

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Sistemas Inteligentes

Clase

Profa.: Raquel Martínez

Aula: 4206

L-M-V – V2

Laboratorio

Prof.: Luis Vega

Aula: Inf B

J – N2

**Manual de usuario**

Índice

[Portada](#Portada) …………………………………………………………………………………………………………………………………………………….. 1

[Índice](#Indice) ……………………………………………………………………………………………………………………………………………………….. 2

[Introducción](#Intro) …………………………………………………………………………………………………………………………………………….. 3

[Instalación](#Installation) ……………………………………………………………………………………………………………………………………………….. 4

[Uso](#Use) ………………………………………………………………………………………………………………………………………………………….. 5

[Terceras partes](#ThirdParties) ………………………………………………………………………………………………………………………………………… 11

[Desarrolladores](#Devs) .………………………………………………………………………………………………………………………………………. 12

[Errores conocidos](#Errores) …………………………………………………………………………………………………………………………………….. 13

Introducción

A través de este proyecto buscamos analizar comentarios desde una página e alojamiento para determinar las malas palabras que comenta la gente.

Hemos establecido 3 niveles de agresividad.

Nivel 1 – Palabras groseras pero de uso cotidiano.

Nivel 2 – Palabras groseras con un uso más fuerte.

Nivel 3 – Palabras que dirigen hacia designaciones raciales o sexuales.

El análisis se divide en 2 secciones

1.- Análisis por diccionario

En este tipo de análisis comparamos cada palabra extraída de la base de datos contra cada palabra de cada diccionario.

Esto si bien es un análisis simple el cual siempre se obtiene un mismo resultado ya que no hay variación en el análisis debido a que no se modifican las letras ni diferentes modos sintácticos y gramaticales lo cual hace que no se detecten palabras que en realidad si merecen ser detectadas.

Un ejemplo de ello es el siguiente:

En el diccionario nivel 1 tenemos la palabra tonta, sin embargo desde la base de datos tenemos que la cadena adquirida es tonta., como se puede ver tonta no es igual a tonta. Por lo cual no se detectará y pasará desapercibida por el algoritmo.

2.- Análisis por red neuronal

Como se apreció en el ejemplo anterior, es necesario ejecutar diversas acciones extras en el análisis para poder detectar un mayor campo de palabras y no dejar pasar palabras que necesiten estar presentes en la selección.

Los pasos extras que definimos son:

* Análisis de longitud de palabra.
* Modificaciones gramaticales.
* Correcciones ortográficas.
* Intercalado de letras por números.
* Paso por diccionarios de nuevo.

Instalación

El proyecto requiere de una base de datos, la librería FANN y otros archivos para funcionar, todos los paquetes se anexan en el proyecto.

Instalación del proyecto

Una vez descargado el archivo ZIP proceda con los siguientes procedimientos

1.- Descomprima el ZIP.

2.- Encontrará la carpeta [INSERTE NOMBRE CARPETA].

3.- Al acceder a ella encontrará el archivo [INSERTE NOMBRE ARCHIVO].jar, ejecuelo haciendo doble click sobre él.

Instalación de la base de datos

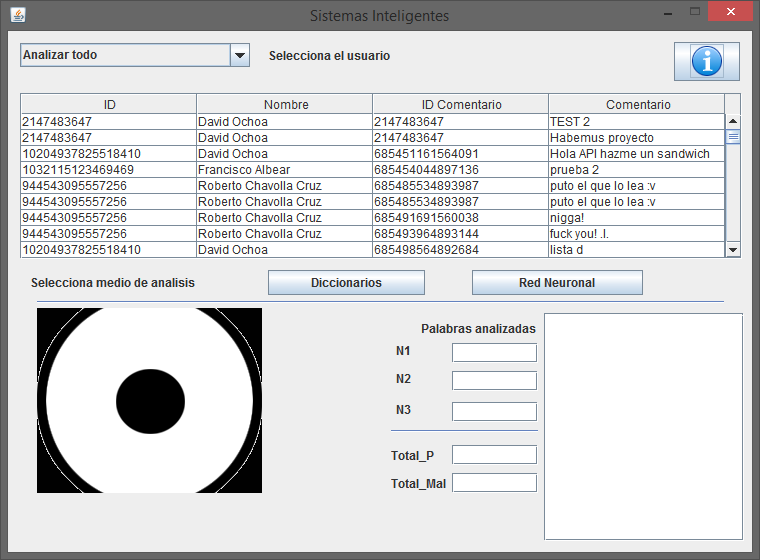
Primeramente se necesita montar la base de datos en un servidor local, ya sea que use WampServer, XAMPP o cualquier otro. Se requiere que el nombre de usuario sea: root y que no requiera contraseña.

Username: root

Password: [Sin Password – Dejar en blanco]

Uso

Interfaz principal



1.- Selector de usuario a analizar o analizar todos.

2.- Tabla que muestra los comentarios por el usuario seleccionado.

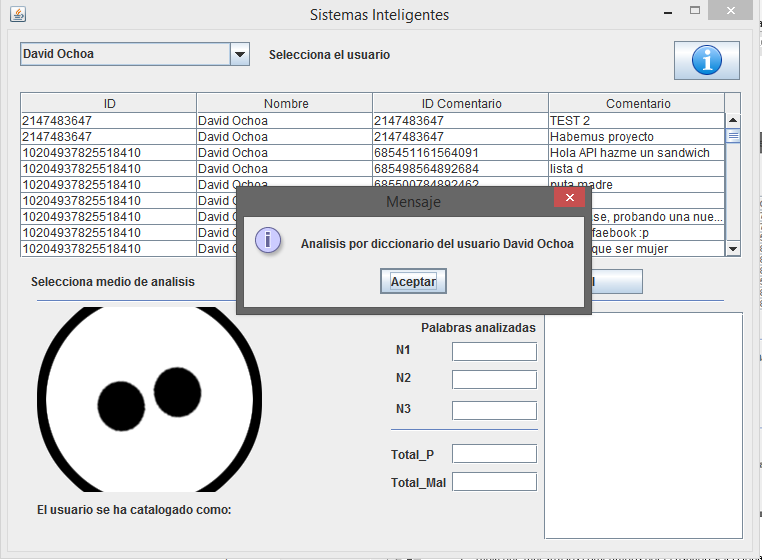
3.- Botón selector de análisis por diccionarios.

4.- Botón selector de análisis por red neuronal y análisis más avanzados.

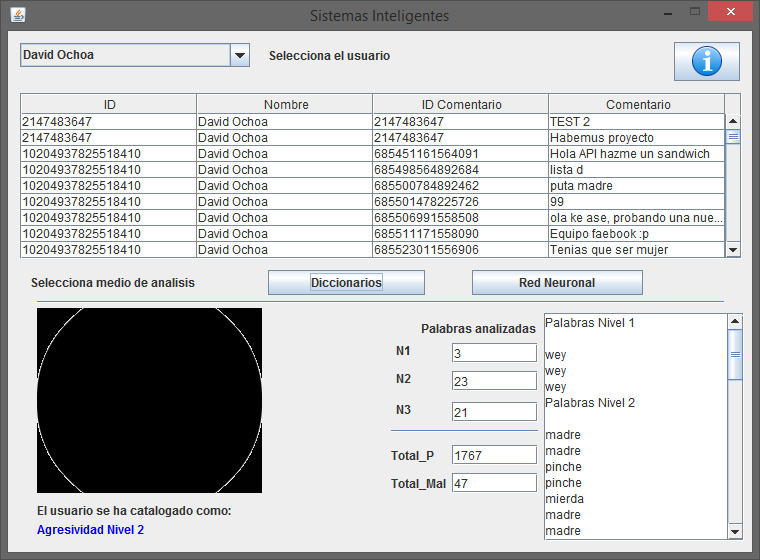
5.- Campo donde se muestran las palabras analizadas y detectadas como positivas al algoritmo.

6.- Contadores de palabras totales, groseras y detectadas por nivel.

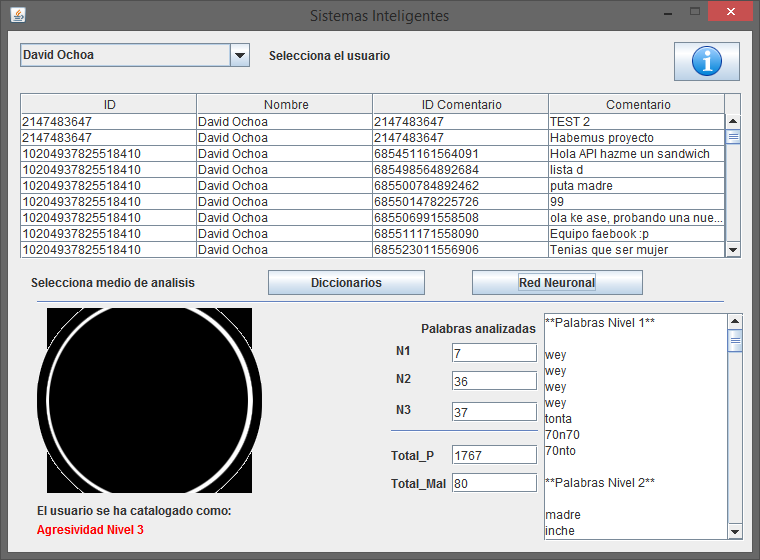
7.- Botón acerca de.



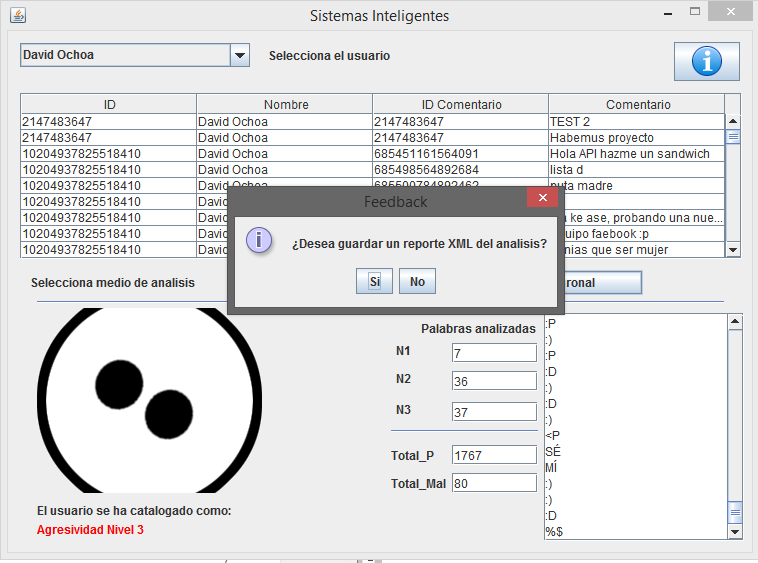
Una vez seleccionado el usuario que de desee analizar, se mostrará una ventana emergente que le mostrará que tipo de análisis escogió y sobre que usuario trabajará.



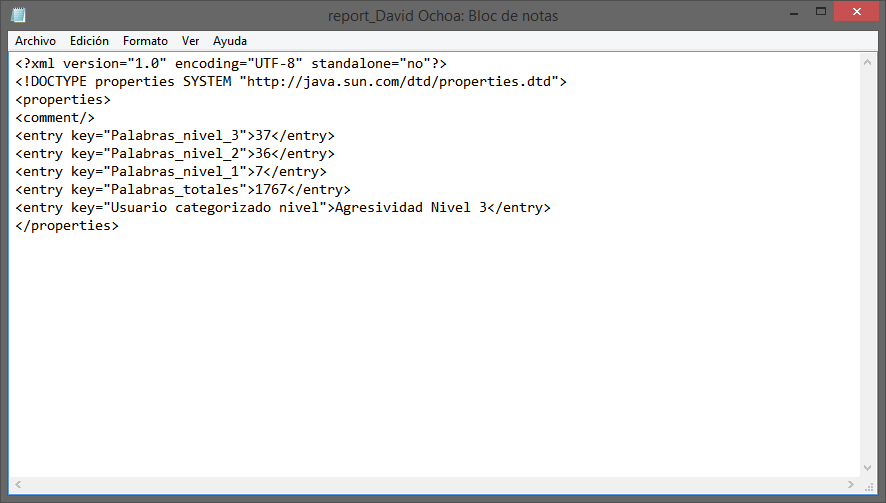
Una vez finalizado el análisis se mostrarán los resultados en sus respectivos campos.



Comparación de los resultados usando ahora el análisis por red neuronal, se puede apreciar un resultado más prometedor.



Cada que finaliza el proceso de análisis por red neuronal se le preguntará si desea guardar un archivo de reporte en formato XML donde podrá guardar los resultados sin tener abierto el sistema.



Terceras partes

El proyecto necesita de las siguientes librerías para un correcto funcionamiento

Fast Artificial Neural Network (FANN) Library

http://leenissen.dk/fann/wp/

(Steffen Nissen, s.f.)

FANN Java Bind for FANN

http://code.google.com/p/fannj/

https://github.com/krenfro/fannj

Kyle Renfro

Java Native Access (JNA) Library

https://github.com/twall/jna

Oracle – Daniel Doubrovkine

Basic Player

http://www.javazoom.net/jlgui/api.html

Desarrolladores

Este proyecto ha sido desarrollado por:

David Ochoa Gutierrez

Gustavo Dávila Treviño

Víctor Manuel Moreno Martínez

Diana Anaid Loza Cerda

Francisco Albear Cárdenas

Errores conocidos

Hemos detectado un error ajeno a nuestro código, el problema es con relación a los archivos FANN

CASO 1.- Ventana de Windows alertando que “Java se ha detenido por un problema”

Posible solución: Dirigirse a la carpeta FANN/bin y dar click derecho sobre el archivo fannfloat.dll, seleccionar propiedades y observar si en la parte inferior de la ventana advierte sobre bloqueo de archivo por fuentes desconocidas, dar click en “Desbloquear” y posteriormente aplicar y aceptar, intente de nuevo arrancar el sistema.

CASO 2.- Ventana de Windows alertando que “Java se ha detenido por un problema”

Posible solución: Cierre y vuelva a arrancar el sistema, si presenta el mismo error, verifique los pasos del caso 1

