16

**Recuperación de Información**

**Sistema de Recuperación Tradicional**

**Jorge Andrés Galindo 679155 Javier Aranda García 679184**

**ANÁLISIS DE LOS DOCUMENTOS**

Previamente, antes de realizar el proceso de indexación, se realizó un exhaustivo examen de todos los documentos, para conseguir la máxima información posible que ayudase a realizar el programa más completo y más adaptado a la información. El primer campo a analizar fue el nombre y número de etiquetas presentes en cada uno de los ficheros XML. Para ello, se observó que había como máximo 7 etiquetas diferentes entre todos los documentos, y que además, todos poseían estas 7, salvo 4 documentos que no tenían idioma. Las etiquetas encontradas fueron “identifier”, “language”, “creator”, “title”, “description”, “publisher” y “date”.

Sabiendo los campos de los documentos, se pasó a obtener información de cada uno de ellos por separado. En primer lugar, se vio que todos los documentos tenían al menos dos identificadores distintos, y que en todos salvo en 8 documentos uno de ellos estaba repetido, por lo que se debía pensar si guardarlo dos veces. A continuación, se prosiguió con el idioma. La etiqueta “language” indicaba en que idioma había sido escrito el trabajo que representaba el documento XML. Se observó que como máximo solo tenían una etiqueta de este tipo (salvo algún documento que no tenía ninguna), y que además tenían valores tales como “eng”, “spa”, “por”,”fre”…

En lo que a la etiqueta “creator” se refiere, todos los documentos tenían al menos un campo de este tipo, pero en la gran mayoría de ellos se nombraba a más de una persona responsable de realizar ese trabajo (hasta 6 creadores en un mismo documento). Analizando la etiqueta “title”, como era de esperar, estaba contenida en todos los documentos a indexar, salvo que se vio que en dos casos estaba repetida en el mismo documento. Con la etiqueta “description” pasaba algo parecido, salvo que aquí estaba repetida pero en dos idiomas distintos.

Mirando detenidamente el “publisher”, se observó que estaba en todos los documentos, y que tomaba hasta tres valores diferentes: “Universidad de Zaragoza”, “Universidad de Zaragoza, Universidad de Zaragoza” y “Universidad de Zaragoza, Prensas de la Universidad”. Finalmente, la etiqueta que faltaba por analizar era “date”, que simplemente era un número de cuatro cifras que representaba el año en el que había sido realizado el trabajo, y que además, tenía una vez presente en cada uno de los documentos de la colección.

**INDEXACIÓN**

Tras analizar los documentos detenidamente, se procedió a indexar la información. En primer lugar se debatió como organizarla en distintos índices o campos, llegando a la conclusión de que lo más cómodo era hacer tantos índices como etiquetas distintas podían tener los archivos XML. Se pensó si se podía desechar alguno de los campos, como podría ser el “publisher”, ya que tomaba valores muy parecidos, y la colección de documentos era tan sólo de la Universidad de Zaragoza; sim embargo, se optó por dejarlo por si en algún momento se añadían documentos de otras universidades.

El campo “identifier”, como ya se vio en el análisis previo de los documentos, constaba de dos códigos identificativos distintos, y además, en muchos documentos estaba repetido uno de ellos. Se almacenaron los dos códigos en el mismo campo del índice sin repetir aquel que estuviera más de una vez en los documentos. Para la etiqueta “creator”, se guardaron todos los posibles creadores de un trabajo en un mismo campo del índice, concatenados uno detrás de otro, y separados por un salto de línea para diferenciarlos con facilidad. En lo que al “title” se refiere, como se comentó en el análisis previo, tan sólo había uno por documento, salvo en contadas ocasiones que había dos pero el mismo repetido. En este caso, se siguió una estrategia similar que para “identifier”, guardándolo en un campo propio, e ignorando la repetición del título si la hubiese. La fecha o “date” se almacenó en un campo propio con normalidad ya que no daba ningún problema en los documentos.

Para el campo “language”, se decidió entre guardarlo tal cual se leía en su campo, o realizar una modificación y guardar la correspondencia del código (“eng” pasarías a inglés). Finalmente, se optó por guardar la correspondencia con el nombre completo del idioma, ya que para las consultas el parser podía dar problemas con la búsqueda del código, siendo más cómodo buscar con el nombre entero del idioma. Además, aquellos 4 archivos sin idioma, se revisaron y se decidió que si el indexador no encontraba etiqueta de idioma pusiera español “a mano”. Esta decisión se tomó porque aquellos de la colección que no tenían, estaban en español, y la probabilidad de que futuros documentos sin idioma fueran españoles era más alta que de cualquier otro idioma.

Para el “publisher”, como ya se ha comentado, se tomó la decisión de almacenarlo tal cual en un campo del índice, sin modificar nada. Finalmente, el último campo a tratar fue “description”, y el que más problemas dio. Se observó en el análisis previo que había documentos que lo tenían repetido y en dos idiomas (mayoritariamente en español y en inglés). La solución tomada para otros campos como podía ser almacenarlos concatenados se desechó porque las consultas serían realizadas casi en la totalidad de ocasiones en español, por lo que el score de una consulta con un gran texto en inglés bajaría considerablemente a pesar de que en la parte en español coincidiera con la necesidad de información. Así pues, se decidió indexar solo el texto en español en caso de que tuviera más de dos etiquetas “description”, que en casi todos los documentos era el primero encontrado.

Una vez decididos los campos a utilizar y delimitado el filtro a realizar con el contenido de cada documento a guardar, se pasó a decidir qué tipo de campos eran necesarios para almacenar cada uno de los índices. La decisión estaba clara y era de tipo texto en todos los casos salvo para la fecha, que se decidió que fuera de tipo “int” ya que era el año de la publicación (número entero), y podía ser de gran ayuda para las comparaciones en las consultas. El campo de tipo texto podía ser de tipo “TextField” o “StringField”. El tipo “StringField” es un campo que se indexa y no se tokeniza, guardado la cadena de texto como un único token. Sin embargo, el campo “TextField” se indexa y tokeniza, por lo que para textos grandes en los que se quiere coincidencia de palabras y no del texto entero sería más adecuado.

La decisión que se tomó fue usar “TextField” con un analizador de tipo “SpanishAnalyzer” que, aunque diera peores resultados que otros en campos como el identificador o en textos en inglés, en su totalidad era el que mejor resultado daría para las futuras consultas sobre el indexado. Además, se estudió la posibilidad de usar algún otro tipo de tokenizador (espacios en blanco, estándar…), pero se vio que el mejor funcionaría era el finalmente elegido, ya que se probó a hacer consultas simples sobre el campo identificador indexado con este analizador, y a pesar de que los resultados no eran los exactos, sacaba en primeras posiciones siempre los documentos más relevantes.

**CONSULTAS**