



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 técnico**



# Índice

<a href="#">Portada</a> .....	1
<a href="#">Índice</a> .....	2
<a href="#">Introducción</a> .....	3
<a href="#">Base de datos</a> .....	4
<a href="#">Interface de entrada de datos</a> .....	5
<a href="#">Interface de procesamiento de datos</a> .....	6
<a href="#">Terceras partes</a> .....	16
<a href="#">Desarrolladores</a> .....	17
<a href="#">Enlaces</a> .....	18
<a href="#">Errores conocidos</a> .....	19



# 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.

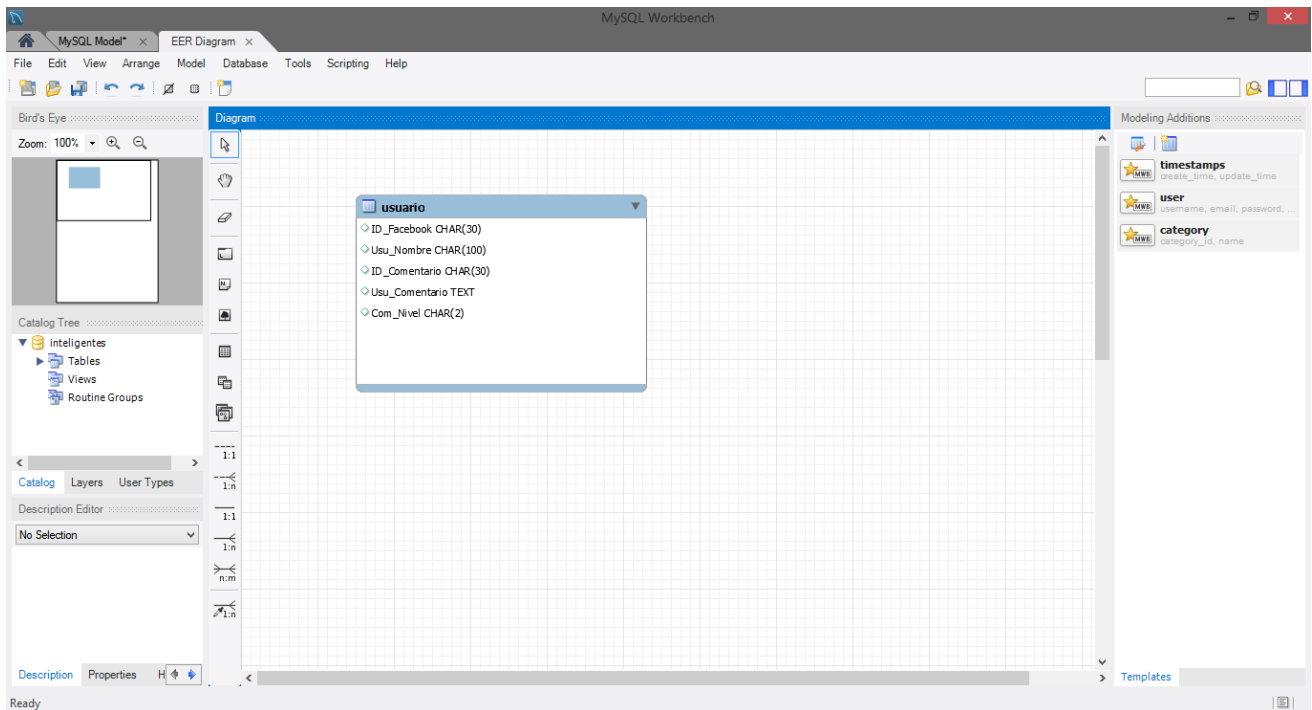


# Base de datos

La base de datos está creada en MySQL [inteligentes], consta de una tabla [usuario] la cual se divide en los siguientes campos:

- ID\_Facebook – Almacén del ID de usuario de Facebook
- Usu\_Nombre – Almacén del nombre de usuario
- ID\_Comentario – Almacén del ID del comentario hecho
- Usu\_Comentario – Almacén del comentario hecho

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1	ID_Facebook	char(30)	utf8_spanish_ci		Sí	NULL
<input type="checkbox"/>	2	Usu_Nombre	char(100)	utf8_spanish_ci		Sí	NULL
<input type="checkbox"/>	3	ID_Comentario	char(30)	utf8_spanish_ci		Sí	NULL
<input type="checkbox"/>	4	Usu_Comentario	text	utf8_spanish_ci		Sí	NULL



# Interface de entrada de datos

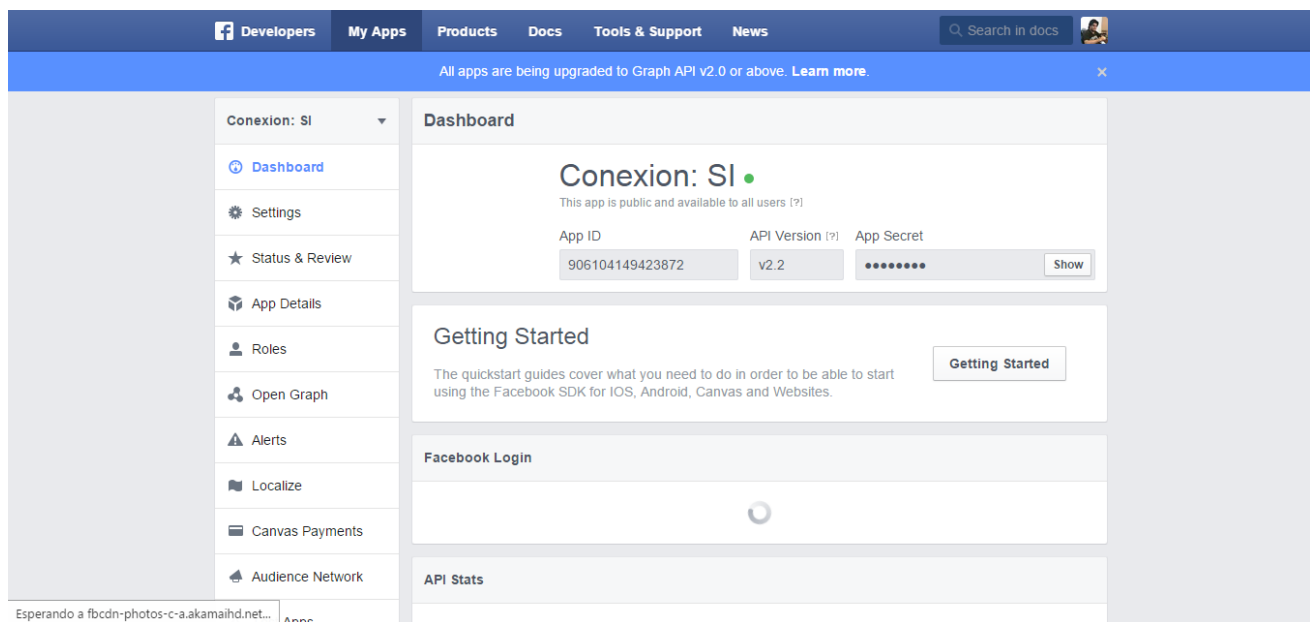
La alimentación de la base de datos proviene desde una página web que incluye una caja de comentarios de Facebook, del cual por medio de la API tenemos acceso al nombre e ID del usuario que comentó y por ende el ID y comentario hecho.

Esto es posible por medio de 2 partes:

- Parte Facebook
- Parte web

## *Parte Facebook*

Para poder obtener acceso a la API, Facebook requiere que nos demos de alta en su sitio para desarrolladores ([www.developers.facebook.com](http://www.developers.facebook.com)) en la cual tendremos que crear una aplicación de Facebook para tener acceso a la API. Una Aplicación de Facebook no es más que un proyecto que nos genera las llaves o tokens y números de identidad únicos a los desarrolladores para tener acceso a la API y poder crear aplicaciones o programas con acceso a Facebook, como por ejemplo: agregar el botón “Me Gusta” en nuestro sitio o aplicación móvil.



## Parte Web

Se creó un sitio de recepción de comentarios, alojado en ([www.theradikalsoftware.tk/sistin](http://www.theradikalsoftware.tk/sistin)) donde se hace uso de una caja de comentarios que detecta cuando se ejecuta dicha acción y la procesa por medio de JavaScript y PHP para poderla almacenar en la base de datos descrita anteriormente.

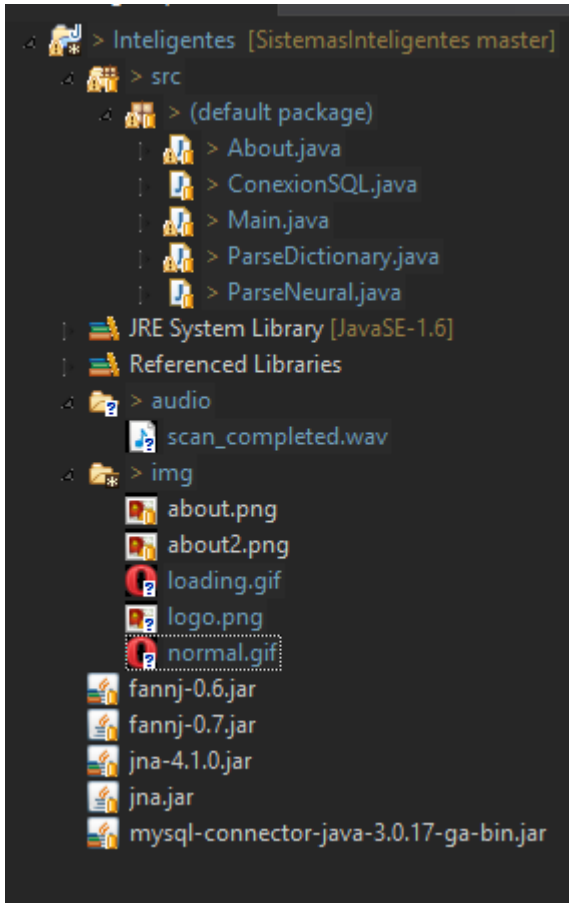
El sitio está construido en los siguientes lenguajes HTML5, JavaScript, PHP, JQuery.



# Interface de procesamiento de datos

Se creó un sistema en el lenguaje Java que hace uso de la base de datos para alimentarse con los comentarios, para después procesarlos por medio de un diccionario de datos y de una red neuronal.

La jerarquización del proyecto se muestra en la siguiente imagen.



## Definición de clases

**About** – Ventana donde se puede ver a los integrantes.

**ConexionSQL** – Clase que se encarga de operar las conexiones a la base de datos.

**Main** – Es la ventana principal donde se podrá interactuar con el programa.

**ParseDictionary** – Clase encargada de recolectar y analizar los comentarios por medio de los diccionarios.

**ParseNeural** – Clase encargada de recolectar, modificar sintácticamente, gramaticalmente, ortográficamente y morfológicamente los datos para proceder a analizarlos por medio de una red neuronal

**Carpeta img** – Contiene las imágenes usadas en las interfaces

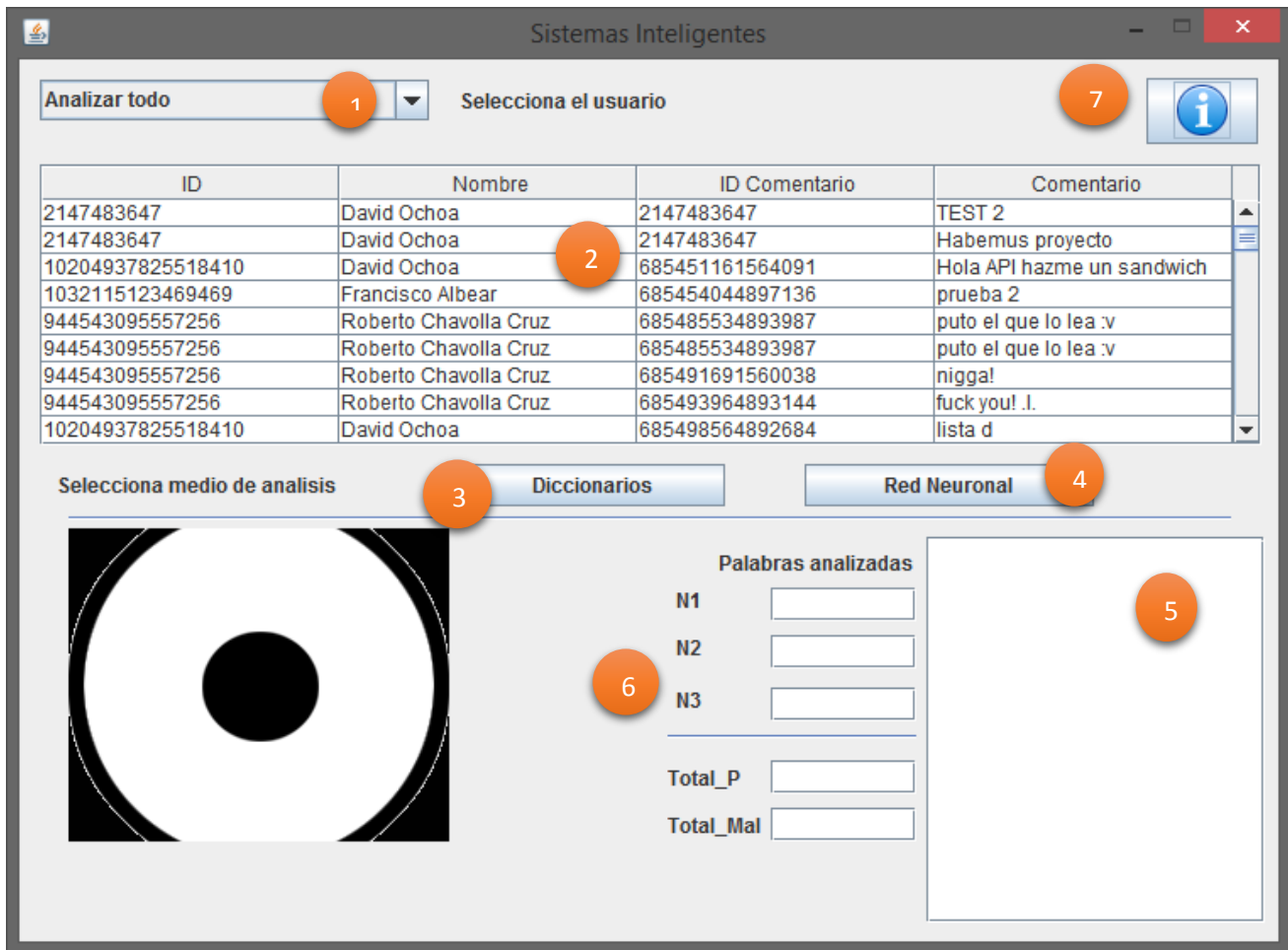
**Carpeta audio** – Contiene sonidos que se usan en la interfaz del sistema

**fannj-XX.jar** – Librería de la red neuronal FANN modificada para Java

**jna-4.1.0.jar** – Librería de acceso nativo a librerías DLL

**mysql-connector.jar** – Librería usada para acceder a la base de datos MySQL

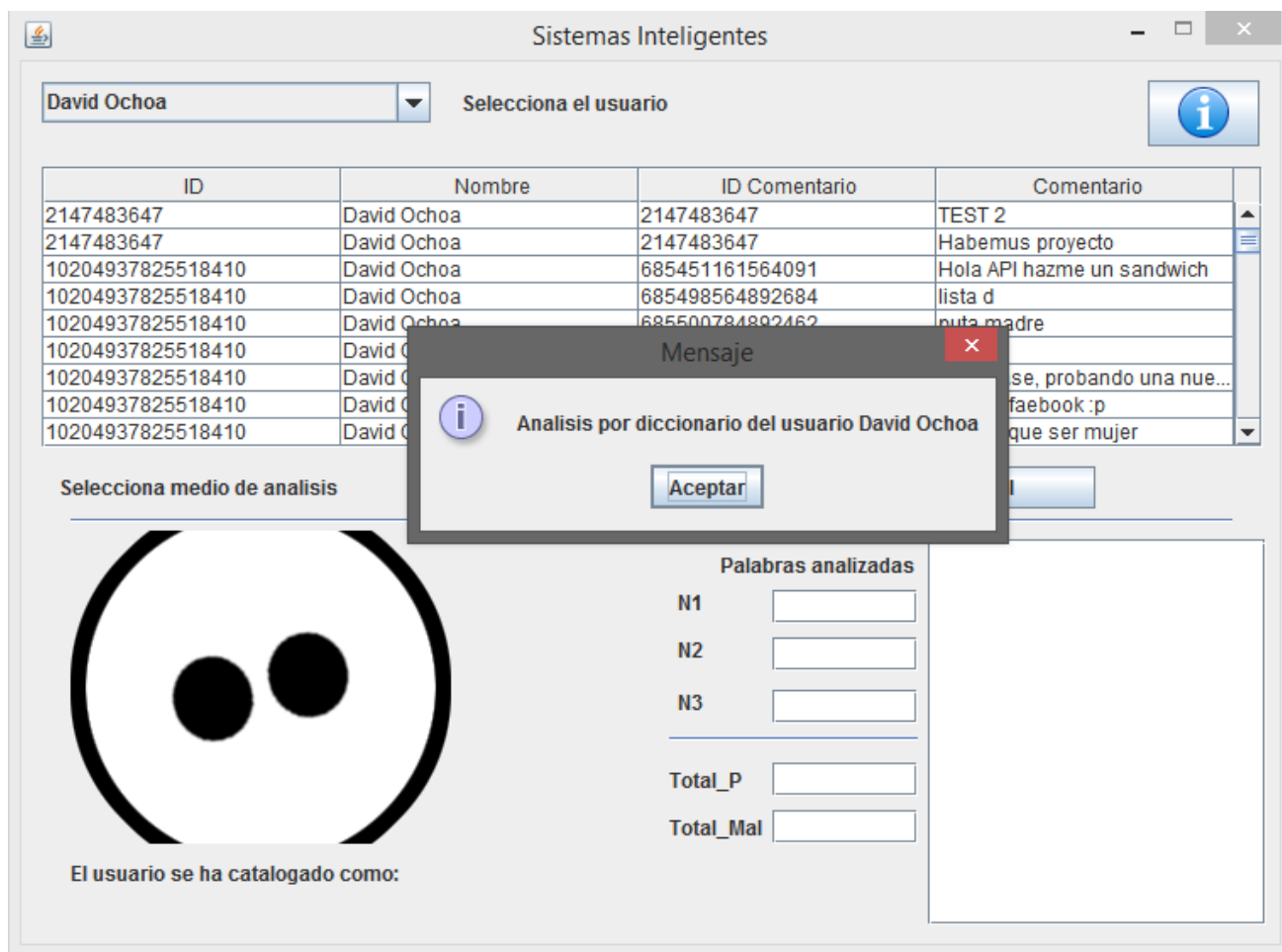




- 1.- Selector de usuario a analizar o analizar todos.
- 2.- Tabla que muestra los comentarios por el usuario seleccionado.
- 3.- Botón selector de análisis por diccionarios.
- 4.- Botón selector de análisis por red neuronal y análisis más avanzados.
- 5.- Campo donde se muestran las palabras analizadas 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.



Sistemas Inteligentes

David Ochoa

Selecciona el usuario

ID	Nombre	ID Comentario	Comentario
2147483647	David Ochoa	2147483647	TEST 2
2147483647	David Ochoa	2147483647	Habemus proyecto
10204937825518410	David Ochoa	685451161564091	Hola API hazme un sandwich
10204937825518410	David Ochoa	685498564892684	lista d
10204937825518410	David Ochoa	685500784892462	puta madre
10204937825518410	David Ochoa	685501478225726	99
10204937825518410	David Ochoa	685506991558508	ola ke ase, probando una nue...
10204937825518410	David Ochoa	685511171558090	Equipo faebook :p
10204937825518410	David Ochoa	685523011556906	Tenias que ser mujer

Selecciona medio de analisis

Diccionarios

Red Neuronal

Palabras analizadas

N1

3

N2

23

N3

21

Total\_P

1767

Total\_Mal

47

Palabras Nivel 1

wey

wey

wey

Palabras Nivel 2

madre

madre

pinche

pinche

mierda

madre

madre

El usuario se ha catalogado como:

Agresividad Nivel 2

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



Sistemas Inteligentes

David Ochoa

Selecciona el usuario

ID	Nombre	ID Comentario	Comentario
2147483647	David Ochoa	2147483647	TEST 2
2147483647	David Ochoa	2147483647	Habemus proyecto
10204937825518410	David Ochoa	685451161564091	Hola API hazme un sandwich
10204937825518410	David Ochoa	685498564892684	lista d
10204937825518410	David Ochoa	685500784892462	puta madre
10204937825518410	David Ochoa	685501478225726	99
10204937825518410	David Ochoa	685506991558508	ola ke ase, probando una nue...
10204937825518410	David Ochoa	685511171558090	Equipo faebook :p
10204937825518410	David Ochoa	685523011556906	Tenias que ser mujer

Selecciona medio de analisis

Diccionarios

Red Neuronal

El usuario se ha catalogado como:

Agresividad Nivel 3

Palabras analizadas

N1

7

N2

36

N3

37

Total\_P

1767

Total\_Mal

80

\*\*Palabras Nivel 1\*\*

wey

wey

wey

wey

tonta

70n70

70nto

\*\*Palabras Nivel 2\*\*

madre

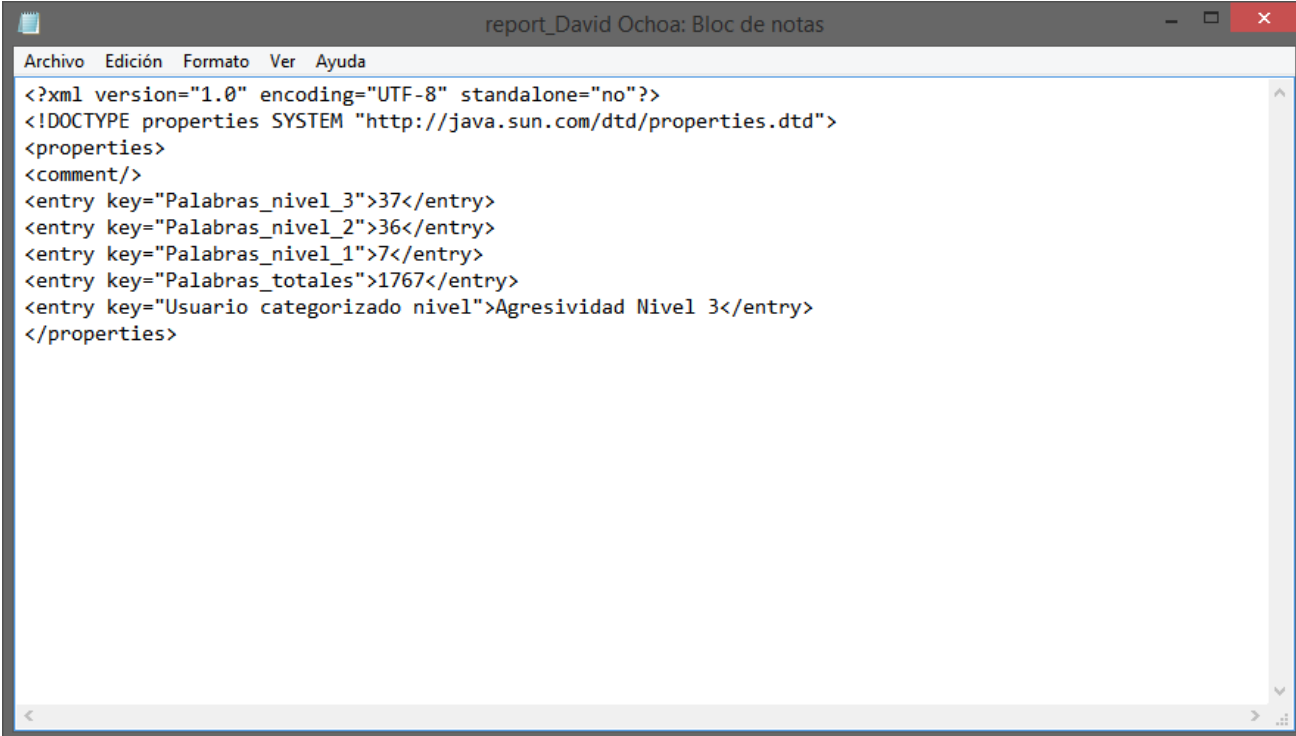
inche

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.



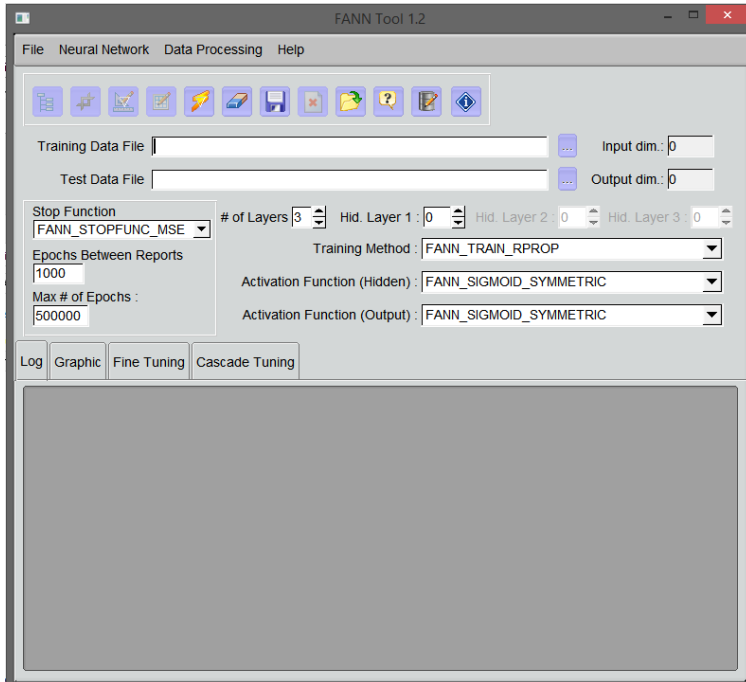
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment/>
<entry key="Palabras_nivel_3">37</entry>
<entry key="Palabras_nivel_2">36</entry>
<entry key="Palabras_nivel_1">7</entry>
<entry key="Palabras_totales">1767</entry>
<entry key="Usuario categorizado nivel">Agresividad Nivel 3</entry>
</properties>
```



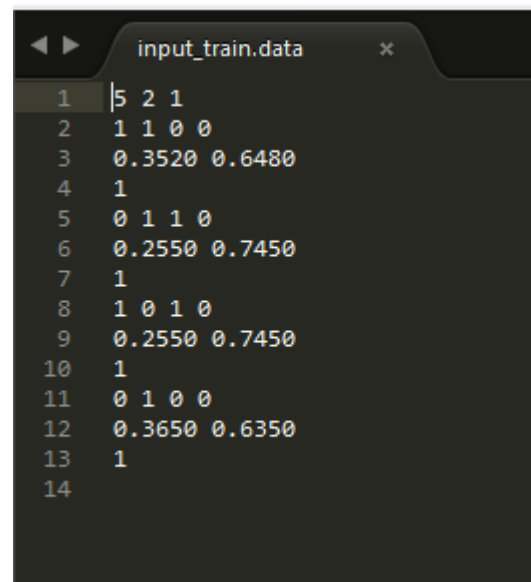
# Red neuronal

Se usó la librería FANN (Fast Artificial Neural Network) para este proyecto, FANN es una librería escrita en C por lo cual se necesita FANNJ que es un bind que permite usar las redes neuronales FANN.

Al no ser nativo, no podemos entrenar la red mientras se ejecuta el código por lo cual se usó la herramienta FANNTOOL, que crea la red neuronal por medio de un archivo de entrenamiento y crea un archivo .NET el cual es el que se introduce en el código java para su utilización.



Por ende, si queremos actualizar un nuevo archivo .NET, producto de un nuevo entrenamiento, es necesario ensamblar el código de nuevo ya que está el archivo se encuentra dentro del paquete compilado del proyecto.



Ejemplos del archivo de entrenamiento (.DATA) y del archivo de configuración (.NET).



```
ANNN.net
1 FANN_FLO_2.1
2 num_layers=3
3 learning_rate=0.700000
4 connection_rate=1.000000
5 network_type=0
6 learning_momentum=0.000000
7 training_algorithm=2
8 train_error_function=0
9 train_stop_function=0
10 cascade_output_change_fraction=0.010000
11 quickprop_decay=-0.000100
12 quickprop_mu=1.750000
13 rprop_increased_factor=1.200000
14 rprop_decrease_factor=0.500000
15 rprop_delta_min=0.000000
16 rprop_delta_max=50.000000
17 rprop_delta_zero=0.100000
18 cascade_output_stagnation_epochs=12
19 cascade_candidate_change_fraction=0.010000
20 cascade_candidate_stagnation_epochs=12
21 cascade_max_out_epochs=150
22 cascade_min_out_epochs=50
23 cascade_max_cand_epochs=150
24 cascade_min_cand_epochs=50
25 cascade_num_candidate_groups=2
26 bit_fail_limit=3.49999994039535520000e-001
27 cascade_candidate_limit=1.000000000000000000e+003
28 cascade_weight_multiplier=4.00000005960464480000e-001
29 cascade_activation_functions_count=10
30 cascade_activation_functions=3 5 7 8 10 11 14 15 16 17
31 cascade_activation_steepnesses_count=4
32 cascade_activation_steepnesses=2.500000000000000000e-001 5.000000000000000000e-001 7.500000000000000000e-001 1.000000000000000000e+000
33 layer_sizes=3 2 2
34 scale_included=0
35 neurons (num_inputs, activation_function, activation_steepness)=(0, 0, 0.000000000000000000e+000) (0, 0, 0.000000000000000000e+000) (0, 0, 0.000000000000000000e+000) (3, 7, 5.000000000000000000e-001) (0, 7, 5.000000000000000000e-001) (2, 14, 5.000000000000000000e-001) (0, 14, 5.000000000000000000e-001)
36 connections (connected_to_neuron, weight)=(0, 7.72181272506713870000e+000) (1, 1.07322525978088380000e+000) (2, -4.04059934616088870000e+000) (3, 4.48162841796875000000e+001) (4, -1.56426340341567990000e-001)
37
```



# 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>

Se usó el siguiente software para la realización del mismo

Eclipse Java IDE ([www.eclipse.org](http://www.eclipse.org))

WampServer ([www.wampserver.es](http://www.wampserver.es))

MySQL ([www.mysql.com](http://www.mysql.com))

FANNTOL ([www.fanntool.blogspot.com](http://www.fanntool.blogspot.com))

Sublime Text ([www.sublimetext.com](http://www.sublimetext.com))

Windows 8.1 ([www.windows.microsoft.com](http://www.windows.microsoft.com))

Fedora Linux ([www.fedoraproject.org](http://www.fedoraproject.org))

Java 8 ([www.java.com](http://www.java.com))

Se usó el siguiente servicio para la realización del mismo

DigitalOcean ([www.digitalocean.com](http://www.digitalocean.com))

GitHub ([www.github.com](http://www.github.com))

Facebook ([www.facebook.com](http://www.facebook.com))





# Desarrolladores

Este proyecto ha sido desarrollado por:

David Ochoa Gutierrez

Gustavo Dávila Treviño

Víctor Manuel Moreno Martínez

Diana Anaid Loza Cerda

Francisco Albear Cárdenas



# Enlaces

Códigos

[www.github.com/TheRadikalStyle/SistemasInteligentes](https://www.github.com/TheRadikalStyle/SistemasInteligentes)

Página WEB

[www.theradikalsoftware.tj/sistin](http://www.theradikalsoftware.tj/sistin)

Videos de funcionamiento

[www.youtube.com/user/TheRadikalStyle](https://www.youtube.com/user/TheRadikalStyle)



# 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

