6 Implementación y resultados

6.1 Presentación general del sistema

El sistema que se ha implementado está constituido por un conjunto de aplicaciones Java EE que trabajan de forma conjunta para realizar individualmente las tareas de obtención de gorjeos, preprocesamiento, clasificación, agrupamiento, presentación y control de grupos previamente calculados.

La siguiente figura muestra la organización y flujo de los datos del sistema en conjunto:

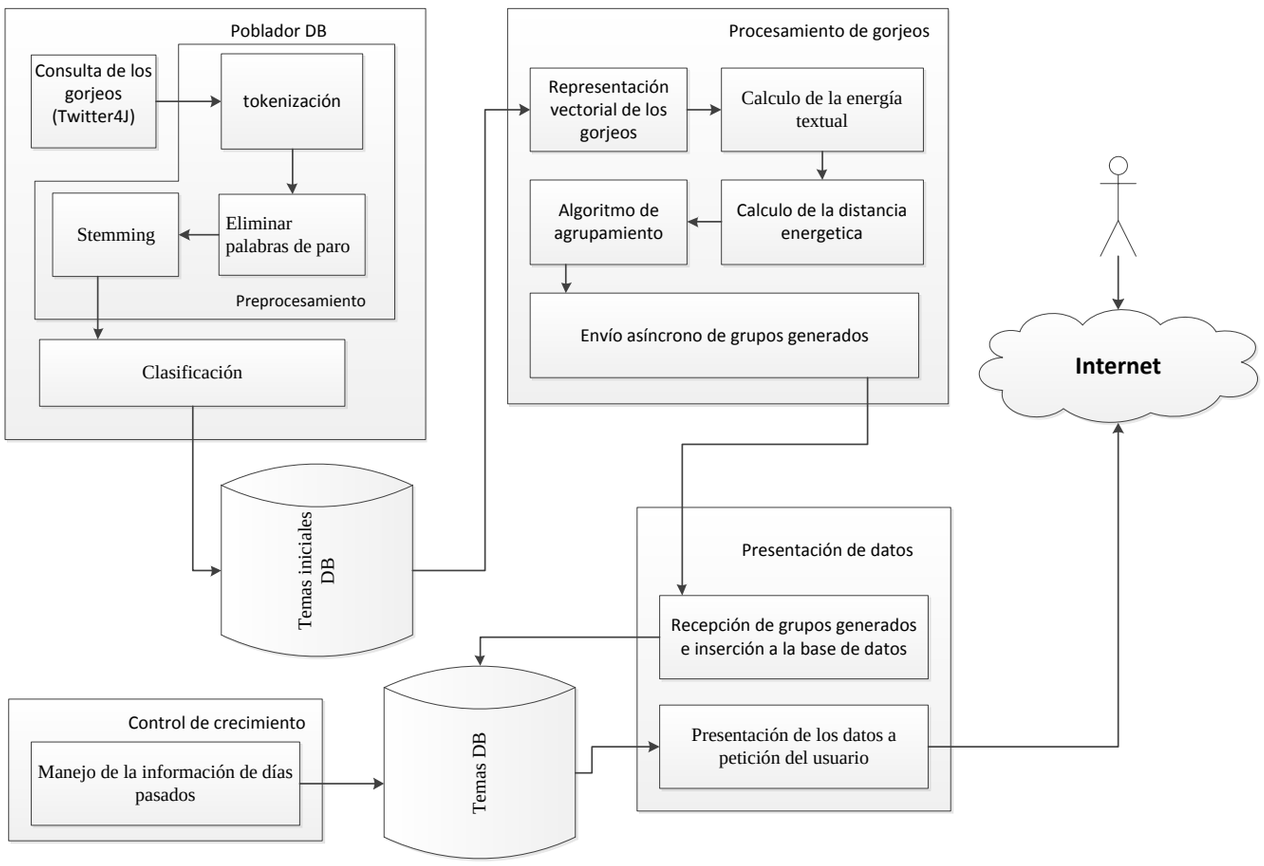

Figura 4 – Arquitectura del sistema propuesto

El primer sistema, llamado *Poblador DB*, es el encargado de obtener los gorjeos al hacer consultas al servidor de Twitter, hacer un preprocesamiento a los gorjeos y realizar una clasificación de los gorjeos en temas previamente seleccionados.

Un segundo sistema etiquetado con el nombre de *Procesamiento de gorjeos* es el encargado de hacer una representación vectorial para cada conjunto de gorjeos previamente clasificados en temas dentro de la base de datos "Temas iniciales DB", de procesarlos para generar una matriz de similitud entre documentos, de agruparlos y finalmente de entregarlos a otra aplicación Java EE en forma de grupos (temas seleccionados previamente) y subgrupos (grupos generados por el algoritmo de agrupamiento).

Presentación de datos es el tercer subsistema encargado de recibir dichos grupos y subgrupos para su adición a la base de datos etiquetada como “Temas DB” y resolver las peticiones de información de los usuarios respecto a los temas y subtemas de la base de datos.

Finalmente el subsistema *Control de crecimiento* es el encargado de eliminar gorjeos considerados como antiguos, para evitar el almacenamiento innecesario de información. En la sección de trabajo futuro se detalla el potencial de este sistema.

En las siguientes secciones se detallan dichos sistemas y subprocesos.

6.2 Obtención de los gorjeos y preprocesamiento

Tanto la obtención de gorjeos como el preprocesamiento se implementaron en una sola aplicación Java EE (*Poblador DB*), esto debido principalmente a la necesidad de una clasificación por temas. La idea es que no todo lo que se diga en Twitter es de interés en este sistema.

Esto es, además de que es conveniente ofrecer la posibilidad de restringir el domino de los gorjeos, hay opiniones de que mucho de lo que se genera no sirve. Por ejemplo, la empresa de investigación de mercado *Pear Analytics*, donde se analizó el contenido de 2,000 gorjeos durante un periodo de 2 semanas [16]. La empresa consideró que *palabras sin sentido*, fue la categoría más importante de los contenidos de Twitter, que componen 811 gorjeos (40.55%) del número total de mensajes incluidos en la muestra[22].

Los mensajes de conversación representaron un total de 751 gorjeos (33%); los *tweets* repetidos, es decir, los *retweets* (RT) representaron 174 gorjeos (8.70%); la autopromoción de empresas forma 117 gorjeos (5.85%); los mensajes basura conforman 75 gorjeos (3.75%), y gorjeos con noticias de medios de comunicación representaron 72 (3,60%).

En este contexto, para la clasificación se decidió insertar en la base de datos de “Temas iniciales DB” un conjunto de temas asociados a una lista de palabras que lo representan, para filtrar lo que serían gorjeos con contenido irrelevante desde el punto de vista de los temas de interés que se analizan.

A continuación se detalla cada uno de los subprocesos realizados por la aplicación *Poblador DB*

[22] La empresa no hizo públicos los criterios que utilizó para determinar que una palabra no tiene sentido.

6.2.1 Obtención de gorjeos

Para la obtención de los gorjeos (*tweets*) la aplicación hace uso del servicio de temporizador de Java EE (*timerservice*) que permite la ejecución de procedimientos en intervalos de tiempo definidos por el desarrollador. Con dicho servicio el sistema realiza las consultas al servidor de Twitter en un determinado intervalo de tiempo (para el presente trabajo se realizan cada 10 segundos) con los parámetros de ubicación geográfica, radio en kilómetros y la cantidad de gorjeos que se desean obtener.

Las consultas se realizan en conjunto con la API Twitter4J, una librería de código abierto escrita en lenguaje Java que interactúa con la API de Twitter. Los gorjeos obtenidos no sólo contienen los 140 caracteres de los mensajes publicados, sino también datos como: quién lo publico, el identificador (*id*) de todos sus seguidores, a qué usuario fue dirigido dicho mensaje e incluso datos relevantes para el presente trabajo, como lo son la ubicación geográfica o el momento exacto en que se generó dicho gorjeo.

Por lo tanto, la entrada del sistema se compone de una colección de gorjeos que contienen entre otras cosas el mensaje publicado por un usuario.

La figura 5 representa un conjunto de gorjeos.

G1: Casi se me había olvidado el tráfico matutino de la ciudad capital!

G2: Me dispongo a salir a la ciudad a sufrir del tráfico, me dirijo a marina nal desde ínter a ver cuanto tiempo hago

G3: Y esto, señoras y señores, es el caos vial Bicentenario

G4: Buenos dias! Miercoles de plaza... y de cierres viales con motivo del Bicentenario, tardaran 100años en llegar a su destino

G5: No bueno k trafico en esta ciudad ossh !!!

Figura 5 – Ejemplo de gorjeos

Con el objetivo de enfocar la atención al mensaje de texto contenido en el gorjeo publicado en Twitter, en el presente trabajo se hace referencia al gorjeo como el mensaje y a sus atributos.
Hay que tener en cuenta que la misma naturaleza de Twitter hace que estos mensajes contengan una gran cantidad de términos impredecibles, como faltas ortográficas y mezcla de mayúsculas con minúsculas. Es importante considerar lo anterior puesto que para una computadora términos como lo son "trafico", "tráfico" y "Tráfico" son totalmente diferentes, lo que hace necesario un método para facilitar el procesamiento de los mensajes y unificación de términos .

La siguiente sección describe el tratamiento por el que pasan los gorjeos para facilitar su representación y procesamiento.

6.2.2 Preprocesamiento

Esta etapa está conformada de seis transformaciones aplicadas a cada uno de los gorjeos para representarlos sólo por *tokens*, un conjunto de términos en minúsculas, truncados (para eliminar las flexiones), libres de puntuación y separados por espacios en blanco para así reducir el tamaño del espacio vectorial.

Como primera transformación se agrega un atributo extra al gorjeo, dicho atributo es una copia del mensaje original publicado por el usuario y es en éste donde se hacen las siguientes transformaciones.

La segunda transformación extrae temporalmente las direcciones URL, ya que éstas se deben considerar como una sola palabra y no deben sufrir ningún cambio.

En la tercera transformación se cambia todo el mensaje a minúsculas y elimina los signos de puntuación y diacríticos con el objetivo de unificar símbolos, como por ejemplo "U","Ú" y "ú" en el símbolo "u".

Posteriormente una cuarta transformación elimina las palabras con poca información semántica haciendo pasar el gorjeo por una comparación con una lista de paro; es decir, una lista de palabras funcionales (preposiciones, determinantes, conjunciones) y palabras frecuentemente utilizadas en el contexto del sistema.

La quinta transformación reduce las palabras a sus bases o raíces mediante el algoritmo de Porter [17]. Dicha transformación es una técnica muy utilizada en el área de recuperación de información, ya que unifica palabras como *"efectúa", "efectuar", "efectuaron", "efectuarse"* en un solo símbolo "*efectu"* que es su raíz, con el objetivo de mejorar la relación de gorjeos semánticamente parecidos.

Finalmente la sexta transformación inserta de nuevo la dirección URL si es que fue extraída en la primera transformación considerándola así como un *token* más.

La figura 6 muestra los gorjeos después de pasar por todas las transformaciones.

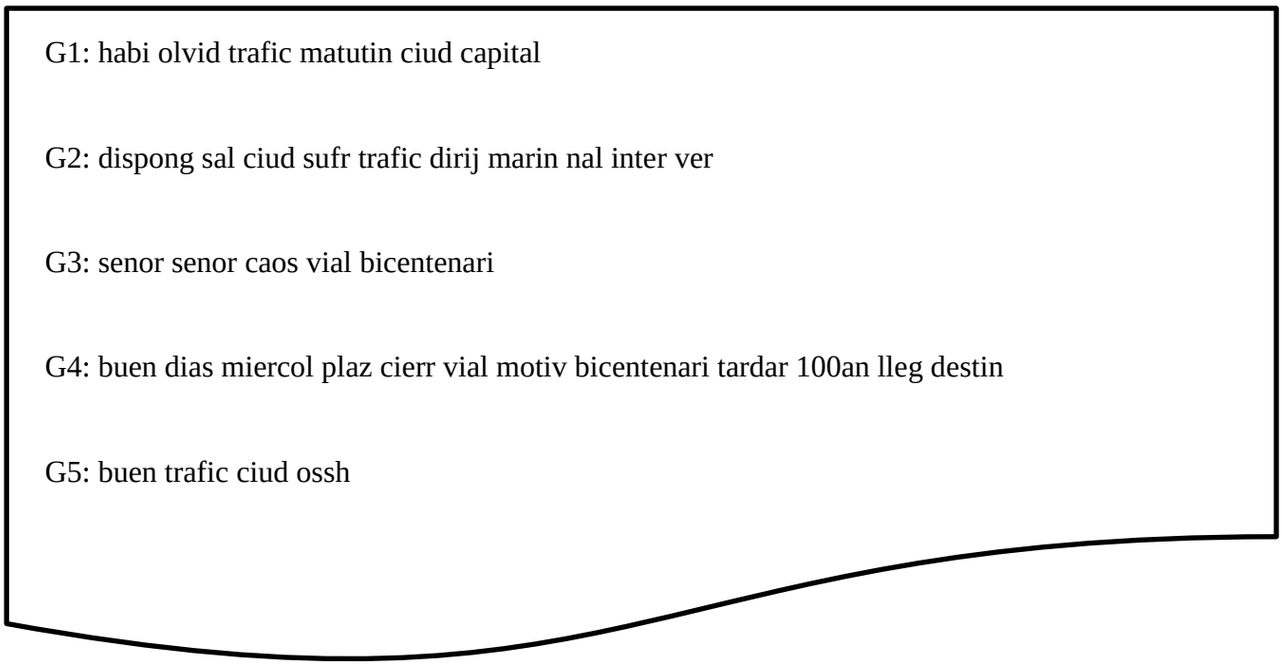

Figura 6 – Resultado de aplicar las transformaciones a los gorjeos

Con todas estas transformaciones se reduce el número de palabras, mejorando así las coincidencias entre todos los gorjeos, pero, por ejemplo, no se hace ninguna diferenciación entre nombres propios o comunes, o los términos que cambian de sentido según el contexto.

6.2.3 Clasificación

Con los *tokens* obtenidos durante el preprocesamiento se hace una búsqueda en la base de datos "Temas Iniciales DB" en la que se encuentran almacenados un conjunto de términos enraizados[23] igualmente por el algoritmo de Porter. Cada uno de estos términos se asocia a otro conjunto de términos que conforman un tema en común, consiguiendo así que los gorjeos que contengan palabras referidas o relacionadas a un tema sean clasificados en uno o varios de los temas en la base de datos.

El lector se preguntará, ¿cuáles son esos temas y que términos los relacionan? La respuesta a esta pregunta es: los temas y las palabras que el usuario desee. El sistema se ha pensado con la idea de ser flexible a los temas de interés. Ver apéndice C.

[23] Palabras truncadas donde se eliminan los sufijos .

6.3 Procesamiento de los gorjeos

La aplicación Java EE encargada del procesamiento realiza un tratamiento al conjunto de gorjeos para facilitar la generación de un espacio vectorial que sea representativo para el agrupamiento, el cálculo de la distancia entre los gorjeos a partir de una matriz de similitud conocida como *energía textual* y la generación de grupos semánticamente similares.

6.3.1 Construcción del espacio vectorial

En esta etapa se construye una matriz concebida como un arreglo de vectores que representan los gorjeos. El número de renglones está determinado por el número de gorjeos en dicho tema y el número de columnas por el tamaño del diccionario de palabras graficas que aparecen por lo menos dos veces en todo el conjunto de gorjeos. La generación de la matriz documento-término se lleva a cabo iniciando una matriz de ceros de *n x m* donde *n* es el número de gorjeos y *m* el número de palabras gráficas en el diccionario. Ya creada dicha matriz, el conjunto de gorjeos es recorrido uno por uno y un valor de 1 es insertado a la matriz documento-término en la posición correspondiente a la entrada de dicha palabra en el diccionario de palabras gráficas.

G1: habi olvid trafic matutin ciud capital

G2: dispong sal ciud sufr trafic dirij marin nal inter ver

G3: senor senor caos vial bicentenari

G4: buen dias miercol plaz cierr vial motiv bicentenari tardar 100an lleg destin

G5: buen trafic ciud ossh

Figura 7 – Palabras tomadas en cuenta para el espacio vectorial

	trafic	ciud	vial	bicentenari	buen
G1	1	1	0	0	0
G2	1	1	0	0	0
G3	0	0	1	1	0
G4	0	0	1	1	1
G5	1	1	0	0	1

Tabla 5 – Espacio vectorial para el ejemplo de 5 gorjeos

La matriz resultante es una matriz con peso local binario, peso global sin cambios y ningún tipo de normalización, pero esto no impide realizar un análisis más sofisticado con valores de frecuencia del término y otras combinaciones. La razón principal radica en que el cálculo de la energía textual es solamente con una matriz binaria [10] y que dicho método resulta computacionalmente menos costoso que otras técnicas.

6.3.2 Cálculo de la energía textual

Una vez generado el espacio vectorial surge la necesidad de tener una representación que sirva de mecanismo para conocer grupos semánticos, es decir, qué tan similares son unos gorjeos de otros. Se eligió como medida de similitud la energía textual propuesta en [10], la cual ofrece numerosas ventajas como la rapidez para realizar su cálculo, su eficacia y que desde sus inicios fue concebida como una aproximación para las relaciones entre documentos.

Dado que el algoritmo de agrupamiento conocido como *método del mínimo* requiere una matriz de distancias, a continuación se deduce la medida de distancia energética a partir de la fórmula de energía textual.

Considérese la matriz documento-término X

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & & & \vdots \\ X_{n1} & X_{n2} & & X_{nm} \end{pmatrix}$$

La matriz de energía textual (matriz de similitud) E asociada a X se calcula como:

$$E = -\frac{1}{2}(XxX')^2$$

Que como se mencionó en el capítulo 4 es equivalente a decir que:

$$E = -E_{textual}$$

Donde E es una matriz de *n* x *n* dada por:

$$E = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & & & \vdots \\ e_{n1} & e_{n2} & & e_{nn} \end{pmatrix}$$

Al ser la energía textual una medida de similitud, la máxima energía (la mayor magnitud) se encuentra en la diagonal principal, lo que significa que un documento es más parecido a sí mismo que otros documentos.

Una vez construida la matriz de energía textual, basta usar la transformada de Gower [9] para calcular la distancia energética entre gorjeos.

$$d_{ij}^2 = s_{ii} + s_{jj} - 2s_{ij}$$

De esta forma, se tiene la posibilidad de utilizar un algoritmo de agrupamiento para generar grupos utilizando la distancia entre gorjeos como criterio.

$$D_{energética} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & & & \vdots \\ d_{n1} & d_{n2} & & d_{nn} \end{pmatrix}$$

En algunas ocasiones se desea que las distancias estén normalizadas y que tengan un rango de [0,1] y para ello sólo hace falta toma la matriz de distancia energética y dividir cada uno de sus elementos entre el valor máximo. Esto es:

$$D_{energéticaNormalizada} = \left(\frac{1}{\max(D_{energética})}\right)(D_{energética})$$

Hay que aclarar que la normalización no tiene efecto sobre los resultados pero consume tiempo de cómputo innecesario, por lo que no se usa en el presente trabajo.

Continuando con el ejemplo, es posible calcular a partir de la matriz documento-término la energía textual y, de ésta, la matriz de distancia energética.

Sea la matriz documento-término calculada anteriormente:

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Con la que es posible calcular la matriz de energía textual:

$$E = \frac{1}{2}(XxX')^2 = \begin{pmatrix} 6 & 6 & 0 & 1 & 7 \\ 6 & 6 & 0 & 1 & 7 \\ 0 & 0 & 4 & 5 & 1 \\ 1 & 1 & 5 & 7 & 3 \\ 7 & 7 & 1 & 3 & 9 \end{pmatrix}$$

Aplicando la transformada de Gower se calcula la matriz de distancias entre documentos o distancia energética (nótese que, como se dijo arriba, no está normalizada):

$$D_{energética} = \begin{pmatrix} 0 & 0 & 100 & 121 & 1 \\ 0 & 0 & 100 & 121 & 1 \\ 100 & 100 & 0 & 1 & 121 \\ 121 & 121 & 1 & 0 & 100 \\ 1 & 1 & 121 & 100 & 0 \end{pmatrix}$$

Esta matriz es necesaria para el algoritmo de agrupamiento jerárquico que en cada paso une dos grupos cuyos elementos más cercanos tienen la distancia mínima.

6.3.3 Agrupamiento de gorjeos

Una vez obtenida la matriz de distancia energética el sistema genera grupos usando un algoritmo de agrupamiento jerárquico conocido como *método del mínimo*.

Este algoritmo jerárquico ofrece la ventaja de no requerir especificación alguna sobre los grupos deseados, pero es necesario especificar un umbral de *corte por distancia* sobre el dendograma. Este umbral indica la distancia máxima que puede haber entre dos grupos. Por ejemplo, si el umbral es de 10 unidades significa que aquellos grupos cuya distancia sea menor a 10 serán unificados. El algoritmo de agrupamiento parte de la idea de que cada gorjeo es un grupo y se agrupan los más cercanos hasta llegar a ser un solo grupo, por lo que es necesario especificar dicha *distancia de corte*. En el caso del presente trabajo no se toman en cuenta los grupos de un solo gorjeo ya que la intención es encontrar gorjeos que compartan rasgos similares.

Realmente no se conoce por el momento un método capaz de calcular la *distancia de corte* óptima pero en general se analizan resultados obtenidos con distintos valores de $\propto$ hasta obtener resultados aceptables. En el presente trabajo eso no es posible puesto que una distancia de corte óptima para un conjunto de datos no es necesariamente la mejor para un nuevo conjunto de gorjeos entrante al sistema, por lo que la distancia es seleccionada arbitrariamente como la mitad del índice de fusión máximo ($\propto$ donde todos los gorjeos se fusionan en un solo grupo) o incluso niveles inferiores a la mitad del dendograma, consiguiéndose así un número variable de grupos generados y muy parecidos entre sí.

Para el ejemplo que estamos manejando, los grupos generados son los siguientes (ver figura 8):

GRUPO 1:

G1: Casi se me había olvidado el tráfico matutino de la ciudad capital!

G2: Me dispongo a salir a la ciudad a sufrir del tráfico, me dirijo a marina nal desde ínter a ver cuanto tiempo hago

G5: No bueno k trafico en esta ciudad ossh !!!

GRUPO 2:

G3: Y esto, señoras y señores, es el caos vial Bicentenario

G4: Buenos dias! Miercoles de plaza... y de cierres viales con motivo del Bicentenario, tardaran 100años en llegar a su destino

Figura 8 – Grupos generados para gorjeos de ejemplo

Ya generados los grupos éste subsistema hace uso del Java Message Service (JMS), un estándar de mensajería para crear, enviar, recibir y leer mensajes permitiendo comunicación síncrona y asíncrona. En el presente trabajo se utilizó JMS para hacer una comunicación con el subsistema *Presentación de datos* con un modelo punto a punto. Este modelo asegura la llegada del mensaje, ya que si el receptor no está disponible para aceptar el mensaje o atenderlo, de cualquier forma se le envía el mensaje y este

entra en una estructura conocida como pila. Los mensajes de esta pila se reciben después según hayan sido enviados, dando como resultado la no dependencia de ambos y la mejora en tiempo de procesamiento.

6.4 Presentación de datos

El subsistema *Presentación de datos* es el encargado de recibir los grupos de gorjeos, de hacer un conteo de los términos más utilizados y de guardarlos en la base de datos *Temas DB* con la fecha y la hora en que ingresan al sistema para posteriormente ser consultados por la capa Web de este subsistema.

Al recibir los mensajes del JMS se toman cada uno de los grupos (subtemas generados por el algoritmo de agrupamiento) que van llegando y se hace un conteo de los términos más utilizados para cada uno de ellos a fin de mostrar al usuario los términos con los que fueron unidos los gorjeos a dicho grupo. Ya por último se guardan los grupos generados en la base de datos *temas DB* donde son asociados a los temas definidos *a priori* para ser consultados por los usuarios.

La figura 9 muestra la ventana principal de navegación, del lado derecho (ver apartado “Selecciona el tema de interés”) se enlistan los temas que se están actualizando constantemente y el número de grupos que conforman a ese tema específico en un determinado lapso de tiempo y debajo del mismo (ver apartado “Selecciona un periodo de tiempo”) es posible navegar por grupos generados tiempo atrás por el sistema.

Figura 9 – Página de inicio

También es posible consultar los resultados obtenidos por el algoritmo de agrupamiento para un conjunto de pruebas (ver capítulo: "análisis de resultados" del presente trabajo).

Al seleccionar un tema de interés el sistema imprime un conjunto de tablas donde cada una representa un subtema (ver figuras 10 y 11); en cada una de las tablas se muestra la imagen del usuario de Twitter y el gorjeo en sí mismo.

En la cabecera de cada una de las tablas se imprimen los 5 términos truncados con por lo menos dos apariciones en subconjunto de gorjeos, o lo que es lo mismo, los términos más relevantes y por los cuales se fusionaron los gorjeos en la etapa de agrupamiento.

Figura 10 – Presentación de datos para el tema *Homicidios*

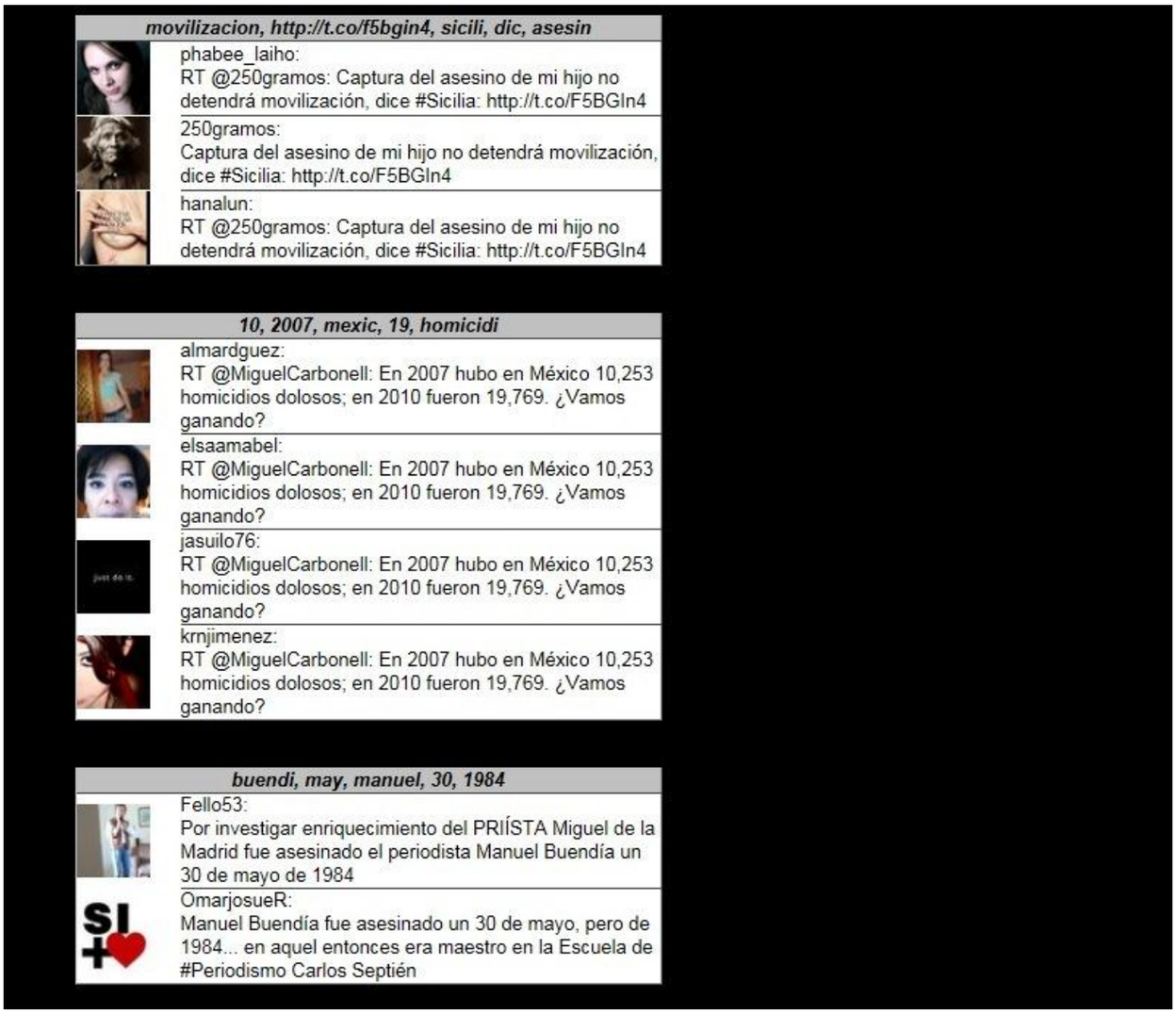

Figura 11 – Continuación de presentación de datos para el tema *Homicidios*

También es posible imprimir el contenido de los gorjeos como fueron publicados ("gorjeo original"), su representación después de la etapa de preprocesamiento ("gorjeo procesado") e incluso imprimir solo los términos que lo componen y que corresponden al diccionario de términos para el procesamiento de los gorjeos ("gorjeo vectorizado"). Ver figuras 12,13 y 14.

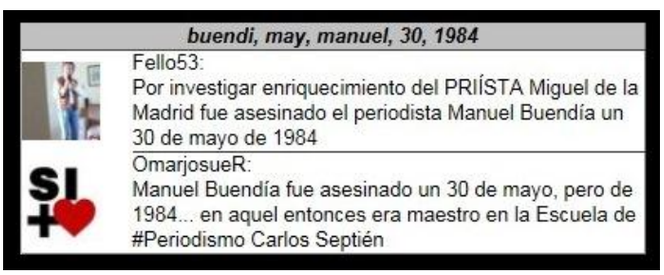

Figura 12 – Presentación de gorjeo original

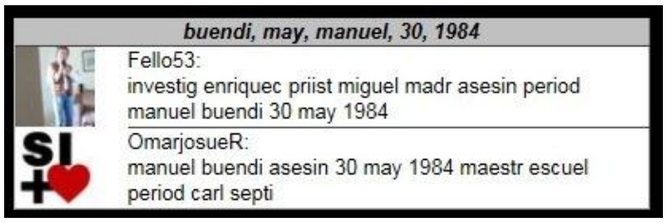

Figura 13 – Presentación de gorjeo procesado

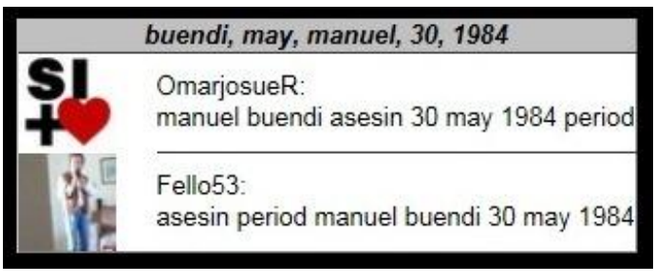

Figura 14 – Presentación de gorjeo vectorizado

El subsistema *Presentación de datos* implementa los componentes de vista y controlador con ayuda del *framework* de Java Server Faces (JSF), tecnología basada en *servlets* y JSPs pero que extiende sus funcionalidades. JSF se enfoca en la vista del patrón MVC y utiliza componentes gráficos en el lado del servidor, procesa peticiones estándares HTTP y además permite a sus propios componentes comunicarse entre sí.

Algunas de las ventajas de usar JSF fueron:

- Internalización[24] del sistema: esto quiere decir que el sistema se muestra en otros idiomas dependiendo de la configuración del usuario.
- Maquetado: se utiliza una tecnología conocida como *facelets* permitiendo cambiar la apariencia con tan solo realizar una modificación a la plantilla de presentación.
- Manejo de sesiones: El sistema recuerda las acciones del usuario para mantener la vista de la aplicación.
- Facilidad de integración: JSF es la manera más natural para la presentación de los datos, siendo tan simple que basta con hacer llamadas a procedimientos de la lógica de negocios para que sea actualizada la vista.

[24] Del termino ingles *internationalization*, utilizado por desarrolladores web para referirse al soporte multilenguaje de una aplicación web.

6.5 Control de crecimiento

El subsistema *Control de crecimiento* es el encargado de administrar la base de datos *Temas DB*. Hay que recordar que son datos traídos del internet cada 15 segundos. Por lo tanto, el número de gorjeos almacenados es increíblemente alto. Para el alcance del presente trabajo, este subsistema sólo se encarga de eliminar grupos de gorjeos que hayan superado un tiempo razonable en el sistema, por ejemplo un mes; lo que es algo trivial y no necesita mayor explicación, puesto que sólo se trata de eliminar información de la base de datos *Temas DB*.

La importancia real de este subsistema se puede ver desde el punto de vista de minería de textos, desde él se ve qué hacer con dicha información, qué más presentar o qué estadísticas obtener a partir de grupos ya generados, puntos que se tomarán en cuenta en el apartado de conclusiones y trabajo futuro.