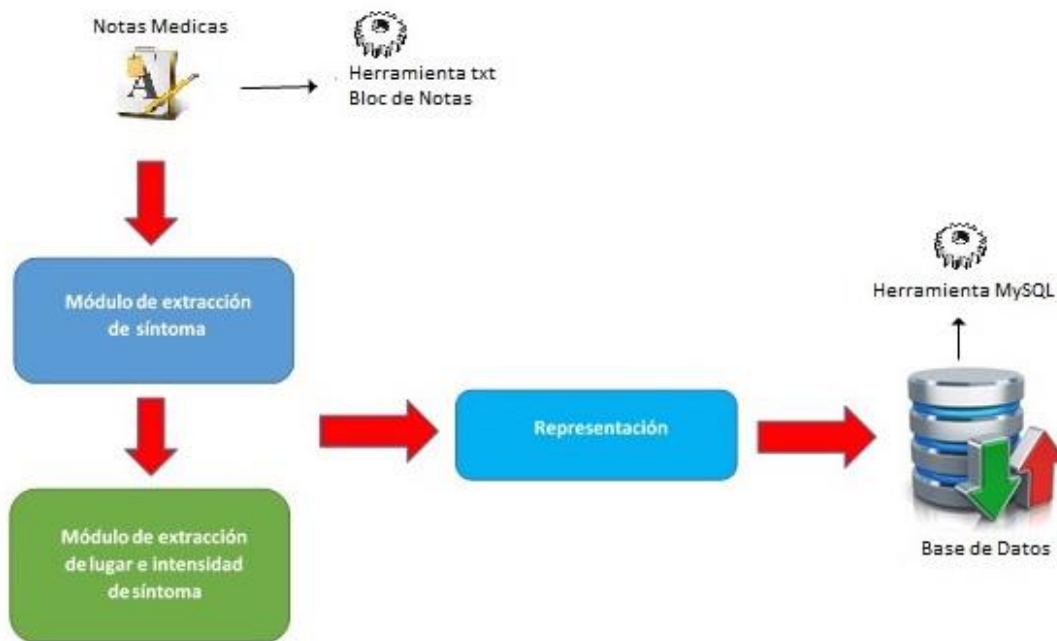


Ilustración 2 Arquitectura del sistema para la extracción y representación de síntomas a partir de notas médicas en texto libre en español.

6.2.1 Módulo de extracción de síntomas.

En la realización de este módulo, se comenzó por analizar 500 notas médicas de forma manual, con el propósito de obtener una lista de síntomas conocidos. Con la cual se formó el diccionario de síntomas para llevar a cabo el análisis.

Una vez hecho el diccionario de síntomas, se creó un método en java llamado "private void btnAnalizarActionPerformed", que funciona de la siguiente manera: comienza por recorrer las cadenas de texto e ir reconociendo palabras, cuando obtiene un palabra la compara con el diccionario de síntomas hasta encontrar coincidencia alguna con algunos de los síntomas declarados en el diccionario de síntomas. Si la palabra concuerda con alguna de la lista, automáticamente se extrae la información y se almacena en la base de datos.

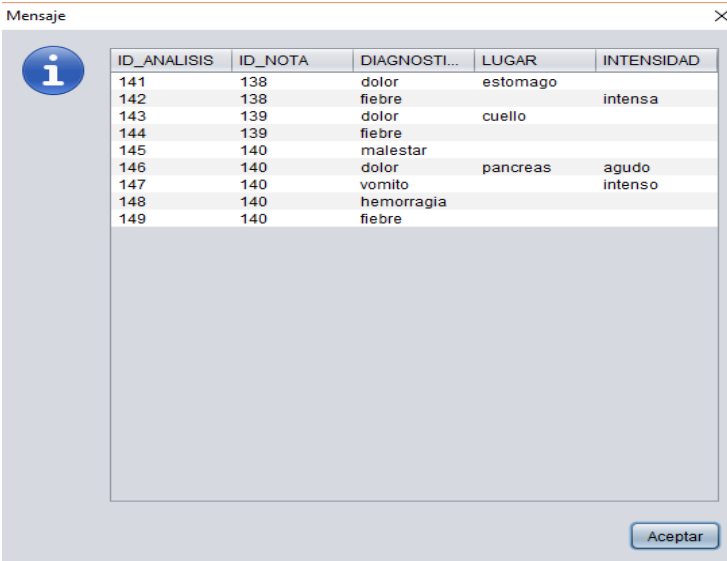
6.2.2 Módulo de extracción de lugar e intensidad del síntoma.

De igual modo que en el módulo anterior, se realizó un análisis para obtener las listas de lugar e intensidad de los síntomas para la creación de los diccionarios pertinentes.

Usando el mismo método del módulo anterior “private void btnAnalizarActionPerformed”. Se realizó un proceso similar al de extracción de síntomas, para realizar de igual forma la extracción de lugar e intensidad de los síntomas. Haciendo trabajar en conjunto los tres procesos para obtener un mejor resultado. Diseñando un algoritmo computacional con diferentes casos y/o condiciones, que nos ayudan a obtener un análisis más exhaustivo.

6.2.3 Módulo de representación de datos extraídos.

Una vez extraída la información, se creó una Base de Datos llamada “notasmedicas” en MySQL, para almacenar toda la información obtenida. Cabe mencionar que para apreciar de una mejor manera la representación de la información, se diseñó un interfaz grafica para facilitar la visualización y manejo de dicha información.



ID_ANALISIS	ID_NOTA	DIAGNOSTI...	LUGAR	INTENSIDAD
141	138	dolor	estomago	
142	138	fiebre		intensa
143	139	dolor	cuello	
144	139	fiebre		
145	140	malestar		
146	140	dolor	pancreas	agudo
147	140	vomito		intenso
148	140	hemorragia		
149	140	fiebre		

Ilustración 3 Resultado de los tres módulos descritos.

La aplicación maneja una base de datos que se compone de cinco tablas:



Ilustración 4 Base de datos de la aplicación

6.3 Uso del sistema

La aplicación consta de varios elementos, como se puede apreciar en la figura 5:



Ilustración 5 Pantalla principal de la aplicación.

El uso de la aplicación lo explicaremos con base al caso de uso principal: Analizar nota médica.

Una vez iniciada la interfaz, da click en el botón “seleccionar carpeta”. Automáticamente aparecerá una ventana (figura 6) para buscar la carpeta donde se encuentran guardados el o los archivos que vayan analizar. Cuando se haya encontrado la carpeta, dar click en cancelar.

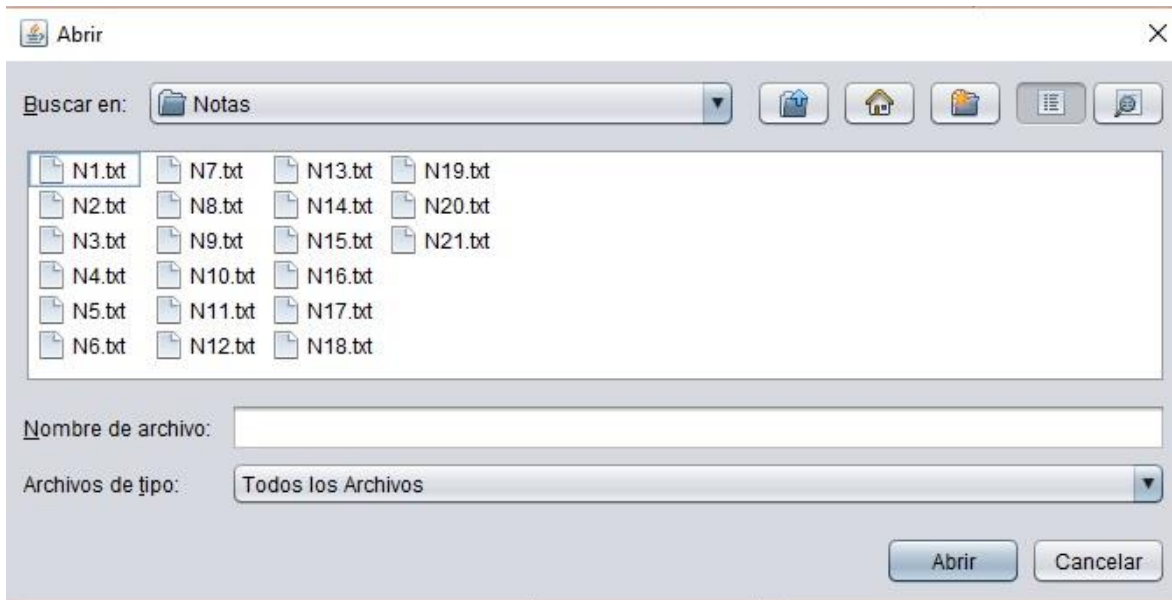


Ilustración 6 Ventana de búsqueda de carpeta.

Inmediatamente de dar click en cancelar en la caja de texto aparecerá la ruta donde se encuentra la carpeta seleccionada. Tal y como se muestra en la figura 7.

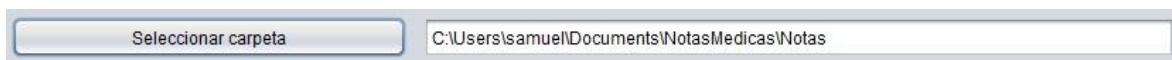


Ilustración 7 Dirección donde se ubica la carpeta.

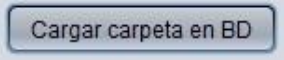
Cuando se tenga la dirección de la carpeta, dar click en el botón  y las notas médicas serán limpiadas (quitar acentos, caracteres especiales y convertir en minúsculas las letras) y guardadas en la base de datos. Al finalizar el proceso se mostrará un mensaje de aviso que nos indica que el proceso se ha realizado. Como se aprecia en la figura 8.



Ilustración 8 Mensaje de confirmación: "Guardado en la base de datos".

Para corroborar que las notas médicas han sido guardadas en la base de datos, basta con presionar el botón "Ver notas". Nos mostrará las notas que se han cargado en la base de datos (figura 9).

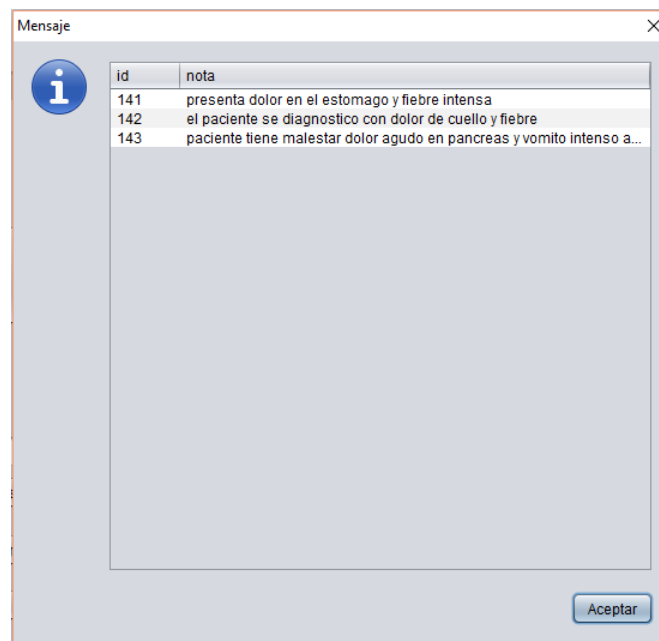
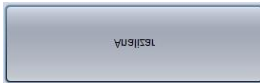


Ilustración 9 Notas guardadas en la base de datos.

A continuación, para realizar el proceso de análisis, basta con presionar el botón



y dejar que la aplicación realice el proceso.

Acabando el proceso de análisis, nos mostrará un mensaje diciéndonos que el análisis ha terminado. Posteriormente para visualizar los resultados que obtuvo, daremos click en el botón “Ver Análisis” y nos aparecerá una ventana como la de la figura10.



Ilustración 10 Ventana de resultados del proceso de análisis.

Antes de realizar el proceso de analizar, es importante mencionar que las notas médicas deben estar cargadas previamente a la base de datos, pues en caso de estar vacía la base no se llevará a cabo el análisis porque no tendrá archivos que analizar.

Al terminar el proceso de análisis, en ocasiones notaremos que no extrajo información de la nota médica. Este suceso se puede suscitar en dos casos:

Caso 1- La nota médica no contiene síntoma y/o lugar y/o intensidad del mismo.

Caso 2- Los síntomas, lugares o intensidades que contenga la nota médica, no estén declarados en las listas de los diccionarios y es por eso que no extrajo información alguna.

Tomando en cuenta el “Caso 2”, la aplicación está diseñada para agregar síntomas, lugares e intensidades, directamente de la interfaz. Como se muestra en las siguientes imágenes:

Agregar síntoma (figura 11). Escribir el síntoma en la caja de texto y posteriormente dar click en el botón “Agregar a diagnóstico”.

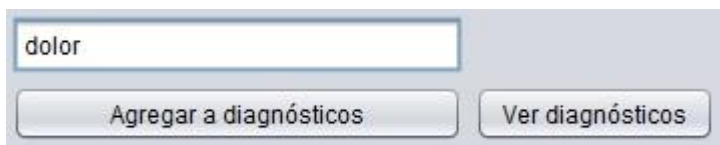


Ilustración 11 Agregar diagnóstico.

Así como también, podemos consultar los diagnósticos con tan sólo presionar el botón “Ver diagnósticos”. Tal y como se muestra en la figura 12.

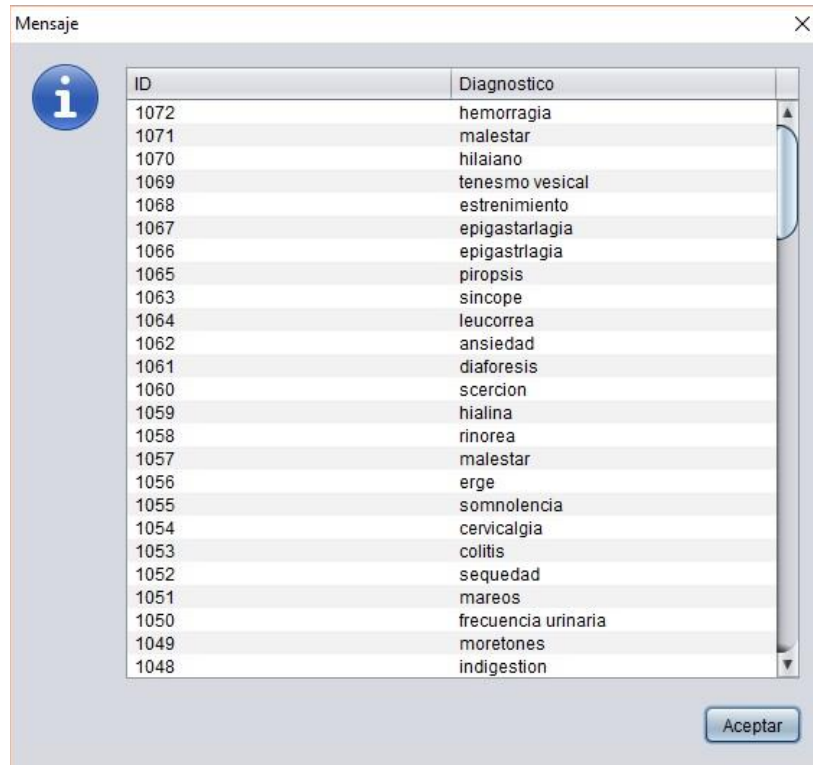


Ilustración 12 Lista de diagnósticos agregados al diccionario.

El proceso es idéntico con los casos de lugar e intensidad.

Agregar a lugares (figura 13).



Ilustración 13 Agregar a lugares.



Ilustración 14 Lista de lugares agregada al diccionario.

Agregar a intensidades (figura 15).

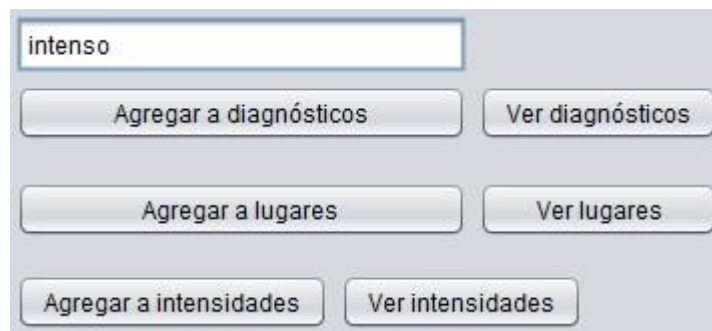
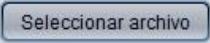


Ilustración 15 Agregar a intensidades.



Ilustración 16 Lista de intensidades agregada al diccionario.

Por último, en dado caso de tener una lista grande de síntomas que se deseen agregar al diccionario. Basta con listarlos en un archivo de texto y presionar el botón  y seleccionar la ruta donde se encuentra el archivo creado (el proceso es similar al de “Seleccionar carpeta”). Posteriormente presionar “Cargar en diagnósticos” (figura 17).

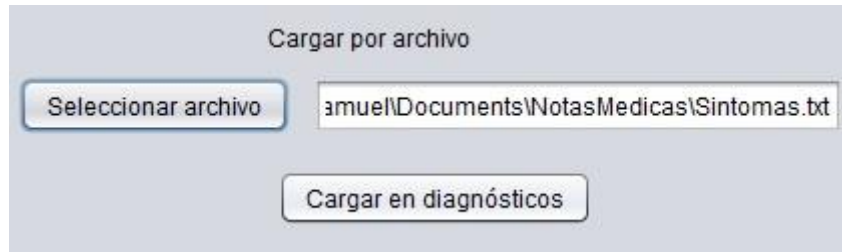


Ilustración 17 Cargar archivo de texto con un listado de síntomas.

7. Software y hardware necesarios

Para la realización del presente proyecto se utilizaron los siguientes recursos:

7.1 Software

Netbeans IDE 8.1

XAMPP with PHP 5.5.34, 5.6.20 & 7.0.5 versions

MySQLConnector

7.2 Hardware

Computadora personal con procesador de 64 bits.

8. Resultados

Los resultados son favorables, de esta manera es posible analizar una gran cantidad de notas médicas en cuestión de segundos. Evidentemente existen áreas de oportunidad a considerar en una posible segunda versión de la aplicación, como una mejor presentación o un análisis más exhaustivo, sin embargo, el mayor problema es que de no contar con una entrada en el diccionario que corresponda a la nota médica, no se puede obtener un resultado.

9. Análisis y discusión de resultados

El proyecto tecnológico desarrollado permite la extracción de información relevante de notas médicas en grandes volúmenes. Cumpliendo con los objetivos planteados. No obstante, esta aplicación aún puede ser mejorada, tanto en su código como en el diseño de su interfaz de usuario.

Uno de los problemas que se presentan es cuando las notas médicas no están estructuradas y redactadas de forma correcta. Sería recomendable hacer un algoritmo que estructure de forma adecuada las notas médicas.

Por otra parte, siento que es necesario rediseñar la interfaz gráfica. Hacerla un poco más factible para el usuario.

10. Conclusiones

Durante la realización del proyecto se presentaron varias circunstancias, tanto de programación, como referentes al mismo idioma. Inicialmente se había planteado usar expresiones regulares para identificar las notas médicas, sin embargo, al intentar usarlas, claramente no fue posible, debido a que no existe un lenguaje o una estructura que permita reconocer términos médicos de otros. Aunque bien, hay sufijos, como itis, que significa inflamación, existe una gran cantidad de palabras que no se pueden reconocer como gripe, o tos, que claramente son comunes y frecuentes. Por eso se optó por usar listas de palabras conocidas, para poder ofrecer un análisis más exacto, aunque más dependiente.

Los errores de programación y bases de datos fueron relativamente fáciles de resolver en comparación a lo mencionado, como por ejemplo, al querer insertar múltiples veces en un periodo muy corto de tiempo, la base de datos no lo soportó por defecto y fue necesario cambiar un parámetro de la misma con:

```
set global max_connections = 1000;
```

También era deseable que en la base de datos aparecieran todas las coincidencias, para tener un panorama más claro del paciente, por lo que se mejoró para obtener un mejor resultado.

Con todo lo anterior, también se puede decir que el análisis de las notas fue correcto, y es posible utilizar la aplicación no sólo para eso, sino cualquier texto en el que se pueda usar un diccionario, y por lo mismo, entre mejor el diccionario, mejores serán los resultados, pero son buenos para una primera versión.

11. Perspectivas del proyecto.

Este proyecto actualmente se encuentra en su primera fase, es un prototipo. Se espera que en un futuro la aplicación realice análisis más exhaustivos.

Se anima al futuro desarrollador a crear una interfaz más dinámica, con diseños personales y no sólo limitarse a usar estos métodos de extracción. Es importante poder diseñar íconos propios para una aplicación, así como utilizar notas médicas mejor estructuradas.

Cabría la posibilidad de hacer un mejor diseño de la base de datos para la representación de los datos.

Por último, se hace una atenta invitación al futuro desarrollador, a dar una detallada lectura a este documento para conocer el funcionamiento de este algoritmo y poder mejorarlo para un mejor desempeño.

12. Bibliografía

- [1] F.A. Gudiño Pérez, “*Representación y extracción de información sobre publicaciones en expedientes curriculares*”, proyecto tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, Distrito Federal, 2014.
- [2] L.D. Hernández Rojas, “*Sistema Web para identificar eventos y actores en textos periodísticos*”, proyecto tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2014.

- [3] J.Padilla Cuevas, “Representación semántica de información espacial y temporal a partir de textos periodísticos mediante reglas lingüísticas”, proyecto tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2014.
- [4] A. Téllez Valera, “*Mediciones de similitud semántica aplicados a resúmenes de artículos científicos*”, proyecto tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2014.
- [5] A. Téllez Valero, “*Extracción de información con algoritmos de clasificación*”, tesis de maestría, Instituto Nacional de Astrofísica Óptica y Electrónica, México, Puebla, 2005.
- [6] J.C. Riquelme, R. Ruiz, K. Gilbert, “*Minería de datos: Conceptos y tendencias*”. Revista Iberoamericana de Inteligencia Artificial, Vol. 10, no 29, p. 11-18., 2006.

13. Apéndice A

A continuación se lista parte del código desarrollado. Su funcionamiento general es el siguiente:

Se agregan entradas de síntomas, diagnósticos, lugares (partes del cuerpo) e intensidades del dolor en la base de datos, previo a la carga de las notas médicas a la misma por parte del usuario, a esto se le llaman diccionarios, pues a partir de ellos se analizan de manera eficiente las notas introducidas por el usuario.

Para analizar notas, es preciso seleccionar una carpeta en la que se encuentren, al hacer click en el botón de Seleccionar carpeta, se abre una nueva ventana.

Una vez seleccionada la carpeta, se hace click en el botón “Cargar carpeta en BD”, que obtiene los nombres de todos los archivos en ella, y después lee la totalidad de cada uno, y los agrega a la base de datos, en una sola entrada por archivo, en la tabla NOTAS, en la columna notas.

El código es el siguiente:

```
private void btnCargarCarpetaActionPerformed(java.awt.event.ActionEvent evt) {

    String ruta = txtCarpeta.getText();
    File directorio = new File(ruta);
    String[] ficheros = directorio.list();
```

```

    {

        for (int i = 0; i < ficheros.length; i++) {
            try {
                Path ubicacion = Paths.get(ruta + "\\\" + ficheros[i]);
                String nota = new String(Files.readAllBytes(ubicacion),
StandardCharsets.UTF_8);

                nota = nota.toUpperCase();
                nota = nota.replace('Á', 'A').replace('É', 'E').replace('Í',
'I').replace('Ó', 'O').replace('Ú', 'U').replace('Ñ', 'N');
                nota = nota.replaceAll("\\\", \"\").replaceAll("\\.\",
\"").replaceAll("\\-\", \"\").replaceAll("\\+\", \"\").replaceAll(\"\\\"\", \"\");
                String Insertar = nota.toLowerCase();
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/notasmedicas","root","");
                String query= "INSERT INTO NOTAS (ID, nota) VALUES (NULL,
'+Insertar+ '');"
                Statement stm = (Statement)con.createStatement();
                stm.executeUpdate(query);

            } catch (IOException e) {
                e.printStackTrace();
            } catch (SQLException ex) {
                Logger.getLogger(interfaz.class.getName()).log(Level.SEVERE, null,
ex);
            }

        }

        lblInfo.setText("Guardado en la base de datos.");
    } // TODO add your handling code here:
}

```

Al mismo tiempo, se reemplazan o eliminan los caracteres inadecuados, como los acentos y los puntos, para un mejor análisis.

La sección de diccionarios permite consultar y añadir entradas a estos, que son las tablas de la base de datos, el código de uno de ellos es el siguiente, los demás se omiten debido a la similitud.

```

private void btnVerDiagActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/notasmedicas","root","");
        String query= "SELECT * FROM DICCIONARIO";
        Statement stm = (Statement)con.createStatement();
        ResultSet rs = stm.executeQuery(query);
        JTable table = new JTable(buildTableModel(rs));
        JOptionPane.showMessageDialog(null, new JScrollPane(table));

    }
    catch (Exception ex)
    {
        lblInfo.setText(ex.toString());
    }
}

```

Para esto se usaron diversos componentes incluidos en el IDE, como JTable y JOptionPane, para mostrar los resultados de las consultas, por lo que la compatibilidad es alta, y no se requiere añadir librerías adicionales.

También se puede añadir a cualquiera de los diccionarios, el código en resumen realiza un insert a la tabla correspondiente:

```

private void btnAgregarDiagActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        String Insertar = txtInsertar.getText().toLowerCase();
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/notasmedicas","root","");
        String query= "INSERT INTO diccionario (ID, Diagnostico) VALUES (NULL, '"+Insertar+
"');";

        Statement stm = (Statement)con.createStatement();
        stm.executeUpdate(query);
        lblInfo.setText("Agregado correctamente");

    }
    catch (Exception ex)
    {
        lblInfo.setText(ex.toString());
    }
}

```

También se puede añadir al diccionario de diagnósticos/síntomas de manera masiva, eligiendo un archivo de texto con cada entrada en un renglón, para esto se utiliza la sección de cargar por archivo, la diferencia es que se realizan múltiples inserciones con un solo clic, haciendo más eficiente el proceso en caso de necesitarse introducir varios.

La parte del análisis es sin duda la más interesante, para ello se hizo lo siguiente:

Se decidió aprovechar las ventajas de la base de datos, y con ayuda de un algoritmo recorrer las cadenas de texto de manera secuencial y realizar operaciones de extracción y comparación varias veces.

Por ello el análisis se hace con las notas cargadas previamente en la base de datos.

Primero se obtiene una palabra y la compara en la lista de síntomas, hasta el caso de existir una coincidencia, esto es que el síntoma esté en la nota original. De esta manera, al finalizar, en varios renglones de una nota, estarán todos los síntomas conocidos que aparezcan en ella. Para las partes del cuerpo, y la intensidad se realiza un proceso análogo. El código explicado es este:

```
private void btnAnalizarActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/notasmedicas","root","");
        //distinct para no obtener repetidos
        String query= "SELECT id, nota FROM notas";
        Statement stm = (Statement)con.createStatement();
        ResultSet rs = stm.executeQuery(query);

        //recorre todos los diagnosticos obtenidos
        while(rs.next())
        {
            int idnota = rs.getInt(1);
            String nota = rs.getString(2);
            int longitudnota = nota.length();
            String buscar = "";
            int tipo = 0;
```

```

boolean palabra = false;
boolean diagnostico = false, lugar = false, intensidad = false;
String DiagnosticoEncontrado = null;
String LugarEncontrado = null;
String IntensidadEncontrado = null;

for(int i = 0; i < longitudnota; i++)
{
    if(Character.isWhitespace(nota.charAt(i)) == false )
    {
        if(palabra)
        {
            buscar = "";
        }
        buscar += nota.charAt(i);
        palabra = false;

    }
    else
    {
        palabra = true;
    }
    if ( i + 1 == longitudnota)
    {
        palabra = true;
    }

    if (palabra)
    { //buscar
        //DIAG
        String queryDiag= "SELECT diagnostico FROM diccionario";
        Statement stmDiag = (Statement)con.createStatement();
        ResultSet rsDiag = stmDiag.executeQuery(queryDiag);
        while (rsDiag.next())
        {
            if (rsDiag.getString(1).equals(buscar) )
            {
                if(diagnostico)
                {
                    //Insertar
                    if(diagnostico && !lugar && !intensidad)
                    {
                        String queryIns= "INSERT INTO analisis (ID_NOTA,
DIAGNOSTICO) VALUES (" + idnota + ", '"+DiagnosticoEncontrado+ "')";
                        Statement stmIns = (Statement)con.createStatement();
                        stmIns.executeUpdate(queryIns);
                    }
                }
            }
        }
    }
}

```

```

    }
    if(diagnostico && lugar && !intensidad)
    {
        String queryIns= "INSERT INTO analisis (ID_NOTA,
        DIAGNOSTICO, LUGAR) VALUES ("+ idnota +",
        '"+DiagnosticoEncontrado+ "', '"+LugarEncontrado+"')";
        Statement stmIns = (Statement)con.createStatement();
        stmIns.executeUpdate(queryIns);
}

    if(diagnostico && !lugar && intensidad)
    {
        String queryIns= "INSERT INTO analisis (ID_NOTA, DIAGNOSTICO,
INTENSIDAD) VALUES ("+ idnota +", '"+DiagnosticoEncontrado+ "', '"+IntensidadEncontrado+"')";
        Statement stmIns = (Statement)con.createStatement();
        stmIns.executeUpdate(queryIns);
    }
    diagnostico = false;
    intensidad = false;
    lugar = false;

    DiagnosticoEncontrado = null;
    LugarEncontrado = null;
    IntensidadEncontrado = null;

}
diagnostico = true;
DiagnosticoEncontrado = buscar;
break;

}
}
////////DIAG
////////LUGAR
if (diagnostico)
{
String queryLugar= "SELECT lugar FROM lugar";
Statement stmLugar = (Statement)con.createStatement();
ResultSet rsLugar = stmLugar.executeQuery(queryLugar);

while (rsLugar.next())
{
    if(rsLugar.getString(1).equals(buscar))
    {
        if(!lugar)
        {

```



```

        LugarEncontrado = buscar;
        lugar = true;
    }
}
} ///////LUGAR
/////INTENSIDAD
if (diagnostico)
{
String queryInt= "SELECT intensidad FROM intensidad";
Statement stmInt = (Statement)con.createStatement();
ResultSet rsInt = stmInt.executeQuery(queryInt);

while (rsInt.next())
{
    if(rsInt.getString(1).equals(buscar))
    {
        if(!intensidad)
        {
            IntensidadEncontrado = buscar;
            intensidad = true;
        }

    }
}
}
/////INTENSIDAD
//Los tres están llenos
if(diagnostico && intensidad && lugar)
{
    String queryIns= "INSERT INTO analisis (ID_NOTA, DIAGNOSTICO, LUGAR,
INTENSIDAD) VALUES (" + idnota + ", '"+DiagnosticoEncontrado+ "', '"+LugarEncontrado+"',
'"+IntensidadEncontrado+"' );";
    Statement stmIns = (Statement)con.createStatement();
    stmIns.executeUpdate(queryIns);
    diagnostico = false;
    intensidad = false;
    lugar = false;
    DiagnosticoEncontrado = null;
    LugarEncontrado = null;
    IntensidadEncontrado = null;
}
//FINAL NOTA
if ( i + 1 == longitudnota)
{
    if(diagnostico)

```

```

        {
            //Insertar
            if(diagnostico && !lugar && !intensidad)
            {
                String queryIns= "INSERT INTO analisis (ID_NOTA, DIAGNOSTICO)
VALUES (" + idnota + ", '"+DiagnosticoEncontrado+ "');";
                Statement stmIns = (Statement)con.createStatement();
                stmIns.executeUpdate(queryIns);
            }

            if(diagnostico && lugar && !intensidad)
            {
                String queryIns= "INSERT INTO analisis (ID_NOTA, DIAGNOSTICO,
LUGAR) VALUES (" + idnota + ", '"+DiagnosticoEncontrado+ "', '"+LugarEncontrado+"');";
                Statement stmIns = (Statement)con.createStatement();
                stmIns.executeUpdate(queryIns);
            }

            if(diagnostico && !lugar && intensidad)
            {
                String queryIns= "INSERT INTO analisis (ID_NOTA, DIAGNOSTICO,
INTENSIDAD) VALUES (" + idnota + ", '"+DiagnosticoEncontrado+ "', '"+IntensidadEncontrado+"');";
                Statement stmIns = (Statement)con.createStatement();
                stmIns.executeUpdate(queryIns);
            }
            }
        diagnostico = false;
        intensidad = false;
        lugar = false;

        DiagnosticoEncontrado = null;
        LugarEncontrado = null;
        IntensidadEncontrado = null;
    }
} // PALABRA
}
}
lblInformacion.setText("Análisis completo.");
// TODO add your handling code here:
} catch (ClassNotFoundException ex) {
    Logger.getLogger(interfaz.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(interfaz.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

} Finalmente se puede consultar la base de datos que contiene las notas analizadas desde la aplicación, para mayor comodidad.