

# Descubrimiento de Información en Textos

## Tarea del Tema 5: Aplicación creada y utilizada para la tarea.

**Jose Alberto Benítez Andrades**

**71454586A**

**Descubrimiento de Información en Textos**

**Máster en Lenguajes y Sistemas Informáticos - Tecnologías del Lenguaje en la Web**

**UNED**

**29/03/2011**

29 de marzo de 2011

## 1.Introducción

La aplicación utilizada para la realización de la tarea, ha sido creada por mí completamente, en lenguaje JAVA y bajo el IDE NetBeans 6.9.1.

Es una aplicación sencilla, contiene 2 pestañas en las que se muestran toda la info en distintos JTextAreas:

- En esta primera parte, mediante el botón añadir fichero, se agregan los distintos ficheros que contienen el texto del cual realizaremos el análisis léxico, la eliminación de stop-words, el truncado y el cálculo de pesos parciales y globales mediante los distintos algoritmos.

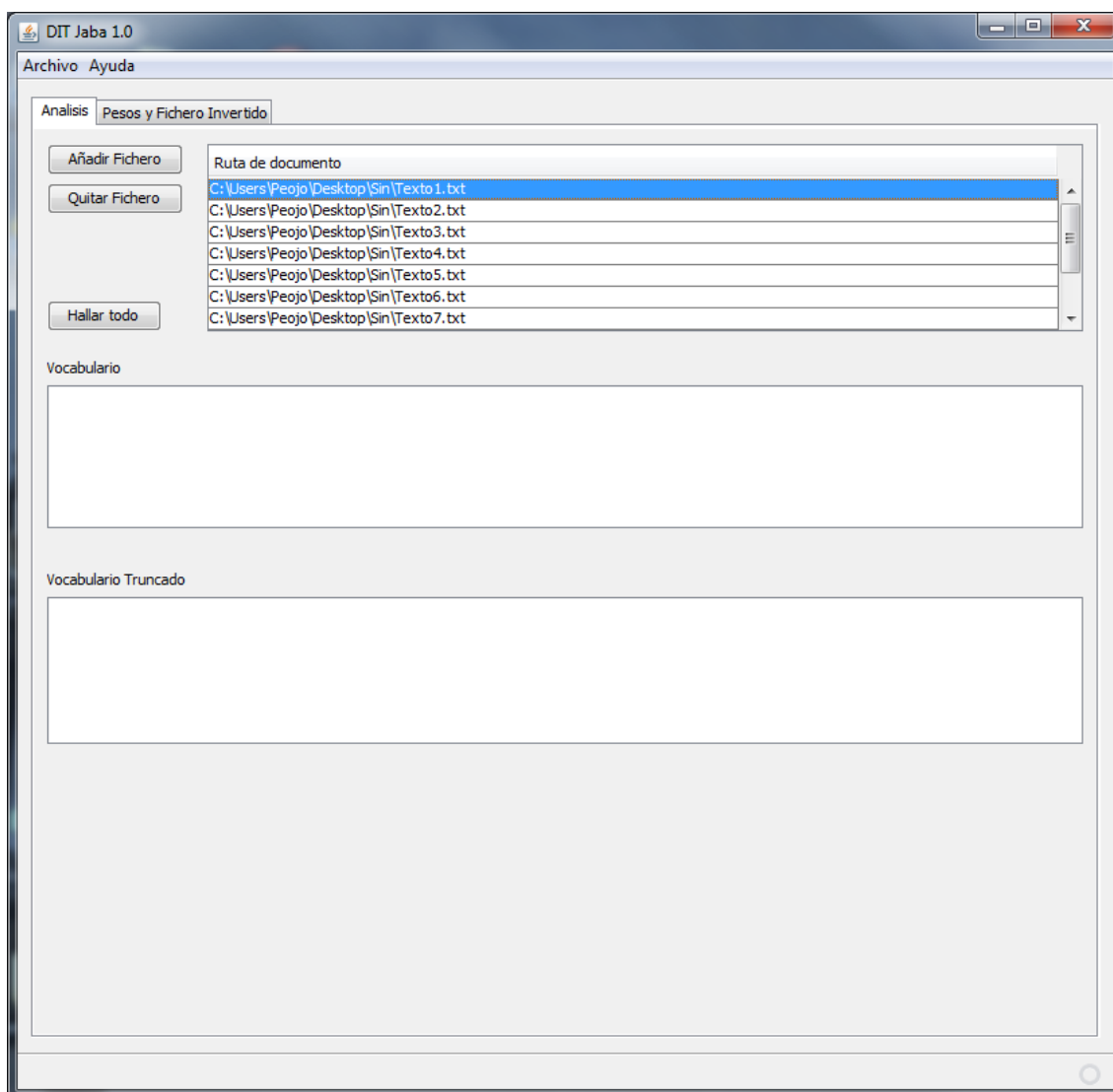


Figura 1: Pantalla inicial del programa

- En esta parte veremos los resultados de los pesos globales y los pesos parciales del texto analizado.

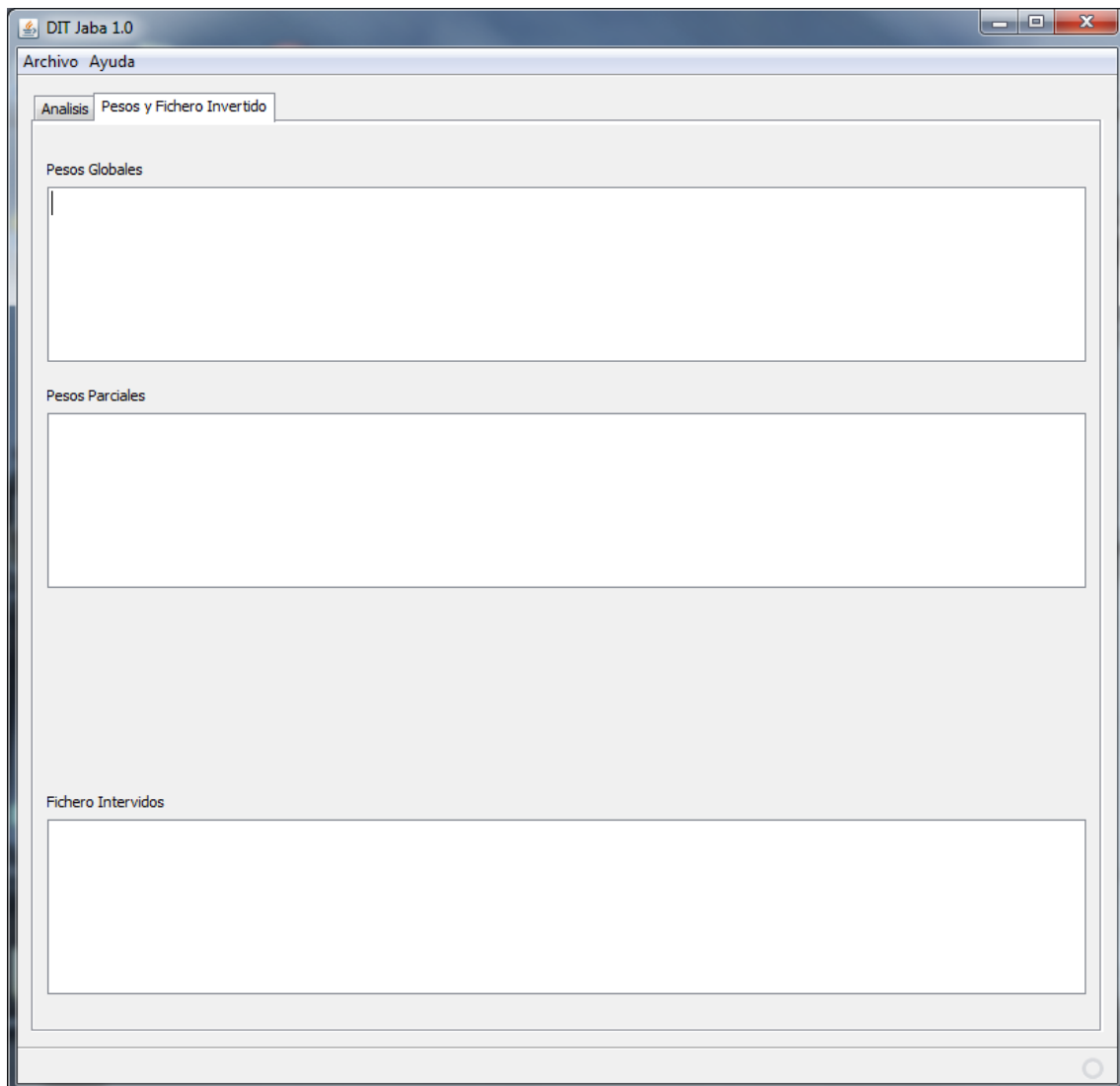


Figura 2: Pestaña Pesos y Fichero Invertido

29 de marzo de 2011

## 2.Introducción

El funcionamiento es muy sencillo, después de insertar todos los ficheros que queremos analizar, pulsamos sobre el botón "Hallar todo" y obtendremos toda la información que queremos sobre el texto que hay en esos ficheros.

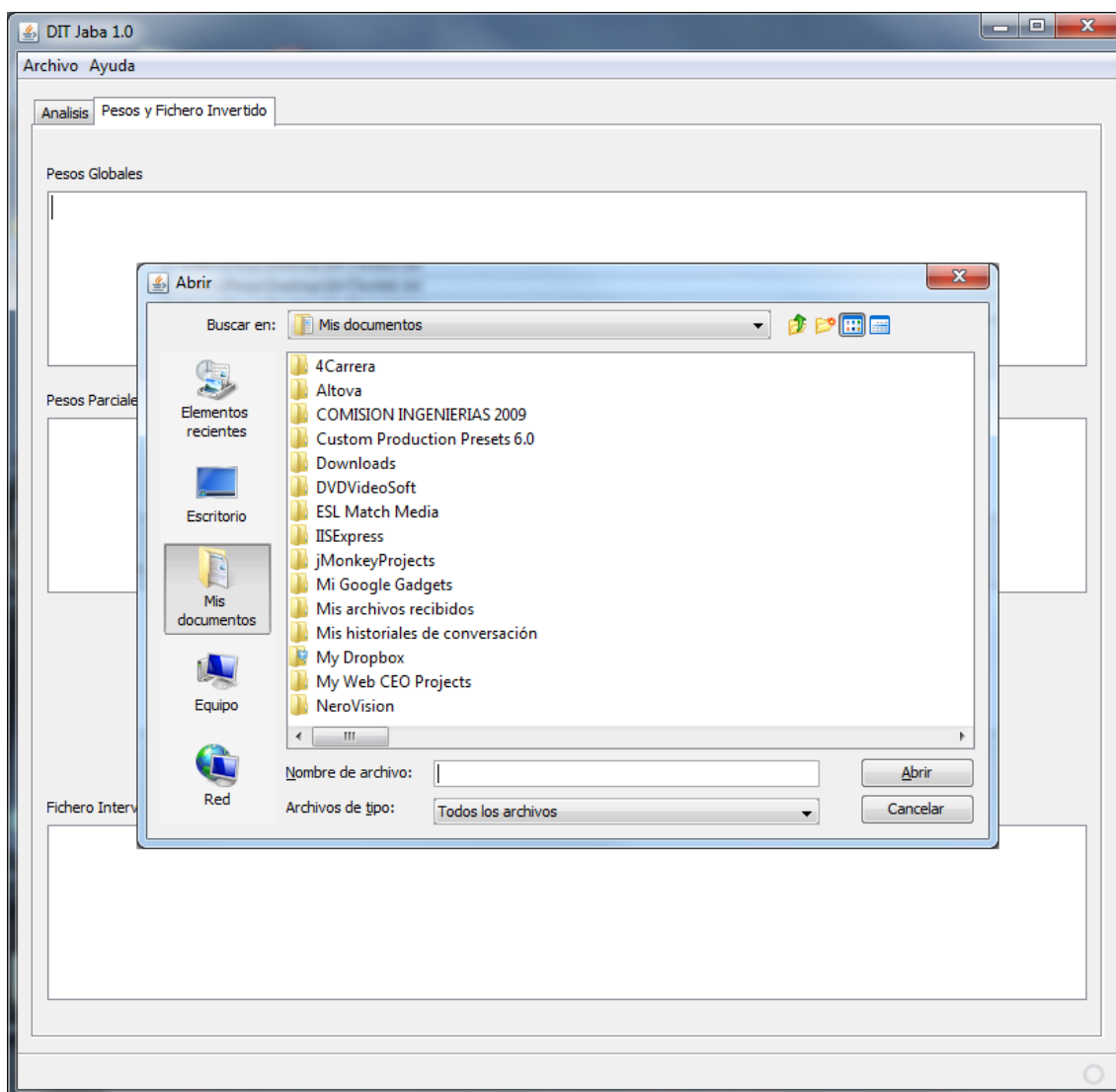


Figura 3: Diálogo que se abre al clickar el botón "Añadir Fichero"

29 de marzo de 2011

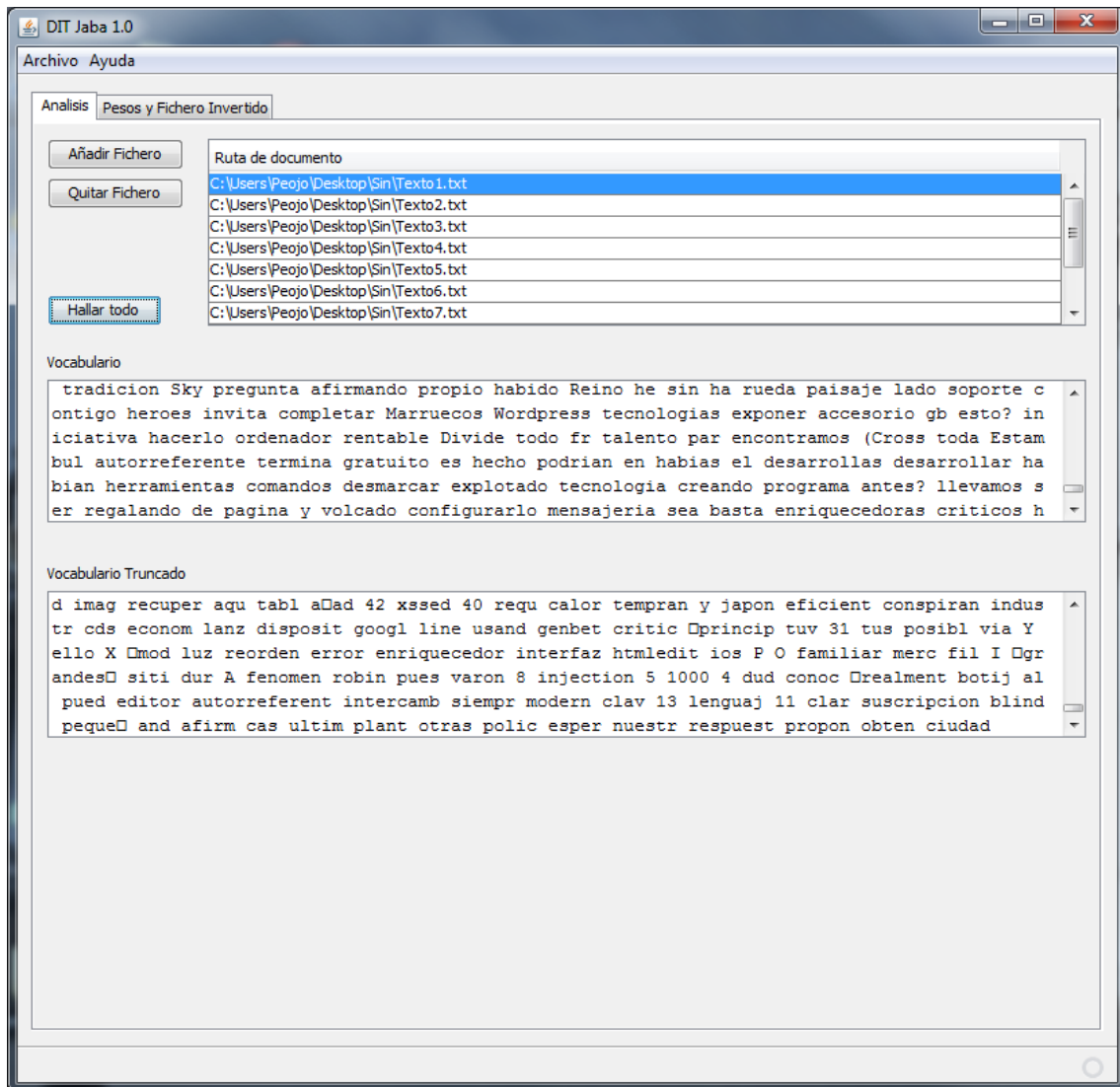


Figura 4: Pestaña de Análisis que muestra el vocabulario obtenido de los ficheros y el vocabulario truncado.

- Al pulsar sobre el botón "Hallar Todo", se ejecuta una función que realiza todos los cálculos necesarios, su funcionamiento es el siguiente:

1. Crea tres tablas Hash, que van a almacenar los siguientes datos:

- Hashtable[] hashParcial : Array de tablas hash, formado por tantas tablas hash, como ficheros ha insertado el usuario. Contendrán los términos del vocabulario truncado junto con el número de veces que aparece en cada documento.
- Hashtable hashGlobal : Tabla hash que almacena todos los términos del vocabulario truncado (es decir, una vez se han eliminado las stop-words y además ha pasado por el stemmer)
- Hashtable hashOriginal : Tabla hash que almacena todos los términos del vocabulario, sin haber sido truncados ni eliminadas las stop-words.

29 de marzo de 2011

2. El siguiente paso es recorrer con un bucle for, todos los ficheros agregados por el usuario y realizar las siguientes funciones:

- Inicializar la tabla hash de hashParcial[i] donde i es el número de fichero que está leyendo en esa iteración.
- Comienza a leer el fichero, obteniendo el nombre del JTable que está en la aplicación.
- Comprueba, mediante un if, si el carácter que lee es : [ ", ' , ' , ' , ' , '\t' , '\n' ]. En caso negativo, va creando una cadena y cuando acaba de crearla (cuando se encuentra con uno de los caracteres anteriormente expuestos) almacena la palabra en las tablas hash que corresponden (hashOriginal y hashParcial[i]) y el número de repeticiones.
- Después de realizar el paso anterior, comprueba con cada palabra si se encuentra entre una palabra de tipo "stop-word", en caso negativo, se trunca la palabra mediante la clase Stemmer que contiene un algoritmo de truncado específico mediante el cual se eliminan distintas terminaciones de palabras o verbos siguiendo unas reglas. Después de truncar la palabra, se almacena en la tabla hash hashGlobal con su respectivo número de apariciones a lo largo de todos los ficheros.

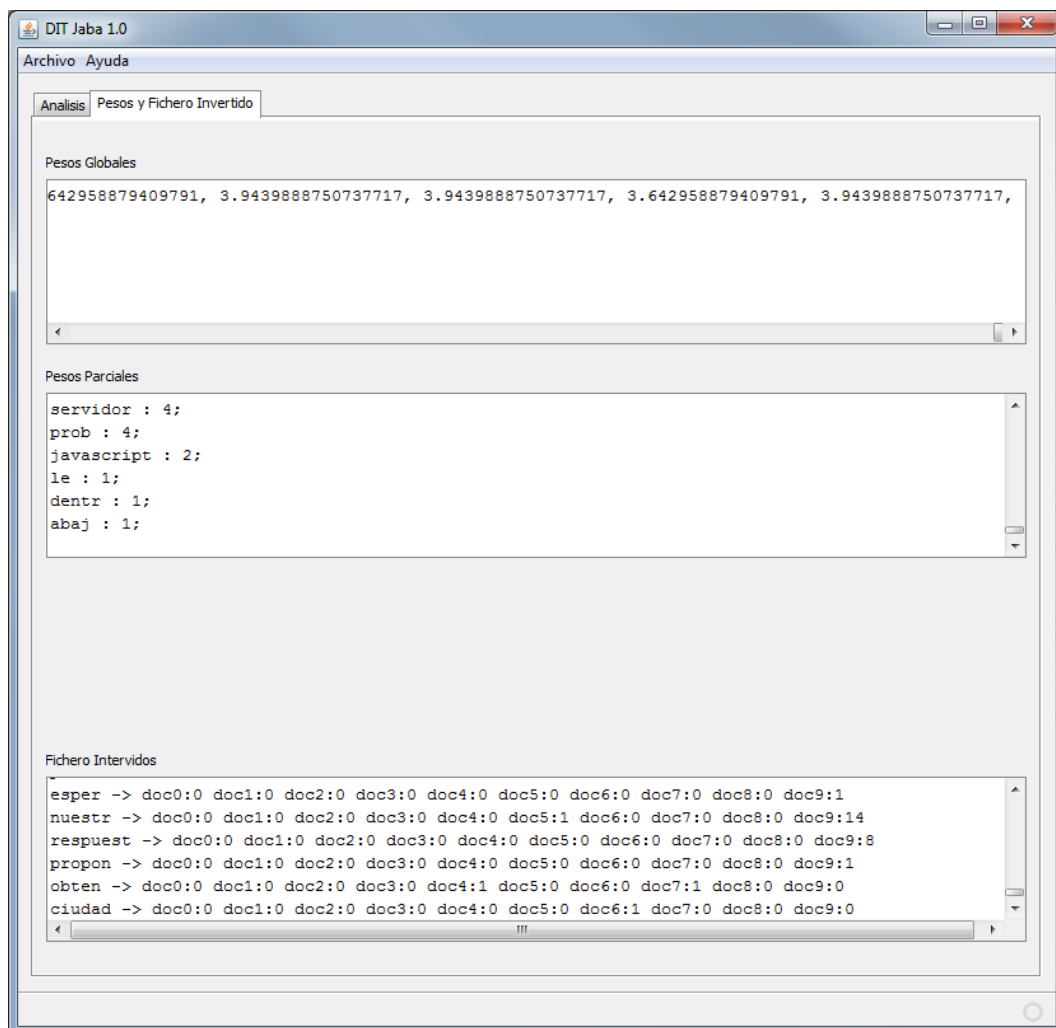


Figura 5: Pestaña de Pesos y Fichero Invertido con la información obtenida de los ficheros anteriormente añadidos.

---

29 de marzo de 2011

3. Después de realizar todos los cálculos, mediante otros bucles necesarios, se obtiene la información de las distintas tablas hash, y se muestran en los respectivos lugares de la aplicación.