UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN: ESPECIALIDAD TELEMÁTICA

PROYECTO FINAL DE CARRERA

REALIZACIÓN DE UN TRAZADOR DE PISTAS INTELIGENTE Y DINÁMICO PARA NAVEGACIÓN EN LA WEB

AUTOR: CRISTINA MARTOS TABERNILLA
TUTORA: IRIA M. ESTÉVEZ AYRES
DIRECTOR: JOSÉ JESÚS GARCÍA RUEDA

OCTUBRE 2014

A José Jesús, mi tutor durante todos estos años, por la paciencia de la que ha hecho gala y su disposición para que este proyecto finalmente haya salido adelante.
A mis padres, y mis abuelos, porque, aunque no siempre entienden este empeño, me apoyan y me animan a seguir mi camino.
A mi hermana Nerea, que siempre confía en mí en todos los aspectos de la vida, y a la que siempre recurro cuando me siento perdida.
A Iria, Cristina, Paloma, Marta, Carlos y Teresa, por sus ánimos continuos en este proyecto.
A Sara, por ser la luz que ha iluminado los momentos oscuros este último año, y mis pies en la tierra incluso de madrugada.
Y a Urko, mi trozo de cielo, que desde el principio creyó que esto era posible, por ser, por estar… porque como dice la canción "sin ti pa'qué".
A todos ellos, GRACIAS.

Resumen

En este proyecto se desarrolla una aplicación cuya intención es ayudar al usuario a navegar por la web, ofreciéndole páginas similares a las que está visitando, y basándose en las elecciones previas que el usuario realizó, tratando así de implementar un trazador de pistas o Trailblazer a partir de un sencillo y simplificado Sistema de Recomendación basado en contenido.

Para ello, la aplicación se centrará en la parte del Sistema de Recomendación, que consiste en analizar el contenido semántico de las URLs que el usuario esté visitando, desarrollando un mecanismo o algoritmo que permita seleccionar las palabras más relevantes y utilizarlas en una consulta a un buscador web, ofreciendo al usuario las respuestas que la aplicación considere las más acertadas para el usuario. Se simplificará la obtención de las páginas webs visitadas en la Red por el usuario, pues queda considerado fuera del alcance de este proyecto.

Índice de Contenido

Índice de figuras

Índice de tablas

1 Introducción

En este apartado se explicará la motivación que ha llevado a realizar este proyecto, el objetivo final del mismo y se introducirán brevemente los Sistemas de Recomendación, ahondando especialmente en los sistemas basados en contenido.

1.1 Motivación y contexto

Durante los últimos años se ha producido un aumento de utilización de la red como medio tanto de información como de entretenimiento a nivel global, y, por tanto, un crecimiento exponencial de contenido web disponible a los usuarios. Las llamadas nuevas tecnologías han dejado de ser tal para convertirse en herramientas incorporadas al día a día del usuario, tanto a nivel profesional como a nivel personal, y convirtiéndose en una herramienta imprescindible en ambos entornos.

Según datos del Instituto Nacional de Estadística, en su "***Encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación (TIC) en los Hogares Año 2012***", (1) el 67,9% de los hogares españoles ya tiene acceso a la Red, lo que supone un total de 10.5 millones de viviendas con conexión a Internet.

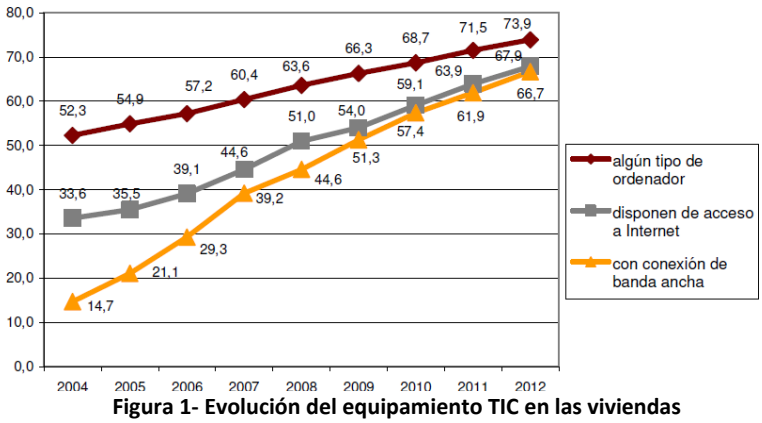


Figura 1- Evolución del equipamiento TIC en las viviendas (% de viviendas)

Además, el número de usuarios frecuentes entre la población de 16 a 74 años es del 65%, es decir, más de la mitad de la población activa accede a la Red a diario, o al menos una vez por semana.

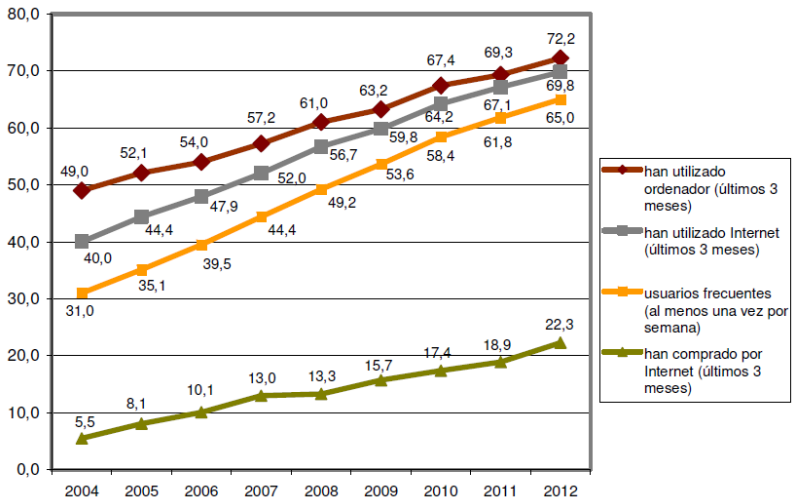


Figura 2- Evolución del uso de TIC por las personas de 16 a 74 años

A la disponibilidad de acceso a la Red en los hogares, hay que sumarle además la posibilidad de acceder a Internet a través de los dispositivos móviles, tipo Tablets o Smartphones, donde el uso de este último en el año 2013 se extiende al 57% de la población, según el último ***Eurobarómetro sobre Comunicaciones electrónicas en el hogar*** (2).

Sin embargo, no siempre resulta fácil e intuitivo encontrar la información que se busca, bien porque se dispone de demasiada información que el usuario no es capaz de clasificar, bien porque el usuario no dispone de cultura tecnológica suficiente para encontrar dicha información.

La proliferación de sitios web tanto particulares como de empresas no siempre supone una ventaja para el usuario, pues a menudo se encuentra con información repetida, de baja calidad o diferente a la esperada, que le supone invertir mucho esfuerzo y tiempo de análisis para extraer contenido que le resulte útil.

Por otro lado, si bien es cierto que las nuevas generaciones de la población está creciendo ya dentro de un entorno completamente digital, donde las nuevas tecnologías están totalmente integradas en su día a día, y su manejo se aprende de forma natural, todavía existe un amplio sector de la población que se encuentra enmarcado dentro de una generación que se podría denominar “analógica”, donde las nuevas tecnologías son algo novedoso con lo que no están familiarizados, y donde la comprensión y el manejo de estas nuevas herramientas no siempre alcanzan un nivel óptimo. Así, a menudo se dan casos de usuarios que ven frustrados sus intentos de búsqueda de información porque su capacidad para navegar por la Red se ve muy limitada, teniendo un conocimiento básico sobre el uso de un PC o teléfono móvil, pero no comprendiendo los mecanismos necesarios, como el funcionamiento de buscadores, para obtener la información deseada.

Este proyecto surge con el propósito de crear una herramienta que facilite la tarea de navegación por la web al usuario, dándole mecanismos con los que pueda encontrar páginas de su interés de una manera más rápida y cómoda, mejorando la experiencia en el entorno web del usuario.

Se pretende que esta aplicación sirva por un lado a que aquellos usuarios que no posean un manejo y conocimiento muy amplio de la Red, vean facilitada la navegación por la misma, encontrando de una manera más fácil páginas web que sean de su agrado, y por otro lado a que los usuarios con una habilidad mayor para usar Internet dispongan de una herramienta que les resulte cómoda y les acorte la tarea de navegación por la Red.

1.2 Objetivos

El objetivo de este proyecto es construir un Trazador de Pistas o Trailblazer que sea capaz de orientar y ayudar a un usuario cuando navega por la Red, ofreciéndole páginas similares a las que está visitando, y que también sean de su interés.

El proceso completo de la aplicación comprendería los siguientes pasos:

- detección de manera dinámica de la página web que el usuario está visitando,
- obtención del código HTML de la misma,
- extracción de la información relevante,
- análisis de la información,
- almacenaje de información relativa a páginas anteriores,
- combinación de información previa y actual para proporcionar al usuario una información lo más ajustada posible a sus intereses,
- evaluación de información global y elección palabras para enviar al buscador,
- búsqueda de contenido similar a través de un buscador para ofrecer de nuevo al usuario,
- evaluación de contenido devuelto por el buscador y muestra de resultados a usuario.

Sin embargo, este proyecto se centrará únicamente en lo que se considera que es el núcleo de la aplicación: el **análisis de la información, combinación con información previa y la elección de palabras para lanzar al buscador**.

De esta forma, el resto de puntos que conforman el sistema completo, se implementarán primando la sencillez, y teniendo en mente que el objetivo final de este proyecto consiste en abordar y desarrollar el núcleo central que compondría la herramienta. Así, tras diversos análisis que se explicarán en capítulos posteriores, se han simplificado diversas tareas que se han considerado que traspasaban el alcance de este proyecto, como por ejemplo la obtención de la URL que el usuario está visitando y el contenido HTML que contiene la misma, y en lugar de resultar un proceso transparente para el usuario, es el propio usuario el que tiene que introducir la URL en nuestro sistema.

El Trazador de Pistas se desarrollará siguiendo el modelo de Sistemas de Recomendación Basados en Contenido, cuyas características principales se explican en el apartado siguiente. Por lo tanto, la información almacenada será dotada de pesos, que irán actualizando su valor conforme el usuario vaya seleccionando unas páginas web u otras, con el fin de mantener actualizadas las preferencias del usuario, dando más valor a contenidos más actuales frente a contenidos anteriores ya almacenados.

Cabe matizar que si bien se tratará de una aplicación que parte de un Sistema de Recomendación Basado en Contenido, pues la información se obtendrá a partir de los sitios web indicados por el usuario, se pretende que la aplicación pueda ser configurable en ciertos aspectos de la misma si así se desea, como en el número de URLs similares que la aplicación le debe devolver, o si se deben tener en cuenta los caracteres numéricos y los acentos en las palabras. De esta manera se dota a la aplicación de un añadido extra que permite que la herramienta se utilice además de para su propósito principal, facilitar la navegación por la Red, para realizar diversos análisis de tipo semántico sobre el contenido en las páginas Web y las búsquedas en la Red mediante buscadores.

1.3 Trazador de Pistas o Trailblazer (3)

Un trazador de pistas o trailblazer es un mecanismo que facilita la navegación a través de un documento digital, bien sea un documento stand-alone alojado en una máquina, bien sea un documento web alojado en un servidor de Internet, ofreciéndole posibles enlaces de interés mientras el usuario navega por el documento buscando información.

Su origen surge a raíz la aparición del ***hipertexto***, que son modelos de texto digitales donde la información puede ser consultada de manera no secuencial (4), concepto creado por Vannevar Bush en 1945 con la invención de *Memex*, un dispositivo que servía como base de datos y que posteriormente daba la posibilidad de interactuar con los usuarios, permitiendo mecanizar y conectar la información con el fin de aumentar el desarrollo en aquella época. Aunque Memex no llegó nunca a materializarse, el concepto de enlazar información mediante asociaciones o sendas (trails en inglés), tal y como funciona el cerebro humano, es el germen de los posteriores trazadores de pistas. (5)

Con la expansión de los hipertextos, que proporcionan acceso a cantidades elevadas de información, que además no está ordenada de manera secuencial, surge el desafío de encontrar fórmulas que ayuden al usuario en la navegación por los mismos. Algunos de los métodos utilizados son el botón de vuelta atrás, el fichero de histórico, los marcadores favoritos o la visita guiada.

La ***visita guiada*** es el primer prototipo de trazador de pistas desarrollado, y consiste en una colección de enlaces que el autor del hipertexto proporciona al lector para marcar un camino <<significativo>> a través del mismo. Fue diseñado en 1988 por R.H. Trigg, partiendo de la idea inicial de Vannevar Bush de trazar sendas de información, para el sistema *NoteCard*. Además, para facilitar a los autores la creación de visitas guiadas, propuso el uso de la herramienta *Tabletop*. Con este nuevo método, se consiguió una visión de conjunto de las unidades de información componentes del hipertexto, permitiendo organizar mejor el recorrido entre nodos del mismo. Sin embargo, aunque para el usuario este método representa la forma más simple de navegación, pues puede desentenderse completamente de cualquier tipo de navegación, dejándose llevar por la ruta que tiene marcada, las visitas guiadas presentan limitaciones, pues al ofrecer rutas primarias y secundarias ya establecidas, lo que hacen es devolver al lector un texto donde la información es de nuevo organizada de forma secuencial, limitando el acceso a la información, y haciendo necesario el uso de herramientas alternativas de navegación.

Con la evolución de los hipertextos y la aparición de internet y los motores de búsqueda, los trazadores de pistas han evolucionado, así como el acceso a la información también lo ha hecho. Internet ha supuesto un cambio en la forma de distribuir y tener acceso a la información, y los motores de búsqueda han ido evolucionando, de manera que cada vez intentan proporcionar al usuario resultados más personalizados, acordes con sus necesidades o preferencias. Orientar la recuperación de información hacia las verdaderas necesidades del usuario requiere enriquecer las expresiones de búsqueda con los elementos contextuales específicos del mismo. Para ello, es necesario recurrir trazadores de pistas, que son herramientas software capaces de interactuar con su entorno, es decir, a herramientas basadas en inteligencia artificial, que tienen capacidad de memorizar experiencias, llevando a cabo un aprendizaje con el que enriquecer el sistema que los está utilizando.

La secuencia de actuación de estas herramientas suele consistir en:

- Búsqueda de la información lanzando simultáneamente la misma pregunta a más motores de búsqueda.

- Filtrado y clasificación utilizando los criterios introducidos por el usuario o adquiridos de forma indirecta sobre él.
- Extrapolación de patrones de conducta y de necesidades.

Ejemplos de trazadores de pistas a este nivel son:

- Letizia: desarrollado por Henry Lieberman en 1995, es un agente localizado en el cliente que analiza las acciones del usuario mientras este navega por la red, sin solicitarle participación directa al mismo, y a través de la extracción de las palabras clave de los documentos que visita determina sus áreas de interés, construyendo así un perfil primario que permite consultar otros documentos próximos a los visitados y confeccionar una lista de documentos similares potencialmente interesantes, que evoluciona a medida que el usuario explora nuevos sitios.
- PowerScout: desarrollado por Lieberman y otros en 2001, mejora el modelo de Letizia. Aunque el perfil del usuario es elaborado también a partir de la extracción de las palabras clave de los documentos consultados, la generación de la lista de documentos similares se obtiene lanzando búsquedas a motores independientes. En este caso el usuario puede participar de forma activa en la definición de su perfil, estableciendo sus áreas de interés.
- WebWatcher: desarrollado por Robert Amstrong en 1995, actúa de manera similar a los agentes anteriores, pero se encuentra localizado en el servidor, por lo que también tiene en cuenta la experiencia de otros usuarios para elaborar los listados de preferencias. En contrapartida, no diferencia entre usuarios.
- MUSAG: desarrollado por C.V. Goldmand, A. Langer y J.S. Rosenchein en 1996, en lugar de utilizar palabras clave como los anteriores, basa sus búsquedas en unos conceptos que genera a partir de la comparación de las palabras empleadas en la formulación de la búsqueda con las que aparecen en los documentos recuperados. Los conceptos resultantes son empleados sucesivamente para expandir la búsqueda.

Posteriores desarrollos en este tipo de software lleva a los mismos hacia técnicas de búsqueda de relaciones entre los hechos y de creación de conocimientos nuevos, utilizando técnicas propias del tratamiento de datos estructurados pertenecientes al *data mining.* El data mining es una de las herramientas principales para la gestión del conocimiento y de los sistemas de toma de decisiones, que tiene como condición esencial la existencia de una base de datos relacional. Esta herramienta utiliza los agentes para la extracción de los datos, construcción de asociaciones, creación de árboles de decisiones, algoritmos genéticos, redes bayesianas, redes neuronales, mapas conceptuales y herramientas de visualización.

Sin embargo, cuando esto se intenta aplicar a entornos de datos textuales que apenas están estructurados, o nada estructurados, el data mining no es eficaz, por lo que se recurre a los llamados *text mining*, que utilizan técnicas de filtrado, procesamiento lingüístico y cálculo estadístico para descubrir conjuntos documentísticos.

Si el ámbito de aplicación es la Web, entonces se debe recurrir al *Web mining*, donde se analizan además los enlaces contenidos en los textos. Existen tres facetas:

- Web content mining: Describe los contenidos del cuerpo documental y la estructura interna de los documentos que lo componen. Si los datos no están estructurados, la representación se produce utilizando conjuntamente técnicas lingüísticas y estadísticas, mientras que si los datos están estructurados, se aplican técnicas estadísticas.

- Web structure mining: Se estudia la relación del documento con la Web a través del estudio de los enlaces que el cuerpo del documento presenta. De esta manera se permite analizar la co-citación, la colaboración y el intercambio de información entre entidades.
- Web usage mining: Estudia la manera en que el usuario interactúa con la Web, basándose en la creación de un perfil de usuario, a través de sus preferencias o hábitos de navegación.

A partir del Web mining surge el *adaptive hypertext,* creado por Peter Brusilowsky en 1998, cuyo objetivo es adaptar el hipertexto y construir un modelo para el usuario individual, centrándose en la recuperación de información, la información y la ayuda en línea, la educación y la personalización del espacio informativo. Esta personalización se plantea a nivel de contenido (el contenido presente en el nodo es el que se adapta al perfil del usuario) o enlaces (se modifican los enlaces).

En este caso, las ayudas a la navegación se pueden dividir de la siguiente manera:

- Guía global: se indica al usuario la vía más corta para alcanzar la información de su interés. Esto se puede llevar a cabo mediante visita guiada u ordenando los enlaces que aparecen en el nodo según la relevancia que tienen en función de los intereses del usuario. A este perfil corresponden los perfiles mencionados anteriormente WebWatcher y Letizia, que recuperan para los usuarios el concepto de visita guiada.
- Guía local: se indica al usuario el enlace más relevante dentro de una página. Es un recurso muy sencillo, por ejemplo mediante un índice al inicio de la página que reenvía directamente a sus secciones.
- Soporte para la orientación local: puede ser proporcionado a través de una información adicional acerca del destino de los enlaces, recurriendo a anotaciones (por ejemplo modificando el color de los enlaces visitados), u ocultando enlaces considerados poco relevantes.
- Soporte para la orientación global: trata de ofrecer al usuario las herramientas para que tenga una visión global de la red y comprenda su situación en ella. Como en el caso del soporte para la orientación local también se recurre a las técnicas de ocultación y anotación, si bien en este caso la ocultación proporciona al usuario una visión gradual de la red, y las anotaciones se utilizan a modo de señalización, para que el usuario pueda saber en todo momento dónde se ubica respecto al lugar de partida y a los nodos intermedios.

Como se observa, los trazadores de pistas han sufrido una evolución a medida que los hipertextos, la Web, y la inteligencia artificial se han desarrollado, convirtiéndolos de meras guías a la navegación en mecanismos complejos que analizan no sólo los datos obtenidos, sino también en el entorno y el usuario con el que están trabajando.

Un ejemplo de un trazador de pistas más potente es el *proyecto Trailblazer* que fue llevado a cabo por la Agencia Nacional de Seguridad (NSA) de Estados Unidos entre los años 1999 y 2007, si bien el objetivo principal de este proyecto no consistía en una ayuda a la navegación de usuario, sino en la recopilación masiva y posterior análisis de datos susceptibles de ser relevantes para la seguridad nacional de Estados Unidos, provenientes de redes, como Internet. Para ello, este proyecto pretendía rastrear entidades a través de teléfonos móviles y correos electrónicos. (6)

El mecanismo del Trailblazer consistía en ordenar los 2 millones de bits de datos que la NSA recoge cada hora, para identificar patrones, palabras clave y enlaces a otros datos. El programa, en teoría, podría traducir toda la información en texto sin formato o datos de voz, ana-

lizar los resultados para identificar temas de interés, almacenar los resultados en una base de datos fácilmente investigable y reenviar elementos seleccionados a los analistas adecuados para el seguimiento.

Esta tarea se dificultaba más de lo suficientemente compleja que era por la aparición constante de nuevos tipos de teléfonos inalámbricos, Internet, teléfonos móviles, y el auge de las comunicaciones de mensajería.

Con este proyecto, la NSA pretendía estar un paso por delante en el avance de las tecnologías, sin embargo, y a pesar de la gran suma de dinero invertida, en torno a 1.2 billones de dólares, y el esfuerzo realizado, no se consiguió lograr el objetivo: no se consiguió diseñar un sistema que fuera capaz de procesar toda la información que la NSA recoge, y el proyecto tuvo que ser cancelado, dando lugar al "mayor despilfarro en la comunidad de inteligencia". (7)

1.4 Sistemas de Recomendación (8) (9) (10)

Los Sistemas de Recomendación son una subclase de Sistemas de Filtrado de Información, que pretenden predecir las preferencias de un usuario en torno a un determinado objeto, que todavía no ha sido tomado en consideración por el mismo, teniendo como base las preferencias mostradas por dicho usuario con respecto a otro objeto de características similares.

Aunque el objetivo principal de la investigación sobre los sistemas de recomendación recae en la búsqueda de los algoritmos de recomendación más precisos, hay un número de factores que también son importantes y deben ser tenidos en cuenta:

- *Diversidad*: Los usuarios tienden a estar más satisfechos con recomendaciones cuando hay una mayor diversidad dentro de la lista.
- *Persistencia en el Recomendador*: En algunos casos, es más eficaz volver a mostrar recomendaciones, o permitir a los usuarios volver a los tipos de elementos que mostrar nuevos artículos. Hay varias razones para esto. Los usuarios pueden pasar por alto los elementos cuando se les muestra por primera vez, por ejemplo, porque no tenían tiempo para inspeccionar las recomendaciones cuidadosamente.
- *Privacidad*: Por lo general, los sistemas de recomendación tienen que hacer frente a problemas de privacidad, ya que los usuarios tienen que revelar información sensible. Por ejemplo, crear perfiles de los usuarios del edificio utilizando filtrado colaborativo puede ser problemáticos desde el punto de vista de la privacidad. Muchos países, sobre todo europeos, tienen una fuerte cultura de la privacidad de datos y los intentos por introducir cualquier nivel de perfil de usuario puede dar lugar a una respuesta negativa por parte de los usuarios. Se ha realizado muchas investigaciones con respecto a este tema y se ha encontrado que la combinación de puntos débiles (una conexión inesperada que proporciona recomendaciones casuales) y otras fuentes de datos se puede utilizar para descubrir las identidades de los usuarios en un conjunto de datos anónimos.
- *Datos demográficos de los usuarios*: La demografía de los usuarios pueden influir en cómo los usuarios están satisfechos con las recomendaciones. Existen estudios que demuestran que los usuarios de edad avanzada tienden a estar más interesados en las recomendaciones de los usuarios más jóvenes.
- *Robustez del sistema:* Cuando los usuarios pueden participar en el sistema de recomendación, la cuestión del fraude debe tenerse en cuenta.
- *Hallazgo por fortuna*: Esto es "lo sorprendente que las recomendaciones son". Por ejemplo, un sistema de recomendación que recomienda leche a un cliente

en supermercado, podría ser perfectamente exacto, pero no es una buena recomendación, ya que es un tema obvio para el cliente a comprar.

- *Confianza*: Un sistema de recomendación es de poco valor para un usuario si el usuario no confía en el sistema. La confianza puede ser construida en un sistema de recomendación explicando cómo genera las recomendaciones, y por qué se recomienda un artículo.
- *Etiquetado*: La satisfacción de los usuarios con las recomendaciones puede estar influenciada por el etiquetado de las recomendaciones. Por ejemplo, existen estudios donde se demuestra que recomendaciones etiquetadas como "patrocinados" tienden a provocar menos satisfacción en los usuarios que recomendaciones idénticas etiquetadas como "orgánicos", mientras que las recomendaciones sin ninguna etiqueta son las más valoradas según este estudio.

Los Sistemas de Recomendación se pueden clasificar en dos tipos, atendiendo a la forma de extraer la información proveniente del usuario: Filtro Colaborador y Filtro Basado en Contenido.

Antes de profundizar en los dos tipos existentes de Sistemas de Recomendación, es necesario presentar los siguientes conceptos a tener en cuenta:

- **Matriz de utilidad:** se llama así a la matriz deseada que representan los pares usuario-elemento con la puntuación dada por el usuario. Se asume que la matriz es escasa, es decir, que la mayoría de los registros son desconocidos porque no se tiene información explícita del usuario sobre el elemento.
  El objetivo de un Sistema de Recomendación es predecir todos los espacios en blanco en dicha matriz, teniendo en cuenta tanto las preferencias del usuario en elementos similares como las puntuaciones de otros usuarios a los elementos similares (el elemento que se evalúa y los demás objetos parecidos). Cabe destacar que no es necesario predecir todos los espacios en blanco, sino sólo una muestra suficientemente grande de los elementos mejor puntuados.
  En el siguiente ejemplo podemos observar una matriz de utilidad donde seis usuarios diferentes (U1, U2, U3, U4, U5 y U6) han puntuados doce objetos distintos (I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11 e I12). Los puntos representan los espacios en blanco mencionados anteriormente, esto es, los objetos que los usuarios no han calificado.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
U1	1	2	•	•	2	•	3	4	•	4	1	•
U2	•	•	1	5	•	5	3	1	•	5	2	1
U3	1	•	•	2	•	1	•	3	4	•	•	•
U4	•	1	4	4	•	•	3	•	5	4	•	1
U5	2	•	5	•	1	•	1	•	•	•	2	1
U6	•	•	5	2	1	•	•	4	•	1	•	2

Figura 3- Ejemplo matriz utilidad.

- **Fenómeno de la cola larga:** se denomina así a la representación de la distinción entre el mundo físico y el mundo online. Mientras que en el mundo real el acceso a los elementos están limitados por cuestiones de espacio para almacenar todos los elementos, en el mundo online el usuario no tiene límites para acceder a la información que busca.

Figura 4- Fenómeno Cola Larga (11)

En el gráfico anterior, el eje vertical representa la popularidad del objeto, mientras que el horizontal muestra los elementos ordenados por popularidad.
Los elementos a la izquierda de la línea vertical, cuya popularidad se ha coloreado en color rojo, representan los elementos populares en un entorno físico, mientras que los elementos populares online representan la totalidad de la curva.
Esto fuerza a los mecanismos online a recomendar elementos a usuarios individuales. Ni pueden presentar los elementos como en el mundo físico, ni los usuarios pueden tener conocimiento de todos los elementos disponibles.

Para ambos tipos de filtro, es necesario conocer técnicas para medir la similitud de dos objetos. Se pueden utilizar las medidas naturales de distancia siguientes:

1. **Distancia de Jaccard**:

$$d(x,y) = 1 - SIM(x,y)$$

donde

$$SIM(x,y) = \frac{|x \cap y|}{|x \cup y|}$$

2. **Distancia del coseno**:

$$\mathrm{d}(\cos(x,y)) = \arccos\left(\frac{x \cdot y}{|x| \cdot |y|}\right)$$

Estas distancias serán utilizadas tanto para Filtros Colaboradores como para Filtros Basados en Contenido.

1.4.1 Filtro Basado en Contenido

Este tipo de filtro construye su modelo de datos analizando las características concretas del objeto elegido por el usuario, y recomendando aquellos objetos con características similares, es decir, se basan en objetos que el usuario eligió en ocasiones anteriores para realizar sus nuevas recomendaciones.

La principal ventaja en este tipo de sistemas radica en que se basa en la propia experiencia del usuario para dar mayor peso a un atributo determinado de un objeto, y poder así proporcionar al usuario información más aproximada a sus intereses. Sin embargo, estos Sistemas tienen como reto ser capaces de aprender de los objetos que el usuario valora para poder ofrecer objetos parecidos en otras áreas, y no limitarse a una sola área de recomendación.

Ejemplos de páginas web que utilizan este sistema son Youtube o Pandora Radio.

En este caso, la extracción de información se realiza analizando los objetos y dando valor o peso a la información extraída, por lo que se hace necesario construir un ***perfil*** para cada elemento, que consiste en una entrada o una colección de entradas que representan características fácilmente apreciables del objeto.

Por ejemplo, en un Sistema donde se recomiendan películas, el perfil puede ser una lista que contenga características de la película tales como actores, director, año de rodaje, género, etc.

También es posible incluir elementos al perfil provenientes de datos introducidos a mano, como descripciones realizadas por el fabricante dando características relevantes del objeto, por ejemplo, el tamaño de la pantalla de una tablet.

Algunos otros elementos con características fácilmente identificables son libros (autor, género, año de lanzamiento), música (compositor, género, año), teléfonos móviles (tamaño, especificaciones técnicas, color, marca), aparatos electrónicos (especificaciones técnicas, tamaño, marca), etc.,

Sin embargo, existen otras clases de objetos cuyas características principales no se pueden identificar tan fácilmente, como son documentos de texto e imágenes.

A pesar de existir numerosos tipos de documentos para los que los Sistemas de Recomendación pueden ser muy útiles, como artículos publicados, blogs, o recetarios, por ejemplo, resulta complicado distinguir entre temas, porque algunos de estos documentos no suelen tener información extra asociada que proporcione características del texto. Se hace necesario por tanto un análisis del texto.

En primer lugar, se deben eliminar las palabras llamadas *stop words,* que son las cientos de palabras más comunes utilizadas en el lenguaje (pronombres, artículos, etc.) y que proporcionan poca información sobre el tema del documento en sí. Para el resto de palabras que se conservan, es necesario contabilizar el número de veces que aparecen en el texto. De esta manera, aquéllas que tengan puntuaciones más altas, es decir, que aparezcan más veces en el documento, serán las palabras que caracterizan el documento.

Entonces, para todo el conjunto de documentos analizado, es posible tomar las *n* palabras con mayor puntuación en cada uno de ellos como su perfil, donde *n* puede ser un número fijo para todos los documentos, o un porcentaje en función del número de palabras que contiene el documento.

Ahora bien, es necesario además medir la similitud de las palabras seleccionadas entre documentos para poder determinar aquéllos que tratan sobre los mismos temas. Para medir la similitud de dos documentos se pueden utilizar las medidas de distancia mencionadas anteriormente:

1. Distancia de Jaccard entre conjunto de palabras:

   En este caso, la distancia de Jaccard es 1 menos la relación entre los tamaños de la intersección y la unión de los conjuntos de x e y, tomando como x e y a los conjuntos resultantes con las puntuaciones más altas de palabras.

2. Distancia del coseno entre conjunto de palabras tratados como vectores:

   En este caso se debe pensar en el conjunto de palabras que más aparece en el texto como en un vector, con un componente para cada palabra, de manera que el vector tendrá un 1 si la palabra está en el conjunto total de palabras y un 0 en caso contrario.

Otra manera más sencilla de clasificar un texto (y una imagen) es mediante ***etiquetas***. Este es un sistema que se puede utilizar en bases de datos o material online. De esta manera se asocia una serie de palabras que describen al objeto con el mismo, eliminando la necesidad de realizar un análisis del texto completo, en el caso de los documentos. Sin embargo, presen-

ta la desventaja de que es el propio usuario el que tiene que estar dispuesto a etiquetar el objeto que está visitando, introduciendo palabras o una frase que lo describa, y por tanto, el Sistema sólo funciona si el usuario está dispuesto a colaborar.

En un Sistema de Recomendación Basado en Contenido, también es necesario elaborar un ***perfil de usuario***, donde quedan reflejadas las preferencias establecidas por el usuario.

De esta manera, con los perfiles del objeto y del usuario, se puede estimar el grado de aceptación de un usuario a un usuario, calculando la distancia del coseno entre el vector resultante del usuario y el del objeto.

Una manera diferente de enfrentarse a un Sistema de Recomendación Basado en Contenido, consiste en tratar el problema como si de una ***máquina de aprendizaje*** se tratara, utilizando la matriz de utilidad y los perfiles de los objetos. Esta aproximación consiste en tratar los datos obtenidos como si fuera un conjunto de datos de aprendizaje y, para cada usuario, construir un clasificador diferente que prediga la puntuación de todos los objetos.

Un clasificador bastante común es el llamado ***árbol de decisión***, que consta de una colección de nodos ordenados como un árbol binario. Cada *hoja* del árbol contiene una decisión, en este caso si el objeto le gusta o no al usuario, y cada nodo interior es una condición del objeto que se está clasificando, en este caso, una o varias características del mismo.

Para clasificar un objeto, se empieza por el nodo raíz, y se analizan las características del mismo. Si alguna de las características es una de las deseadas, se avanza hasta el nodo *hijo* izquierdo, y en caso contrario, se avanza hasta el nodo *hijo* derecho. Este proceso se repite hasta que se alcanza una *hoja* del árbol y es esa hoja la que determina si al usuario le gusta el objeto o no.

Construir un árbol de decisión implica analizar las características de cada nodo interior, tratando de ordenar los nodos de manera que se dividan los elementos en dos grupos: aquéllos que contienen la característica buscada y aquéllos que no la tienen. Esta acción se repetirá hasta que el grupo sea homogéneo, creando una hoja.

La desventaja que ofrece este método es que requiere un tiempo largo para construirse, pues es necesario crear un árbol de decisión para cada usuario, y, además, es necesario evaluar todas las características de cada elemento para construir diferentes combinaciones, que pueden resultar complejas para el análisis y toma de decisiones.

Por ello, este método suele ser usado solo para sistemas de un tamaño relativamente pequeño.

1.4.2 Filtro Colaborador

Este tipo de filtro construye el modelo de datos coleccionando y analizando información proveniente del usuario, tal como gustos, preferencias y selecciones anteriores para ofrecerle objetos que han sido valorados positivamente por otros usuarios que comparten características similares.

En este caso, la extracción de la información puede ser explícita, por ejemplo mediante formularios que el usuario debe rellenar o puntuación dada al objeto, o implícita, por ejemplo realizando estadísticas sobre las veces que el objeto ha sido visitado por el usuario.

Este tipo de sistemas parten con la ventaja de que no es la máquina la que tiene que analizar el objeto propiamente dicho obteniendo sus características, sino simplemente analizar las características ya dadas. Por el contrario cuenta con tres desventajas principales: Comienzo Frío (necesidad de gran cantidad de información para proporcionar resultados óptimos), Escalabilidad (millones de usuarios y productos a ser analizados y almacenados) y Escasez (pocos objetos evaluados por los usuarios en el total disponible en las bases de datos).

Ejemplos de páginas web que utilizan este sistema son Facebook, Last.fm, FilmAffinity.

Un Filtro Colaborador no utiliza las características de los objetos para determinar su similitud, sino que determina si dos objetos son similares basándose en las puntuaciones que el usuario ha concedido a dichos elementos. Por tanto, para elaborar este tipo de filtro, nos basta con utilizar la matriz de utilidad, dejando de lado tanto el perfil del objeto como el perfil del usuario. El vector asociado al elemento se obtendrá de su propia columna en la matriz de utilidad, y el usuario quedará representado con su propia fila de la matriz.

Para determinar si dos usuarios tienen preferencias similares, será necesario medir su similitud tomando como referencia sus filas en la matriz. Estas filas compondrán sus respectivos vectores, y será necesario medir la distancia entre ellos, mediante la distancia de Jaccard o la distancia del coseno, para determinar si sus gustos son similares o no.

Determinar si dos usuarios o dos objetos son similares o no a través de matriz de utilidad se puede realizar mediante los siguientes métodos:

1. Distancia de Jaccard: Este método resulta útil si la matriz de utilidad está rellena sólo con valores sencillos, por ejemplo, valores de 1 donde el elemento ha sido visualizado y vacío en el resto. Sin embargo, para matrices donde se almacena información más precisa, por ejemplo las puntuaciones de cada usuario para cada elemento, este método pierde información relevante.
2. Distancia del coseno: Este método se podría utilizar tomando los espacios en blanco como 0. Sin embargo, en este caso trataría los elementos no puntuados por el usuario como elementos con puntuación baja por el usuario, cuando no han sido evaluados. Podría por tanto llevar a conclusiones erróneas sobre la cercanía de dos usuarios.

Para intentar eliminar las limitaciones que estas técnicas presentan, se aplican además los siguientes métodos:

- *Redondeo de datos*: Este método trata de eliminar la aparente similitud entre los elementos que el usuario ha puntuado con puntuaciones más elevadas y los que ha puntuado con puntuaciones más bajas. Para ello, redondea los valores más altos a un mismo valor, y los más bajos a otro valor diferente. Por ejemplo, si un usuario ha otorgado puntuaciones a elementos con valores comprendidos entre 1 y 5, a los valores 1 y 2 se les otorga el valor nulo, y a los valores 3, 4 y 5 se les otorga el valor 1. De esta manera, al aplicar la distancia de Jaccard o la distancia del coseno, se obtienen valores de similitud entre usuarios más precisos.
- *Normalización de puntuaciones*: Este método consiste en normalizar cada puntuación que el usuario ha dado a cada elemento. Esta normalización se realiza restando a cada puntuación la media aritmética de todas las puntuaciones del usuario. Así, las calificaciones más bajas se vuelven valores negativos, mientras que las calificaciones más altas serán valores positivos. Aplicando después la distancia del coseno, se obtendrá que los usuarios que proporcionaron puntuaciones muy diferentes a un mismo elemento tendrán vectores casi en direcciones contrarias, mientras que los usuarios que calificaron objetos de manera similar, obtienen vectores con un ángulo relativamente pequeño entre ellos.

Generalmente es mejor normalizar la matriz de utilidad primero, pues de esta forma se ajustan las puntuaciones en caso de que un usuario tienda a dar calificaciones muy elevadas o muy bajas. Hay que tener en cuenta también que es necesario estimar más de una entrada en la matriz de utilidad para poder calcular la similitud entre dos usuarios o dos objetos. Esto quiere decir que para recomendar elementos a un usuario, necesitamos estimar cada registro en la fila de la matriz para dicho usuario, o por lo menos, encontrar la mayoría de valores en esa fila que están en blanco pero de los que poseemos un valor alto de estimación.

Si el proceso que se realiza es encontrar usuarios similares, sólo será necesario estimar los valores en blanco del usuario una vez, pues se pueden esos valores a partir del conjunto de usuarios similares. Si el proceso que se realiza es encontrar objetos similares, tenemos que calcular elementos similares para cada casi todos los objetos antes de estimar la fila para el usuario en cuestión.

Sin embargo, calcular la similitud entre objetos a menudo ofrece más información fiable, debido a que es más fácil encontrar elementos que pertenezcan al mismo género que encontrar usuarios a los que solamente les guste objetos de un mismo género.

En cualquier caso, para cualquiera de los dos enfoques, se debería pre calcular los elementos mejor puntuados para cada usuario antes de que se necesite tomar esa decisión. Dado que la matriz de utilidad evoluciona despacio, generalmente es suficiente con calcularla frecuentemente y asumir que permanece fija entre recálculos.

Como se observa, calcular similitudes entre usuarios o entre objetos es complicado, porque se tiene poca información sobre los pares usuario-objeto en la matriz de utilidad, dado que ésta está escasamente rellena.

Una forma de abordar esta limitación consiste en agrupar elementos y/o usuarios. Para se utilizaría alguno de los métodos para calcular distancias que se han mencionado anteriormente, agrupando los elementos. Por ejemplo, en una base de datos de películas, se agruparían todas las películas pertenecientes a una misma saga en un mismo elemento, calculando la distancia entre todas las puntuaciones que el usuario ha dado a cada película. Dado que este paso se puede repetir cuantas veces sea necesario, resulta conveniente realizar este proceso jerárquicamente y en tantos pasos como sea necesario, en lugar de reducir la matriz a un número pequeño de elementos inmediatamente.

1.4.3 Sistemas Híbridos

Existe un tercer tipo que combina las características de ambos filtros, el llamado ***Sistema Híbrido***. Este sistema se puede implementar de diversas maneras, bien utilizando cada filtro anterior por separado y combinando, los resultados, bien añadiendo ciertas características de un modelo al otro, o bien unificando ambos modelos en uno solo.

Existen siete técnicas de hibridación:

- Ponderado: La puntuación de los diferentes sistemas de recomendación que forman el sistema híbrido se combinan de forma numérica.
- Cambio: El sistema elige entre los sistemas de recomendación y se aplica el seleccionado.
- Mixtos: Las recomendaciones de diferentes sistemas recomendadores se presentan juntos.
- Combinación de características: Características derivadas de diferentes fuentes de conocimiento se combinan y dan a un único algoritmo de recomendación.
- Característica de Aumento: Se utiliza una técnica de recomendación para calcular una característica o conjunto de características, que es a su vez parte de la entrada a la siguiente técnica.
- Cascada: Se otorga a los sistemas de recomendación una prioridad estricta, con los de menor prioridad rompiendo los lazos en la puntuación de los superiores.
- Meta-nivel: Se aplica una técnica recomendación que produce algún tipo de modelo, que a su vez es la entrada utilizado por la técnica siguiente

Ejemplos de utilización son Amazon y Netfix.

2 Trazador de pistas

2.1 Descripción de las tareas

Para alcanzar el objetivo de este proyecto, se plantean inicialmente tres bloques de trabajo diferenciados entre sí que pueden ser abordados de manera independiente y paralela:

- Obtención del contenido HTML de la página web que se está visitando: se plantea inicialmente la obtención de todo el contenido HTML a través de la propia aplicación para proceder a su posterior parseo, aunque finalmente, como se explicará en los siguientes apartados, la aplicación obtendrá directamente la URL introducida por el usuario, y se obtendrá el texto del HTML asociado utilizando la herramienta JSOUP.
  Será necesario por tanto realizar las siguientes tareas:
  - Obtención de la URL introducida manualmente por el usuario a través de la aplicación.
  - Conexión con la URL y extracción de todas las palabras catalogas como texto en el código HTML.
  - Almacenaje de todas las palabras obtenidas.

- Parseo del contenido HTML: Análisis de dicho contenido y obtención de las palabras más relevantes a través de un algoritmo que otorgue peso a las palabras en función por un lado, del número de veces que aparece dentro del HTML, y por otro lado de su antigüedad, de manera que las más recientes sean las que obtengan mayor peso.
  Esta tarea se divide en los siguientes pasos:
  - Análisis del total de palabras extraídas y recuento para almacenar cada palabra asociada al número de veces que se repite.
  - Limpieza semántica del contenido para eliminar palabras que no aporten valor, como pronombres, artículos, preposiciones, adverbios, etc. Adicionalmente, limpieza de palabras que no aparezcan en los diccionarios precargados por la aplicación y limpieza de caracteres numéricos, en función de la configuración establecida por el usuario.
  - Aplicación de algoritmo para obtener las palabras más repetidas (el número puede ser mayor si hay palabras con igual peso siempre que hayan aparecido más de una vez) y sumarlas a las palabras ya almacenadas en el sistema que fueron analizadas anteriormente, dando mayor peso a las palabras nuevas que se incorporan a la lista.
  - Obtención de las palabras más repetidas en el conjunto total, y almacenaje de las palabras más repetidas.

- Obtención de URLs similares a las visitadas por el usuario. Una vez se obtienen las 5 palabras más relevantes del sistema, se lanzan al buscador elegido y se procesan las URL que el buscador devuelve como resultado de la búsqueda, las primeras URLs al usuario que no enlacen con documentos Word ni PDF.
  En este apartado es necesario realizar las siguientes tareas:
  - Componer la URL de consulta al buscador elegido, en este caso Trovator, adaptándola a la sintáctica específica del navegador.

- Lanzar la consulta, procesar la respuesta para obtener únicamente las URLs de interés y almacenar las primeras direcciones en orden de respuesta del buscador.
- Mostrar al usuario las URLs en orden. El orden variará en función de las preferencias del usuario, que puede indicar el número de enlaces a mostrar.

2.2 Herramientas y tecnologías analizadas

Para llevar a cabo todas las tareas expuestas en el apartado anterior y desarrollar la funcionalidad completa de la aplicación, se deben elegir qué tecnologías conviene utilizar para facilitar la labor.

Antes de obtener las tecnologías utilizadas finalmente, se han analizado diferentes soluciones, con el fin de determinar qué herramientas se adaptan mejor a las necesidades del proyecto.

En este apartado se presentan las tecnologías que han sido descartadas después de su análisis, bien porque se hayan encontrado herramientas que cumplan su función más eficazmente, bien porque su análisis tiene por objeto únicamente proporcionar una visión global del sistema.

2.2.1 Herramientas para la obtención de código HTML

La parte inicial del proceso de este trazador de pistas consiste en conocer qué páginas Web está visitando el usuario y poder descargar el código HTML para evaluarlo después.

Es en este apartado donde más tecnologías se han analizado, pues es la tarea más compleja de llevar a cabo, dado que se requiere que el sistema sea capaz de obtener de manera dinámica y en tiempo real la URL primero y el código fuente, o HTML, de dicha URL después.

Dado que el interés principal de este proyecto consiste en el análisis del contenido del código HTML, será el usuario el que indique a la aplicación la URL en la que está interesado, introduciéndola a mano. Sin embargo, se han analizado además diversas soluciones que fluctúan desde instalar un plugin de Mozilla a escuchar el puerto 80 del equipo que el usuario utiliza, con el fin de tener una perspectiva más amplia sobre el proceso completo.

2.2.1.1 Plug-in Mozilla (Gecko Plug-in)

Una de las posibilidades existentes para obtener las URLs que el usuario visita en tiempo real es el desarrollo e implantación de un plugin en el navegador, de manera que sea capaz de proporcionar directamente las URLs al trazador, o incluso almacenar el código HTML en alguna localización específica indicada, donde el sistema pueda acceder después.

Al inicio de este proyecto, el navegador que ofrece esta posibilidad es Mozilla Firefox, dado que dispone de un área abierta para desarrolladores, donde se proporciona APIS para elaborar programas e integrarlos en el navegador, por lo que se analiza el API para plug-in disponible, denominado Gecko. (12)

Un plug-in es un módulo de código dinámico, cuyas fuentes deben seguir la sintaxis del lenguaje de programación C, que está asociado con uno o más tipos MIME. Su ciclo de vida está controlado por la página web que lo llama. Cuando el navegador comienza la sesión, busca todos los plug-in disponibles en sitios concretos del sistema. Por lo tanto, también es dependiente de la plataforma donde se esté ejecutando (Windows, Unix o MAC).

Cuando el usuario abre una página que contiene datos embebidos de un tipo de media que invoca un plug-in, el navegador responde con la siguiente secuencia de acciones:

- Comprobar si hay un plug-in con un tipo MIME correspondiente
- Cargar el código de plug-in en la memoria
- Inicializar el plug-in
- Crear una nueva instancia del plug-in

Es posible cargar varias instancias del mismo plug-in en una sola página, o en varias ventanas abiertas al mismo tiempo. Por ejemplo, si se está navegando por una página que tiene varios clips de RealAudio incrustados, el navegador va a crear tantas instancias del complemento RealPlayer como sea necesario. Cuando el usuario abandona la página o cierra la ventana, se elimina la instancia de plug-in. Cuando se elimina la última instancia de un plug-in, el código de plug-in se descarga de memoria. Un plug-in no consume más recursos que espacio en disco cuando no está cargado.

El API ofrece 3 tipos de estructuras:

- Métodos propios del plug-in, que son funciones que implementa el desarrollador y que Gecko llama. El nombre de estas funciones debe comenzar por NPP_, por ejemplo NPP_New.
- Métodos propios del navegador que son funciones implementadas por Gecko y que el plug-in llama. El nombre de estas funciones debe comenzar por NPN_, por ejemplo NPN_Write.
- Estructuras de datos, que son tipos de plug-in específicos definidos en el API. Deben comenzar por NP, por ejemplo NPWindow.

Un plug-in puede ser mostrado en el navegador, o ejecutar en segundo plano sin que el usuario tenga visibilidad sobre él. Existen tres posibilidades al respecto: que el plug-in esté embebido en una página web y visible, que esté embebido en una página web y escondido, o que se muestre como página web entera en su propia ventana.

Para lograr esto, es posible combinar el código del plug-in con elementos HTML.

Los pasos necesarios para implementar un plug-in son:

- Planificar el plugin: decidir qué servicios debe proporcionar el plugin y cómo va a interactuar con el navegador, para qué se va a crear.
- Configurar el entorno de desarrollo correctamente: Es necesario asegurarse que se dispone de los archivos necesarios desde el SDK de plugins.
- Crear un proyecto de plug-in.
- Escribir el código del plugin con los métodos adecuados del API.
- Construir el plug-in según el Sistema Operativo en el que se va a implantar.
- Instalar el plug-in en el directorio de plug-in para el sistema operativo
- Probar el plug-in y depurar si es necesario.
- Crear una página HTML e incrustar el objeto plug-in.

Como contrapartida, utilizar un Plug-in de un navegador específico, implica restringir el uso del Sistema de Recomendación a un solo navegador.

2.2.1.2 Extensión Mozilla

Una extensión es un complemento que añade nuevas funcionalidades al navegador, desde un botón para la barra de herramientas, hasta características nuevas. De esta manera se puede personalizar el navegador para ajustarlo a las necesidades del usuario, sin aumentar el tamaño del mismo significativamente.

Una extensión difiere de un plug-in en que éste último simplemente ayuda a mostrar al navegador contenido específico, como la reproducción de archivos multimedia. (13)

Las extensiones están programadas en lenguaje XUL (*XML-based User-interface Language*) y tienen una estructura específica de carpetas y archivos, que está contenido en un fichero *.xpi*. Los elementos que debe contener una extensión son:

- /install.rdf: El manifiesto de instalador. Es el archivo que le indica a la aplicación cómo debe ser instalada la extensión, y contiene información básica de la extensión (versión, autor, etc.) en forma de metadados. (14)
- /components/*
- /components/cmdline.js
- /defaults/: Permite definir los textos y mensajes que serán utilizados en la extensión.
- /defaults/preferences/*.js
- /plugins/*
- /chrome.manifest: Se declara aquí el tipo de material que se encuentra dentro de un paquete Chrome, el nombre del paquete y la ruta hacia el mismo. También se define cual es el archivo XUL que se va a fusionar con el archivo del navegador que viene por defecto con Firefox (browser.xul).
- /chrome/icons/default/*
- /chrome/: Esta carpeta se utiliza para cargar los archivos XUL del navegador. Aquí se sitúan los archivos correspondientes a la definición de la interfaz de la extensión (XUL), la lógica (js) y estilos (css) que esta utilice.
- /chrome/content/: Contiene todo el código de definición de interfaz (archivos XUL), y toda la lógica necesaria para implementar el comportamiento de la extensión, programado principalmente con lenguaje Javascript. (15)

Al igual que ocurre en el caso del plug-in, la creación e instalación de una extensión de Mozilla restringe el uso del Sistema de Recomendación a un solo navegador.

2.2.1.3 DDE /JNI

Dado que el navegador Internet Explorer está integrado en el Sistema Operativo Windows, se estudia también la posibilidad de conseguir de forma dinámica las URLs que el usuario visita a través del uso de ***DDE*** (Dynamic Data Exchange), que es un conjunto de librerías que permiten la comunicación entre varias aplicaciones en Microsoft Windows con el objetivo de compartir datos. DDE permite que una aplicación abra una sesión con otra aplicación, enviar comandos al servidor de aplicaciones y recibir respuestas. (16)

Las librerías DDE están escritas en lenguaje de programación C, por lo que, si el lenguaje de programación elegido para el Sistema de Recomendación es Java, resulta necesario utilizar una herramienta que permita conectar las librerías .dll propias de DDE con el resto de la aplicación.

Para ello, existe una librería llamada ***JNI*** (Java Native Interface), cuyo cometido es permitir que un programa escrito en Java ejecutado en la máquina virtual java (JVM) pueda interactuar con programas escritos en otros lenguajes. (17)

Sin embargo el uso de esta tecnología conlleva ciertos riesgos y limitaciones, entre otros: (18) (19)

- Un pequeño error al usar JNI pueden desestabilizar la Máquina Virtual de Java de una manera que resulte muy complicada para reproducir y depurar.
- Sólo las aplicaciones y applets firmados pueden invocar JNI.

- JNI no proporciona ninguna recolección automática de basura para los recursos de memoria que no pertenecen a la Máquina Virtual de Java que se ejecuta en el lado nativo, en nuestro caso las librerías .dll. Por tanto, deberían ser estas librerías las que asuman la responsabilidad de liberar explícitamente dichos recursos de memoria.

2.2.1.4 Información almacenada en PC (Historial/Archivos Temporales Internet)

Las navegaciones a través de un Navegador Web usualmente dejan rastro en el equipo donde se estén realizando, en forma de Cookies, Archivos Temporales de Internet e Historial de Navegación, por que se plantea la posibilidad de analizar estos ficheros para rastrear los lugares que el usuario visita. En concreto, los Archivos Temporales de Internet y el Historial de Navegación.

No obstante, el acceso a estos directorios plantea diversos problemas:

- Por un lado, según la instalación del navegador en el equipo, la ruta de estos directorios pueden variar, pudiendo haber varios directorios diferentes, si en un mismo equipo acceden varios usuarios con diferentes instalaciones, o dependiendo del idioma configurado en el sistema, el nombre de la ruta puede variar, por lo que su localización no es trivial.
- Por otro lado, atendiendo a motivos de confidencialidad de datos, los propios navegadores limitan y encriptan de alguna manera el contenido en estas carpetas, así, la información almacenada en el directorio, en unos casos no es completa, mientras que en otros debe ser descifrada.
  Además, en algunos navegadores, como Firefox, estas carpetas permanecen ocultas, de modo que su acceso a través de una aplicación que lea estos directorios no es posible por falta de permisos.
- Además, los propios usuarios pueden eliminar el contenido de estos directorios a través del propio navegador, o iniciar sesiones privadas de navegación, donde se limita el almacenaje de este tipo de información en el equipo, por lo que la aplicación no dispondría de datos que analizar.

2.2.1.5 Escucha Puerto 80 (HTTP)

Un puerto de red es una interfaz para comunicarse con un programa a través de una red. En concreto, el puerto 80 u 8080 es el que se utiliza para la comunicación con el protocolo ***HTTP*** (HyperText Transfer Protocol), que es el protocolo encargado de realizar las transacciones en la World Wide Web (20), por lo que una manera de extraer la información necesaria puede ser establecer un mecanismo de escucha de dicho puerto, y filtrando después el contenido para obtener la información de la URL que el usuario visita.

Aunque la escucha al puerto 80 se puede realizar mediante Sockets Java, el análisis posterior de la información extraída debe hacerse con alguna herramienta que sepa interpretar la encapsulación realizada por el protocolo HTTP para el envío de paquetes, y que los ensamble para extraer después la información adecuada.

Esto requiere utilizar librerías a muy bajo nivel, programadas con lenguaje Ensamblador o C.

En este caso, se plantea además una restricción adicional a nivel de seguridad, pues en función del equipo donde se instale, la escucha de estos puertos puede estar restringida por los administradores del sistema, para evitar una posible intrusión de programas espías y una fuga de datos sensibles o confidenciales. (21)

2.2.2 *Herramientas para la extracción de información del código HTML*

Para poder parsear y analizar el código HTML obtenido, es necesario primero estandarizarlo en formato XHTML. De esta manera se solventan los problemas de malformación de código y se simplifica la tarea de parseo del documento.

Además, es necesario recorrer después los nodos para poder extraer la información deseada.

Para esta tarea, se han analizado las herramientas JTidy y JDOM antes de elegir JSoup, tecnología que se expondrá más adelante, en el apartado 2.3.

2.2.2.1 JTidy

JTidy proporciona una librería Java para llevar a cabo dicha labor de estandarización y limpieza del código HTML a código XHTML. Para ello, suministra una librería .jar que se incluye en el proyecto para su utilización, sin necesidad de configuraciones adicionales, con una integración sencilla en el código de la aplicación. (22)

A pesar de que realiza una buena labor de parseado, se puede observar que, dado que la última actualización de la herramienta proviene del año 1999, la herramienta no es capaz de manejar algunas nuevas etiquetas, asociadas a interacciones con redes sociales. También se observa que el tiempo empleado en realizar el parseado con esta herramienta se eleva, ralentizando el conjunto de la aplicación.

Concretamente, tras testear la herramienta, se comprueba que, si bien es una herramienta que cumple su función en cuanto a transformar texto HTML en XHTML, presenta graves limitaciones en cuanto a adaptación a la versión actual de HTML (actualmente la versión utilizada es HTML 5) y velocidad de ejecución, realizando un parseado incompleto con páginas web que incluyen etiquetas pertenecientes a versiones más actuales de HTML, y elevando el tiempo de ejecución de las pruebas unitarias a valores de más de un minuto por procesado de página en algunos casos.

Estas pruebas se han realizados con páginas web de diferente procedencia:

- URLs de periódicos, donde se enlazan muchas noticias y existe una cantidad elevada de texto, perteneciente a cada noticia. Se han utilizado las direcciones siguientes: www.elpais.com, www.elmundo.com, www.marca.com
  En estos casos, el procesado de cada página resulta muy elevado, con valores superiores a los dos minutos en algunos casos, debido a la cantidad de información existente en la URL.
- URLs de blogs, donde existe una cantidad elevada de texto, y se enlaza a redes sociales. Las direcciones usadas son: www.webosfritos.es, www.unodedos.com, http://invitadoinvierno.com/
  En estos casos, el procesado presenta dificultades tanto a nivel de parseado, ya que existe información que no es capaz de analizar, como a nivel de tiempo de procesado, pues las entradas de algunos blogs contienen textos de tamaño más grande y el tiempo de parseo es superior al minuto.
- URLs de contenido académico. Se ha testeado esta tecnología con la página web de la Universidad Carlos III de Madrid: http://www.uc3m.es/Inicio, http://www.it.uc3m.es/
  En este caso, la herramienta responde correctamente, con tiempos de parseo algo elevados, pero mucho más cortos en comparación con los otros casos.

Por todo ello, esta herramienta se descarta a favor de JSoup.

2.2.2.2 JDOM

Un fichero HTML o XHTML es en realidad un fichero de tipo XML (Extensible Markup Language) con una serie de reglas o características específicas. A su vez, un fichero XML posee una estructura llamada de árbol, donde los elementos se dividen por nodos que contienen otros elementos.

Por lo tanto, para extraer información de estos elementos, se necesita alguna herramienta que sea capaz de recorrer estos nodos, navegando también entre los nodos hijos.

Existen dos aplicaciones estándar para ello, ***SAX*** (Simple API for XML) y ***DOM*** (Document Object Model), que difieren entre sí en la manera de recorrer el fichero para extraer la información y en el uso que hacen de la memoria del equipo, pues mientras SAX recorre y parsea el fichero por partes, DOM lo carga completo en memoria para recorrerlo y parsearlo entero después. Por ello, DOM utiliza mayor parte de la memoria disponible, aunque es más rápido, y está recomendado para archivos más pequeños, mientras que SAX utiliza menos memoria, pero su parseo resulta más lento. (23) (24)

Sin embargo, existe una herramienta que complementa y mejora ambos estándares, llamada ***JDOM***, que permite recorrer y parsear un archivo XML desde una aplicación basada en Java, pues es una herramienta pensada para ser utilizada en este lenguaje de programación.

JDOM recoge los mejores conceptos de APIs existentes y crea un nuevo conjunto de clases e interfaces que proporcionan una interfaz para el usuario más aproximada a lo que él espera.

Esta herramienta no es un parseador XML propiamente dicho, sino que es un modelo de objetos de documento (en inglés *Document Object Model*), que utiliza parseadores XML para construir documentos.

La filosofía de JDOM se basa en los siguientes conceptos:

- Debe ser sencillo para los programadores de Java. Dado que es una herramienta pensada y creada para ser utilizada con esta tecnología, la intención de JDOM es presentar sus librerías tal y como un programador Java esperaría. reduciéndole la carga de trabajo siempre que sea posible.
- Debe permitir la modificación de documentos fácil y eficiente.
- Debe ocultar las complejidades de XML siempre que sea posible, sin dejar de ser fiel a la especificación XML. Los usuarios no necesitan tener un amplio conocimiento sobre XML para desarrollar un código eficiente y correcto.
- Debe integrarse con DOM y SAX. Dado que son los estándares, JDOM puede leer a partir de fuentes de DOM y SAX existentes, y puede dar proporcionar ficheros a receptores DOM y SAX.
- Debe ser ligero y rápido. La carga y manipulación de los documentos debe ser rápida, y la carga de memoria lo más baja posible.
- Debería resolver el 80% (o más) de problemas Java/XML con el 20% (o menos) de esfuerzo. (25) (26)

Por todo ello, JDOM resulta una buena herramienta para recorrer un fichero XML con el fin de extraer información de él, y su uso está muy extendido.

Sin embargo, para este proyecto, se ha sustituido esta herramienta por JSOUP, que resulta una herramienta más completa, puesto que integra el parseo y la extracción directa de información, utilizando DOM en sus librerías, y que resulta más rápida y directa para el propósito de este proyecto.

2.2.3 Buscadores para la obtención de contenido similar al visitado

Una vez que se ha aplicado el algoritmo desarrollado y se han determinado cuáles son las palabras clave que conforman el interés actual del usuario, es necesario buscar contenido similar que ofrecerle.

Para ello, es necesaria la ayuda de un buscador online que realice una búsqueda basada en las palabras elegidas y devuelva a la aplicación un conjunto de URLs similares a la actual.

Se han analizado diversos buscadores antes de elegir finalmente ***trovator.com***. El resto de buscadores se han descartado porque han presentado restricciones de acceso a la hora de establecer conexión con ellos o en el momento de proporcionar el contenido solicitado.

2.2.3.1 Google (27)

Google es sin duda el buscador online más conocido y utilizado hoy en día para obtener información de la red, por ello, es el primer buscador con el que se plantea trabajar.

Lo primero que se determina, es la URL que nuestra aplicación debe lanzar para que el buscador devuelva resultados. Se realiza una búsqueda con la palabra *"uc3m"* desde un navegador web y se observa la URL que es presentada al ser mostrados los resultados:

https://www.google.es/#q=uc3m

No obstante, al utilizar esta URL desde el código del proyecto, no se obtiene ningún resultado, por lo que se investiga y se llega a la conclusión de que la URL que se debe lanzar en el navegador es:

https://www.google.es/search?hl=es&q=uc3m

Es decir, es necesario introducirle el idioma en el que se desea que se la consulta sea devuelta.

Una vez que se consigue establecer conexión con el buscador se procede a analizar el código fuente que la página de resultados posee, para determinar qué nodo es el que contiene la URL que nuestra aplicación necesita almacenar.

Sin embargo, al analizar dicho código fuente, se observa que el buscador presenta dos claras desventajas frente a otros buscadores analizados:

- Por un lado, es el buscador que ha desarrollado un mecanismo más complejo en cuanto al código fuente que devuelve una búsqueda. Esto es, al realizar una búsqueda con Google, el archivo HTML que devuelve con los resultados es el más grande y complejo, porque incorpora el código de scripts necesarios para implementar las funcionalidades que Google ofrece, como las distintas aplicaciones integradas (Maps, Traductor, Google+, etc.), la estructuración de resultados por patrocinados o no patrocinados, o la posibilidad de ver al lado del link que el buscador ofrece como resultado, un avance de su contenido en forma de imagen.

Figura 5- Fragmento código fuente Google: scripts y funciones

- Además, encapsula la URL del link que ofrece como resultado bajo un árbol de nodos más extenso que los demás buscadores, dificultando el acceso desde aplicaciones externas desarrolladas para tal fin.

Figura 6- Fragmento código fuente Google: primera URL ofrecida como resultado de búsqueda

- Por otro lado, Google posee mecanismos para detectar robots que intenten conectarse a su buscador de forma automática, bloqueando la conexión si detecta que un mismo programa realiza varias búsquedas consecutivas, por lo que la aplicación podría obtener respuestas erróneas y no funcionar si se realiza un uso prolongado de la misma. (28)

2.2.3.2 Hispavista (29)

Otro de los buscadores que se han analizado en este proyecto es Hispavista.

Igual que para el caso de Google, lo primero que se determina es la URL que se debe lanzar desde código del proyecto para obtener los enlaces sugeridos por el buscador.

Para ello, nuevamente, se procede a realizar una búsqueda en el navegador, utilizando la palabra clave *"uc3m"*, y se recoge la URL que aparece en el navegador:

http://buscar.hispavista.com/?q=uc3m&str=uc3m&cadena=uc3m

En este caso, al utilizar esta URL desde el código del proyecto, el buscador devuelve a la aplicación el código fuente correcto, por lo que la URL es válida para ser usada.

Una vez realizado este paso, se procede a analizar el código fuente proporcionado.

El aspecto del código HTML que se observa es más sencillo en comparación con el de Google, pues no posee tantas funcionalidades, como se puede comprobar en el navegador.

Sin embargo, se observa que éste no muestra las URL de las páginas web sugeridas, sino que lo que se incluye en el código HTML es la llamada a una función embebida dentro de una tabla, que crea las URL dinámicamente:

```
                    <script type='text/javascript'><!--//<![CDATA[
                              var m3_u = (location.protocol=='https:'?'https://a.hspvst.com/delivery/ajs.php':'http://a.hspvst.com/delivery/ajs.php');
                              var m3_r = Math.floor(Math.random()*99999999999);
                              if (!document.MAX_used) document.MAX_used = ',';
                              document.write ("<scr"+"ipt type='text/javascript' src='"+m3_u);
                              document.write ("?zoneid=7&amp;source=hispavista_resultado&amp;e=100&amp;goo=0&amp;e2=29");
                              document.write ('&amp;cb=' + m3_r);
                              if (document.MAX_used != ',') document.write ("&amp;exclude=" + document.MAX_used);
                              document.write (document.charset ? '&amp;charset='+document.charset : (document.characterSet ?
'&amp;charset='+document.characterSet : ''));
                              document.write ("&amp;loc=" + escape(String(window.location).split("?")[0]));
                              if (document.referrer) document.write ("&amp;referer=" + escape(String(document.referrer).split("?")[0]));
                              if (document.context) document.write ("&context=" + escape(document.context));
                              if (document.mmm_fo) document.write ("&amp;mmm_fo=1");
                              document.write ("'><\/scr"+"ipt>");
                            //]]>--></script>
                    </td>
                </tr>

            </table>
```

Figura 7- Fragmento código fuente Hispavista: algoritmo de resultados de búsqueda

2.2.3.3 Lycos (30)

El siguiente buscador que se analiza para determinar su viabilidad es Lycos.

En primer lugar, se realiza una búsqueda desde este buscador a través de un navegador, para observar la URL que se debería lanzar a través del código del proyecto.

La palabra clave que se utiliza es, de nuevo, *"uc3m"*, y la URL que el navegador presenta al devolver los resultados obtenidos es:

http://search.lycos.com/web/?q=uc3m&keyvol=00bb90cdc255895c1ff3

Como se puede observar, la URL que se muestra contiene una clave de autenticación, por lo que es necesario verificar si dicha clave se podría utilizar siempre, a modo de *clave patrón*, o en cada búsqueda se genera una nueva asociada a las palabras clave introducidas, y por tanto, no sería posible utilizar el buscador en un código donde sea necesario introducir una URL fija a modo de patrón para búsquedas.

Se realizan varias pruebas incluyendo en la URL la clave denominada *clave patrón*, primero desde el navegador web, y posteriormente desde el código de la aplicación.

Los resultados que se obtienen son los siguientes:

- Si inicialmente, y durante un corto periodo de tiempo, se realizan distintas búsquedas forzando la clave patrón para que coincida en todas ellas, se observa que los resultados obtenidos, tanto desde el navegador como desde el código, son los esperados, esto es, devuelven enlaces a páginas web cuyo contenido está relacionado con las palabras claves introducidas.
- Sin embargo, pasado un tiempo, si se vuelve a forzar la misma clave patrón y utilizando las mismas palabras clave, el resultado que se obtiene es distinto, mostrando referencias a páginas web relacionadas con el buscador, incluso si la búsqueda se realiza con la palabra clave inicial, en este caso *"uc3m"*, que generó la primera clave patrón.
- Cabe pensar, por tanto, que esta clave que Lycos utiliza se genera teniendo en cuenta otros factores diferentes a las palabras claves introducidas como elementos de búsqueda, como pueden ser la fecha y hora, y quizá información sobre la procedencia de la consulta.

Las siguientes ilustraciones muestran este fenómeno:

- Primera búsqueda realizada al buscador Lycos:

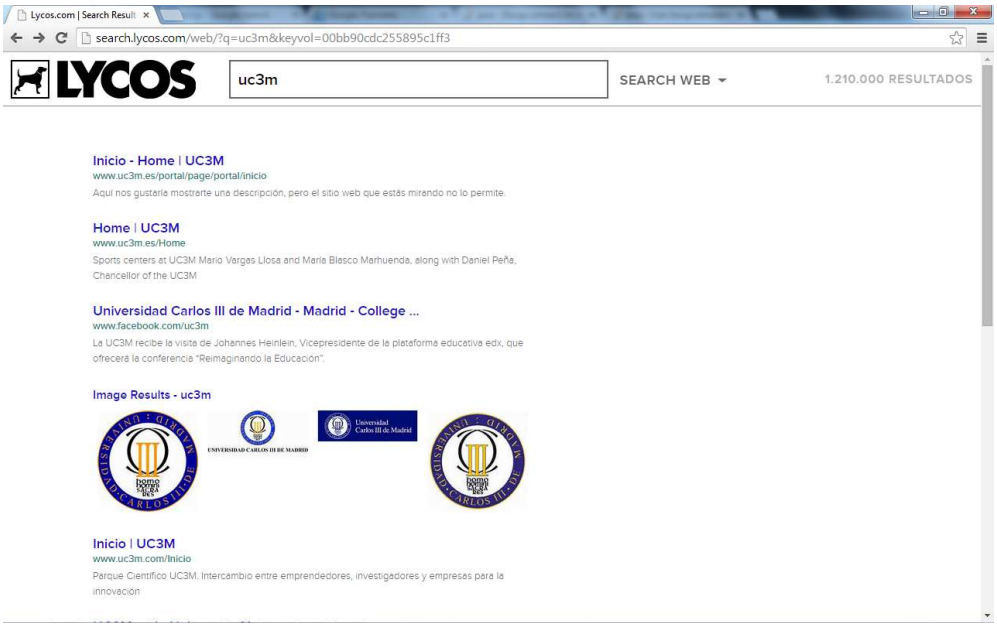


Figura 8- Primera búsqueda en Lycos

- Segunda búsqueda consecutiva realizada al buscador Lycos, forzando la clave patrón:

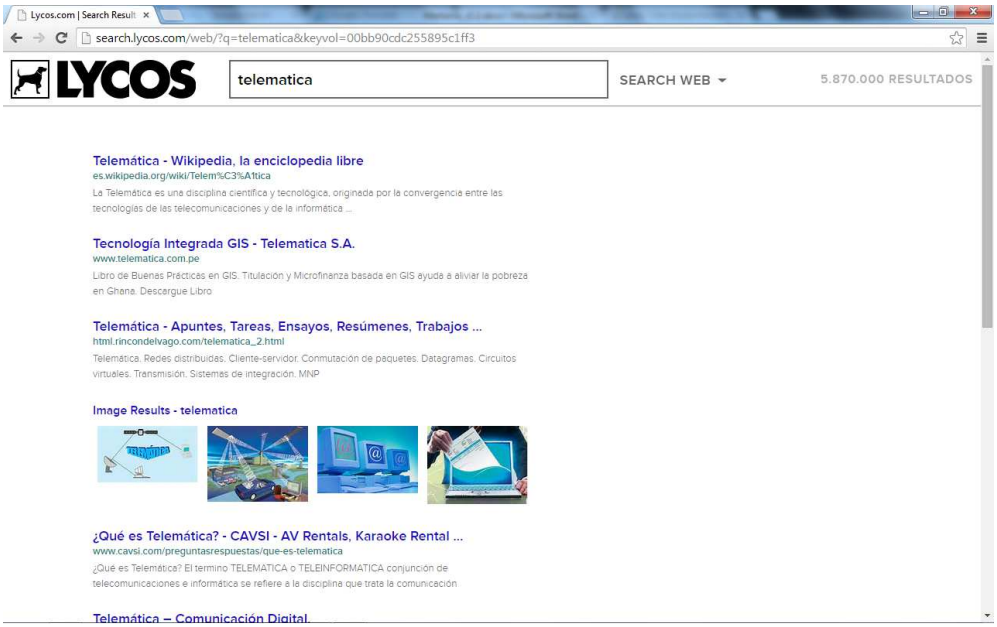


Figura 9- Segunda búsqueda consecutiva en Lycos

- Búsqueda realizada de la primera palabra clave empleada, forzando la clave patrón, transcurrido un periodo de tiempo más elevado:

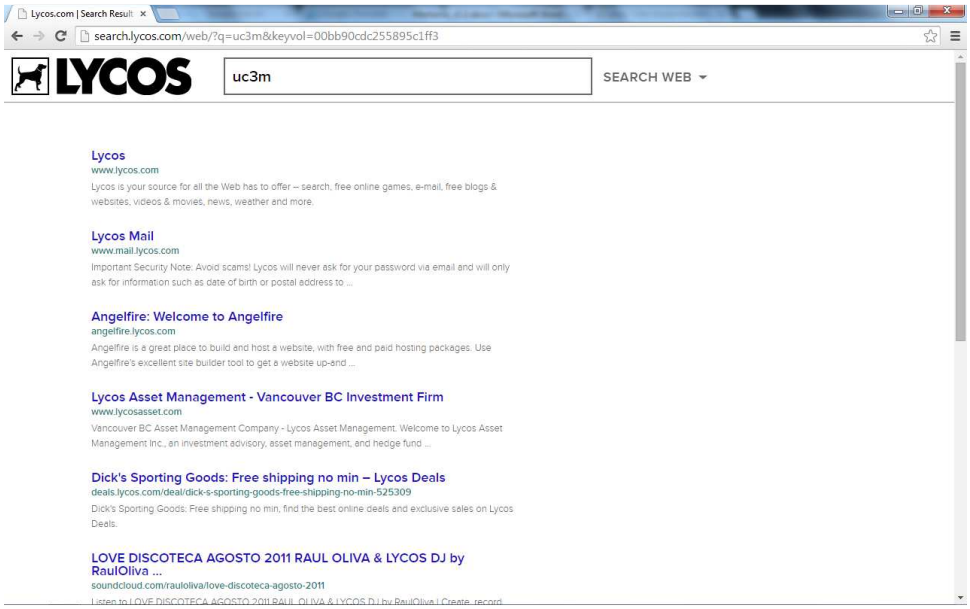


Figura 10- Segunda búsqueda consecutiva en Lycos

2.2.3.4 Trovator (31) (32)

Trovator es un motor de búsqueda español, orientado al mundo hispano.

Surgió en 1996 y fue el primer buscador español basado en arañas. En un principio comenzó indexando únicamente todas las páginas de las webs existentes en el directorio *Señas*, y a partir de ahí fue rastreando todas las webs escritas en español.

Algunas características de este buscador son:

- Enfoque dirigido al usuario hispano-hablante,
- Clasificación temática de parte de sus resultados,
- Análisis de los textos orientado al idioma español,

- Búsqueda de noticias
- Posibilidad de efectuar búsquedas restringidas a los contenidos de enciclopedias online.
- Posibilidad de filtrar las búsquedas por contenido restringido para adultos, por idioma e inclusión de imágenes o no en los resultados.
- Posibilidad de ordenar los resultados por fecha o relevancia.

Se incorpora como buscador en este proyecto debido a su orientación al usuario hispano – hablante, que se considera el público de la aplicación desarrollada, y por la sencillez y claridad del código fuente que proporciona de vuelta al realizar una búsqueda con él.

En primer lugar se analiza la URL que se debe usar dentro de la aplicación para lanzar búsquedas a este buscador. Para ello, se realiza una búsqueda a través de un navegador.

La palabra clave que se utiliza es *"uc3m"*, y la URL que el navegador presenta al devolver los resultados obtenidos es:

http://www.trovator.com/buscador/?q=uc3m&cmd=Buscar&fil=no

Se observa que la URL que maneja el buscador contiene varios parámetros:

- El primero, *q*, corresponde con los parámetros introducidos en la búsqueda.
- El siguiente, *cmd*, indica la acción que el usuario ha introducido.
- El último, *fil*, indica si se ha seleccionado el filtro con contenido para adultos que el buscador ofrece.

Tras realizar varias pruebas con la URL, y probando las distintas opciones que Trovator ofrece, la URL del buscador que se lanza en el Sistema de Recomendación será:

http://www.trovator.com/buscador/?fil=no&img=0&leng=&ord=rel&q=

Esto es, sin filtro para contenido adulto, sin imágenes, utilizando el idioma por defecto (español) y ordenando los resultados por orden de relevancia.

Se verifica que la URL es válida para ser utilizada dentro del código del proyecto, por lo que se procede a analizar el código fuente que el buscador proporciona como respuesta a una búsqueda:

```
<br><table cellpadding="0" cellspacing="0" width="96%"><tr><td width="100%"><a id="http://feeds.weblogssl.com/~r/elblogsalmon2/~3/mRI7eOq0Tdk/los-intermediarios-son-necesarios-incluso-para-los-
corruptos" href="http://feeds.weblogssl.com/~r/elblogsalmon2/~3/mRI7eOq0Tdk/los-intermediarios-son-necesarios-incluso-para-los-corruptos"><font size="3"><u>Los intermediarios son necesarios
(incluso para los corruptos)</u></font></a>
      <a id="http://feeds.weblogssl.com/~r/elblogsalmon2/~3/mRI7eOq0Tdk/los-intermediarios-son-necesarios-incluso-para-los-corruptos"
href="http://feeds.weblogssl.com/~r/elblogsalmon2/~3/mRI7eOq0Tdk/los-intermediarios-son-necesarios-incluso-para-los-corruptos" target="_BLANK">
      <img src="http://www.trovator.com/nventana.gif" border="0"></a>
       </td></tr>
          <tr><td width="100%">

<font size="2"> <font color="grey">28/08/2014</font> - Hace ya tiempo que no me importa ejercer de abogado del diablo de los intermediarios. Creo en su contribución a una sociedad
desarrollada. y mira por donde, recientemente un pequeño experimento ha venido a demostrar que los intermediarios son necesarios incluso para los corruptos. La <b>UC3M</b>, junto ..
                <br>  <font size=2>

<font color="#008000">feeds.weblogssl.com/~r/elblogsalmon2/~3/</b>...</b></font></font> </td> </tr></table>
<br><table cellpadding="0" cellspacing="0" width="96%"><tr><td width="100%"><a id="http://laflecha.net/las-mujeres-solo-aparecen-en-un-5-de-la-prensa-deportiva/" href="http://laflecha.net/las-
mujeres-solo-aparecen-en-un-5-de-la-prensa-deportiva/"><font size="3"><u>Las mujeres solo aparecen en un 5% de la prensa deportiva</u></font></a>
      <a id="http://laflecha.net/las-mujeres-solo-aparecen-en-un-5-de-la-prensa-deportiva/" href="http://laflecha.net/las-mujeres-solo-aparecen-en-un-5-de-la-prensa-deportiva/"
target="_BLANK">
      <img src="http://www.trovator.com/nventana.gif" border="0"></a>
       </td></tr>
          <tr><td width="100%">

<font size="2"> <font color="grey">15/08/2014</font> - La presencia de la mujer en la prensa deportiva es notablemente inferior a la del hombre. En concreto, la mujer tan solo aparece en un 5%
de los casos, mientras que el hombre lo hace en más de un 92 por ciento, según una investigación de la Universidad Carlos III de Madrid (<b>UC3M</b>).
LaFlecha ..
                <br>  <font size=2>
```

Figura 11- Fragmento código fuente primer resultado ofrecido por Trovator

La obtención de las URLs para recomendar al usuario se realiza accediendo al elemento [*bl_res / table / a[href] /]**abs:href*** del código fuente HTML que Trovator proporciona, utilizando los métodos disponibles de la librería Jsoup.

```
doc = getHtmlFromUrl(naviUrl+pageParam+i);
log.info("URL Buscador: "+naviUrl+pageParam+i);
Element res = doc.getElementById("bl_res");
// cada resultado se engloba en una tabla para mostrar en la web, que contiene 3 enlaces haciendo referencia
// al resultado mostrado: dos para indicar la busqueda exacta y uno para indicar la pagina raiz
Elements tables = res.getElementsByTag("table");
for(Element table:tables){//recorro cada tabla y obtengo el primer enlace, para no almacenar enlaces repetidos
   if(log.isDebugEnabled())
      log.debug("Obteniendo links...");
   Element ref = table.select("a[href]").first();
   links.add(ref.attr("abs:href"));
}
```

Figura 12- Obtención URLs ofrecidas por Trovator

2.3 Herramientas y tecnologías empleadas

Se exponen en este apartado aquellas herramientas y tecnologías que, tras un exhaustivo análisis, han sido elegidas para elaborar este proyecto.

2.3.1 Java 1.6 Standard Edition

Para el desarrollo de este proyecto se ha elegido el lenguaje de programación Java, que es un lenguaje de programación de propósito general, orientado a objetos, concurrente y basado en clases. Está diseñado para que el mayor número de programadores posible pueda utilizarlo.

Además, es un lenguaje de programación fuertemente tipado, lo que significa que la distinción entre errores en tiempo de compilación y errores en tiempo de ejecución es posible, y se debe realizar por separado.

Java es un lenguaje de programación a nivel alto, a diferencia de C y C++, con los que tiene relación, pero que permiten la programación a bajo nivel. Sin embargo, Java no ofrece acceso a los detalles de representación de la máquina. También incluye la gestión de almacenamiento automático mediante un recolector de basura, para gestionar el espacio utilizado y

proporcionar seguridad a la aplicación, al no ser posible la des asignación explícita de recursos. (33)

Algunas de las características principales de Java son (34):

- Independencia de la plataforma: Java funciona con las principales plataformas de hardware y sistemas operativos, o bien con el software JVM directamente desde Oracle.
- Alto rendimiento: Java es uno de los entornos de programación más rápidos.
- Fácil de aprender: El modelo de Java para la gestión de la memoria, los procesos múltiples y la gestión de excepciones lo convierte en un lenguaje eficaz para los desarrolladores nuevos y para los más experimentados.
- Basado en estándares.
- Optimizado para los dispositivos integrados: Compatibilidad con procesadores integrados, la gestión de potencia, los despliegues con huella , etc.
- Aplicaciones portátiles con alto rendimiento: Java alcanza un rendimiento nativo y proporciona portabilidad en una amplia gama de procesadores y sistemas operativos integrados.
- Modelo con seguridad probada: Java ofrece un entorno de aplicaciones avanzado con un alto nivel de seguridad que es idóneo para las aplicaciones de red.

La estructura de una aplicación Java consta de la siguiente sintaxis:

- Clases: son el elemento contenedor de un objeto. Cada clase contiene las características y funciones del mismo.
- Métodos: forman las acciones que el objeto puede realizar. Está contenido dentro de una clase.
- Atributos: son las características específicas del objeto, sus cualidades. También está contenido dentro de una clase.

```
package src.commons;

import java.io.File;
import java.util.*;
import org.apache.log4j.Logger;

public class CommonsUtils {

 private static Logger log=Logger.getLogger(CommonsUtils.class);


/*
* Deletes all file from a directory
*/
  public static void deleteDirectory (File dir){
     try{
        File[] files = dir.listFiles();

        for (int x=0;x<files.length;x++){
           if (files[x].isDirectory()) {
              deleteDirectory(files[x]);
           }
           files[x].delete();
        }

     }
       catch(Exception e){
          log.error("Error while deleting dir: ",e);
       }
  }


}
```

Figura 13- Clase Java con atributo y método

2.3.2 *Java Swing*

Java Swing es la librería Java utilizada para crear una Interfaz Gráfica de Usuario (***GUI***) con apariencia personalizable, lo que significa que se puede diseñar un conjunto único de componentes GUI que pueden tener automáticamente la apariencia de cualquier Sistema Operativo donde se ejecute.

Esta librería incluye toda la funcionalidad Java de la ***AWT*** (Abstract Window Toolkit), que es la librería gráfica básica inicial de Java (elementos *Button*, *Scrollbar*, *Label*, etc.), y además una serie de componentes de nivel superior, como *Tree View*, *List Box* o *Tabbed Panes.* (35)

Algunas de las características de Java Swing son (36):

- Componentes Swing GUI: Java Swing incluye un extenso listado de componentes, desde botones para dividir paneles a tablas. Además, muchos componentes son capaces de clasificar, imprimir y arrastrar y soltar.
- Soporte de personalización de la apariencia: como ya se ha mencionado, Java Swing posee la capacidad de personalizar la apariencia de la interfaz gráfica creada, por lo que un mismo programa puede ser visto con la apariencia típica de Java o de Windows, por ejemplo.
- API de accesibilidad: Java Swing permite utilizar tecnologías de ayuda como lectores de pantalla de lenguaje Braille o pantallas Braille para obtener información de la interfaz de usuario.
- Java 2D API: Permite a los desarrolladores incorporar fácilmente gráficos de alta calidad en 2D, texto e imágenes en aplicaciones y applets. Incluye también APIs para la generación y envío de resultados de alta calidad a dispositivos de impresión.
- Internacionalización: Permite a los desarrolladores crear aplicaciones que puedan interactuar con los usuarios de todo el mundo en sus propios idiomas.

Un ejemplo de código Java Swing es el siguiente:

```
public class DisplayApp extends JPanel implements ActionListener  {

 private static Logger log=Logger.getLogger(DisplayApp.class);


  static JFrame frame;
  //JPanel panel;
  JButton botBuscar, botRestaurar/* botCarga2, botCarga3*/;

  JTextField urlField;
  JFrame frame2;

  Image img;

   public DisplayApp(){
     //super();
     img = new ImageIcon("./img/fondo.jpg").getImage();
     //GridBagLayout lo = new GridBagLayout() ;
    // this = new JPanel(lo);
     this.setLayout(new GridBagLayout());
     this.setBackground(new Color(128,128,255));
     GridBagConstraints c = new GridBagConstraints();

         //botón buscar
     JLabel label = new JLabel("Introduce una URL para buscar páginas relacionadas: ");
     c.gridx = 1;
     c.gridy = 1;
     this.add(label, c);

     urlField = new JTextField("http://", 30);
     c.gridx = 1;
     c.gridy =10;
     this.add(urlField, c);
     urlField.addActionListener(this);
     urlField.setToolTipText("La dirección web debe empezar por \"http://\" o \"https://\"");

     botBuscar= new JButton("Buscar");
```

Figura 14- Fragmento código Java Swing

Que proporciona la interfaz gráfica:

Figura 15- Interfaz generada con Java Swing

2.3.3 JSoup (37)

Jsoup es una biblioteca Java para trabajar con el código HTML que se encuentra actualmente en la Red, que no siempre cumple los estándares establecidos por el Consorcio World Wide Web. La API proporcionada por esta herramienta permite la extracción y manipulación de datos, utilizando las herramientas DOM, CSS, y métodos similares a jQuery.

Jsoup está diseñado para hacer frente a todas las variedades de HTML que se encuentran en la Red: desde código correcto y validado, a código con etiquetas inválidas.

Jsoup implementa la especificación WHATWG HTML5, y parsea el código HTML a DOM tal y como los navegadores modernos hacen.

Sus características principales son:

- Es capaz de localizar y analizar código HTML de una URL, archivo o cadena.
- Puede encontrar y extraer los datos, bien utilizando DOM para recorrer el código, o bien mediante selectores CSS.
- Permite manipular los elementos HTML, atributos y textos.
- Es capaz de limpiar contenido enviado por los usuarios contra una lista segura de usuarios para evitar ataques XSS (Cross-Site Scripting) de seguridad.
- Genera salidas de código HTML ordenado.

Jsoup es un proyecto de código abierto distribuido bajo la licencia liberal MIT, por lo que el código fuente está disponible para el desarrollador. Además, en la página web de la herramienta, ofrecen un API completo y un tutorial para su uso.

Para utilizarlo, únicamente es necesario descargarse su librería .jar e incluirlo en el proyecto, en este caso, dentro de la carpeta *lib*.

Para este proyecto se ha utilizado la versión **jsoup-1.7.2** en los aspectos siguientes:

- Extraer el contenido existente en las URLs proporcionadas por el usuario, a través de una conexión realizada utilizando la interfaz Connection de Jsoup.

```
public static Document getHtmlFromUrl(String url){
   Document doc = null;

   try{
      doc = Jsoup.connect(url).userAgent("Mozilla").timeout(60000).get();
   // log.error(doc.outerHtml());

   }
       catch(Exception e){
         log.error("Error al obtener html de url: ",e);


      }
   return doc;

}
```

Figura 16- Conexión a URL mediante Jsoup

- Lanzar búsqueda de resultados similares al buscador web con las palabras seleccionadas como relevantes por el Sistema de Recomendación implementado y parseo del resultado devuelto para almacenar las URL devueltas por el buscador.

```
public static LinkedList<String> getLinkFromNavigator(LinkedList<String> words){
   Document doc = null;
   String text = "";
   String naviUrl = "http://www.trovator.com/buscador/?fil=no&img=0&leng=es&ord=rel&q=";
   String pageParam ="&np=";
   LinkedList<String> links = new LinkedList<String>();
   String prev = "";
   try{

      for(int i=0;i<5;i++){
         naviUrl += ProcessWords.normalizeWordForUrl(words.get(i));
         if(i!=4)
            naviUrl+="+";
      }
      for(int i=0;i<5;i++){
         doc = getHtmlFromUrl(naviUrl+pageParam+i);
         log.info("URL Buscador: "+naviUrl+pageParam+i);
         Element res = doc.getElementById("bl_res");
         // cada resultado se engloba en una tabla para mostrar en la web, que contiene 3 enlaces haciendo referencia
         // al resultado mostrado: dos para indicar la busqueda exacta y uno para indicar la pagina raiz
         Elements tables = res.getElementsByTag("table");
         for(Element table:tables){//recorro cada tabla y obtengo el primer enlace, para no almacenar enlaces repetidos
            if(log.isDebugEnabled())
               log.debug("Obteniendo links...");
            Element ref = table.select("a[href]").first();
            links.add(ref.attr("abs:href"));
         }
      }
   }
       catch(Exception e){

         log.error("Error al extraer texto: ",e);
      }
    if(log.isDebugEnabled())
   log.debug("Numero links:"+links.size());
   return links;
}
```

Figura 17- Parseo HTML mediante Jsoup

2.3.4 Log4j (38)

Log4j es la herramienta que se ha utilizado en este proyecto para generar una traza a nivel de ejecución del mismo, donde se almacenen tanto los sucesos relevantes que ocurren en la aplicación, como posibles fallos que se puedan generar, facilitando la labor de análisis y rastreo de errores.

Log4j es una librería para generar ficheros de Log o registro en Java. Esta herramienta hace posible habilitar el registro de eventos en tiempo de ejecución sin modificar el binario de la aplicación, y está diseñado para que su uso no suponga un coste elevado en cuanto a rendimiento en la aplicación.

El comportamiento de Log4j se puede controlar mediante la edición de un archivo de configuración, sin tener que modificar el código fuente de la aplicación.

Esto implica además, que esta herramienta permite controlar qué eventos o sucesos en la aplicación se almacenan en el fichero de Log, permitiendo varios niveles de importancia del suceso, desde eventos que se almacenan únicamente para debuggear la aplicación, elementos que se almacenan a nivel informativo o elementos que se almacenan como error grave del sistema, lo que proporciona una noción rápida y eficaz del funcionamiento del sistema implementado, a la que permite reducir el volumen de salida y el coste de ejecución del sistema.

La versión de Log4j empleada en este proyecto es **log4j-1.2.17**. El archivo de configuración se encuentra en la carpeta raíz del proyecto, con nombre **log4j.properties** y los ficheros de log se encuentran dentro de la carpeta *log*, contenido dentro del directorio raíz.

2.3.5 Xstream (39)

XStream es una librería sencilla para serializar objetos a XML y viceversa.
Sus características principales son:

- Facilidad de uso. Se trabaja a alto nivel para simplificar casos de uso comunes.
- No requiere mapeos. La mayoría de los objetos pueden ser serializados sin necesidad de mapeos.
- Rendimiento. Es una librería que se ejecuta velozmente, y que consume poca memoria, por lo que sirve para serializar documentos de volumen considerable.
- XML limpio. No duplica información y el XML.
- No requiere modificación de objetos. Serializa elementos internos de una clase, como los elementos privados o finales.
- Soporte gráfico de objetos completo. Las referencias duplicadas existentes en un modelo de datos se mantienen. Permite además referencias circulares.
- Se integra con otras APIs XML. Mediante la implementación de una interfaz, Xstream puede serializar directamente a/desde cualquier estructura de árbol (no sólo XML).
- Estrategias de conversión personalizables.
- Marco de seguridad. Control sobre objetos no identificados para evitar problemas de seguridad con entradas de datos manipuladas.
- Mensajes de error. Cuando se produce una excepción debido a código XML mal formado, la herramienta es capaz de notificarlo detalladamente.
- Formato de salida alternativo.

En este proyecto se ha usado la versión **xstream-1.4.4** para los siguientes cometidos:

- Almacenar en un fichero XML las palabras más relevantes obtenidas dentro de una página web junto al número de veces que aparecen.
- Almacenar en un fichero XML las palabras más relevantes a nivel global, junto al número de veces que han aparecido, cuyo uso será válido en posteriores búsquedas, pare ser utilizadas en el aprendizaje del Sistema de Recomendación.

```
public static void getXml(Map< String, Integer > map, File outputFile) throws Exception{

  BufferedWriter bw = null;
   String xml="";
  try{
  // convert to XML
     XStream xStream = new XStream();
     xStream.alias("map", java.util.Map.class);
     xml = xStream.toXML(map);
  // log.debug("xml:"+xml);
   }
      catch(Exception e){
        log.error("Error al usar la libreria xstream",e);
      }

  try{
     bw = new BufferedWriter(new FileWriter(outputFile));
     bw.write(xml);
  }
      catch(Exception e){
        log.error("error al crear xml en xstream:",e);
      }
  finally{
     bw.close();
  }
}
```

Figura 18- Conversión Java a XML con Xstream

- Extraer de un fichero XML las palabras almacenadas con su frecuencia de apariciones para trabajar con ello dentro de la aplicación.

```
public static Map< String, Integer > obtainFromXml(File inputFile) throws Exception{

  BufferedReader br = null;
  String xml ="";
  String str="";
  Map<String,Integer> map = null;
  try{
     br = new BufferedReader(new FileReader(inputFile));
     str=br.readLine();
     while(str!=null){
        xml = xml+"\n"+str;
        str=br.readLine();
     }
  // convert to XML
     XStream xStream = new XStream();
     xStream.alias("map", java.util.Map.class);
     map = (Map<String,Integer>) xStream.fromXML(xml);

  }
      catch(Exception e){
        log.error("Error al extraer xml", e);
      }
  finally{
     br.close();
  }
  return map;
}
```

Figura 19- Conversión XML a Java con Xstream

2.4 Funcionalidad

Como se ha mencionado anteriormente, el cometido de este proyecto consiste en desarrollar el núcleo de un Trazador de Pistas, a partir de un Sistema de Recomendación Basado en Contenido, que permita ofrecer al usuario páginas web similares a las que él mismo introduce en el sistema.

La aplicación desarrollada consiste, por tanto, en una aplicación de escritorio con acceso a Internet, donde se muestra una pantalla principal en la que el usuario puede acceder a las diferentes posibilidades que ofrece el programa:

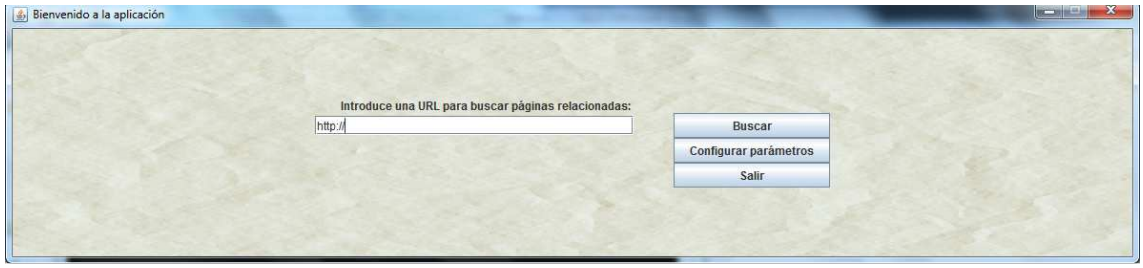


Figura 20- Pantalla principal de la aplicación

- **Buscar páginas web similares**. Es la funcionalidad principal del sistema, donde se ha desarrollado toda la lógica que forma el Trazador de Pistas, que consiste en las siguientes tareas:
  - conexión con la URL introducida,
  - extracción del contenido de la URL y análisis del mismo,
  - obtención de las palabras con significado más repetidas,
  - combinación de las mismas con las palabras relevantes almacenadas que son fruto de búsquedas anteriores, si las hubiera,
  - obtención del nuevo set de palabras más relevantes para lanzar al buscador,
  - conexión con el buscador para proponerle una nueva búsqueda con el set de palabras obtenido,
  - extracción del contenido que devuelve como resultado el buscador obtención de las primeras páginas web propuestas,
  - y finalmente, generación de una nueva pantalla con las páginas web elegidas por la aplicación para ser mostradas al usuario como resultados similares a la página web que él introdujo.

El flujo de interacción entre el usuario y la aplicación es el siguiente:

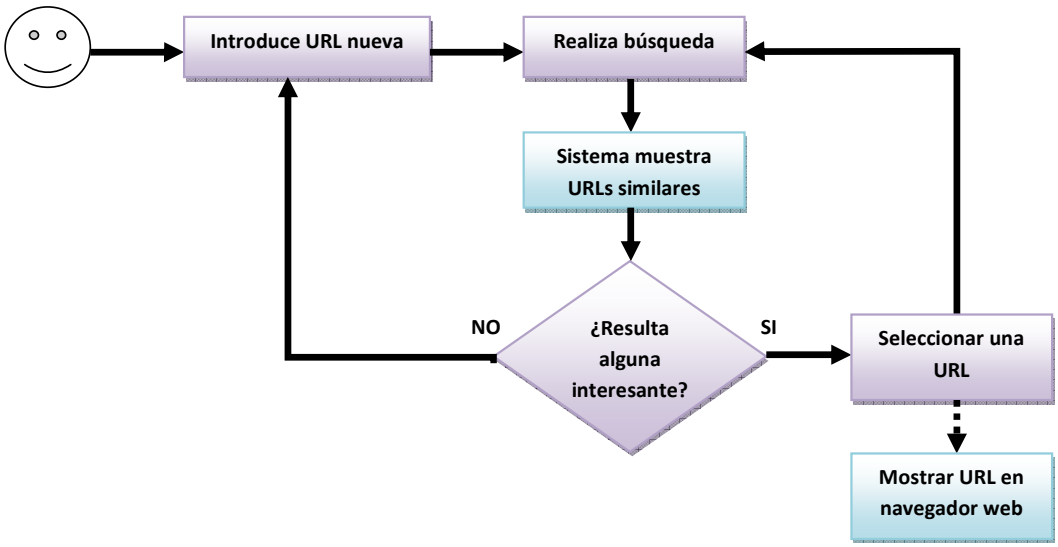


Figura 21- Flujo actuación usuario búsquedas URL

Esta funcionalidad está formada a nivel gráfico por dos elementos en la pantalla principal: un área de texto donde el usuario puede introducir la URL en la que está interesado, y un botón “Buscar” donde ejecutar toda la lógica si el usuario lo pulsa.

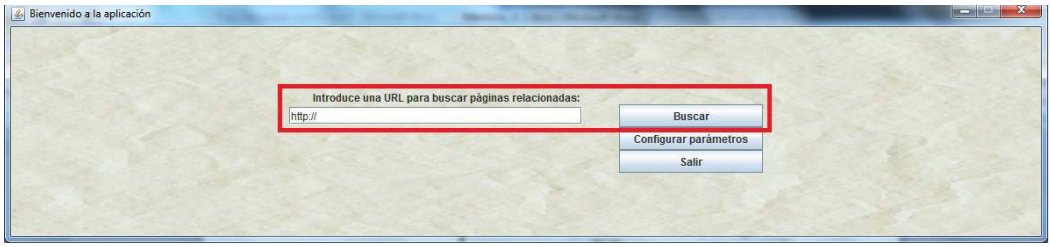


Figura 22- Delimitación gráfica en pantalla inicial de funcionalidad Buscar

Además, genera una nueva pantalla, donde se muestran los resultados obtenidos por el Trazador de Pistas, para que el usuario evalúe si alguno de los enlaces ofrecidos es de su interés:

Figura 23- Pantalla gráfica con resultados similares mostrados a usuario

Si alguno de los enlaces es de su interés, puede seleccionarlo si lo desea tanto como para volver a ejecutar el Trazador de Pistas en busca de nuevos enlaces, desde la pantalla principal, como para continuar con la navegación por la Red (si ha activado esta opción en el menú de configuración).

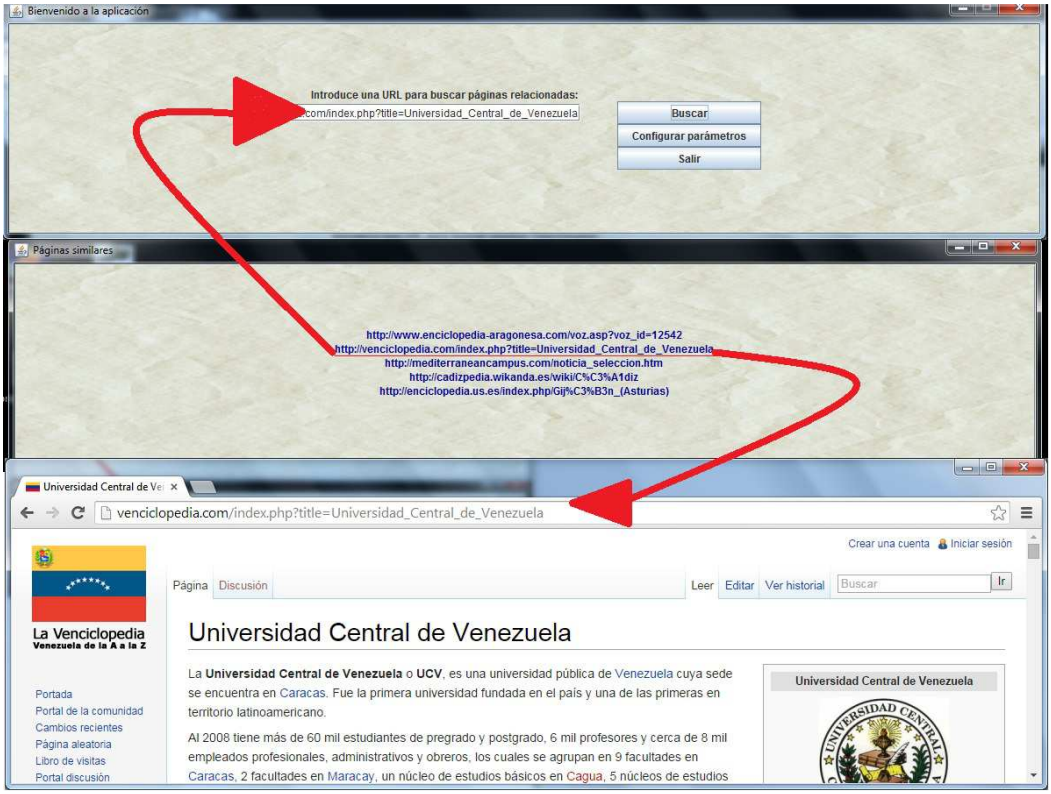


Figura 24- Selección de enlace en página de resultados

Si el número de resultados obtenidos por el Trazador de Pistas es menor que el número de links configurado en la aplicación para mostrar, la pantalla de resultados similares mostrará al final de todos los links disponibles un mensaje de texto indicando este hecho:

Figura 25- Pantalla resultados con menos resultados disponibles de los configurados

- **Configurar la aplicación**. Aunque el objetivo principal del proyecto es realizar una herramienta que permita al usuario navegar con más agilidad por la Red, se ha dotado al sistema de una pantalla de configuración mediante la cual se le ofrece la posibilidad de configurar ciertos aspectos de la aplicación, para poder así elegir :
  - realizar distintos tipos de búsquedas en función de los parámetros seleccionados. Se le ofrece la posibilidad de utilizar los lemarios de palabras precargados en la aplicación o no, normalizar las palabras que se analizan para evitar la distinción entre mayúsculas y minúsculas y el uso de tildes, y permitir contabilizar caracteres numéricos o no.
  - el número de enlaces que la aplicación le debe mostrar en la pantalla de resultados cada vez que realiza una búsqueda. Puede elegir un número entre 1 a 50.
  - si desea que al seleccionar una página sugerida, dicho enlace sea mostrado en un navegador web, además de ser cargado en el área de búsqueda de la pantalla principal. Por defecto, al seleccionar uno de los enlaces propuestos, la aplicación únicamente lo utilizará para cargar esa URL en el área de búsqueda de la pantalla principal, pero si el usuario lo desea, la aplicación puede abrir también la URL en el navegador web.
  - restaurar los datos almacenados en la herramienta para poder comenzar una nueva búsqueda, eliminando las palabras que el sistema ha guardado como relevantes fruto del aprendizaje de búsquedas anteriores.

De esta manera se permite, por un lado, adaptar la aplicación al gusto de cada usuario, dejándole elegir aspectos visuales de la misma, como el número de links mostrados o la apertura de la página web en el navegador.

Y por otro lado, ofrece la posibilidad a usuarios más avanzados, que están más familiarizados con la navegación web en la Red, de realizar análisis semánticos sobre la información existente en la web, pues al poder distinguir entre el uso de tildes en las palabras, el uso de lemarios en castellano, o la inclusión de números en las búsquedas, los resultados obtenidos variarán, ya que en muchas ocasiones el contenido en la web está formado por palabras mal formadas o escritas, palabras procedentes de otros idiomas, o neologismos que todavía no han sido incluidos en el diccionario de la Real Academia de la Lengua Española.

Además, poder restaurar los datos almacenados en la aplicación, permite que el Trazador de Pistas pueda empezar de cero con la nueva configuración que el usuario establezca, de manera que sea el usuario el que elija si desea que el Trazador de Pistas aplique la información obtenida a través del aprendizaje del Sistema de Recomenda-

ción, o si por el contrario, prefiere que el Sistema Recomendador comience de nuevo el aprendizaje.

La configuración inicial con la que se ejecuta la aplicación está definida en el fichero *project.properties* de la herramienta, que contiene además de estos parámetros, las rutas de los directorios que la herramienta necesita crear para su funcionamiento. Por lo tanto, si el usuario no desea realizar ninguna configuración adicional, el sistema ya posee una configuración por defecto que le permite utilizar el Trazador de Pistas.

Esta configuración por defecto es la siguiente:

- Se muestran al usuario 5 páginas web similares a la introducida
- Se normalizan las palabras para no tener en cuenta tildes ni mayúsculas/minúsculas
- No se tienen en cuenta los caracteres numéricos
- Se utilizan los lemarios precargados en la herramienta para saber si una palabra es relevante o no.
- No se mostrará el resultado seleccionado por el usuario en un navegador web independiente, sino que únicamente se utilizará para realizar la siguiente búsqueda.

En el ámbito gráfico de la aplicación, esta opción se muestra al seleccionar el botón "Configurar parámetros" existente en la pantalla principal:

Figura 26- Delimitación gráfica en pantalla inicial de funcionalidad Configurar parámetros

Una vez pulsado el botón, se despliega una nueva pantalla de configuración, donde se encuentran presentes todas las opciones de configuración ya mencionadas, más un botón "Volver", que permite al usuario cerrar la ventana de configuración y regresar a la pantalla principal cuando lo desee.

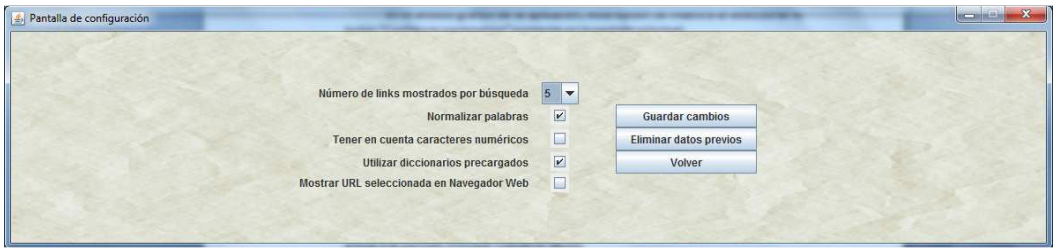


Figura 27- Pantalla Configuración aplicación

Una vez realizado algún cambio, al pulsar el botón "Guardar cambios", se procede a registrar la nueva configuración de los parámetros elegidos, y se muestra una ventana para confirmarle al usuario que se han realizado los cambios. Si, por el contrario, hubiera algún problema a la hora de modificar los parámetros, se mostraría un mensaje de error al usuario.

De igual manera, cuando el usuario desea restaurar la aplicación, pulsando el botón "Eliminar datos previos", si se realiza la acción correctamente se le muestra un mensaje notificándolo, mientras que si se produce algún error, se le mostrará un mensaje de error.

- **Salir de la aplicación**. En el momento que el usuario desee abandonar la aplicación, puede hacerlo seleccionando la opción "Salir" que se le ofrece en la pantalla principal. Este botón ofrece las siguientes opciones:
  - Abandonar la aplicación sin guardar los posibles cambios de configuración que se hayan realizado. En este caso, se cerrará la aplicación directamente, y la siguiente vez que se ejecute, la configuración del sistema será la última almacenada en el fichero *project.properties.*
  - Abandonar la aplicación guardando los posibles cambios de configuración. En este caso, la aplicación almacenará los nuevos parámetros en el fichero *project.properties* primero, sobrescribiendo la configuración existente, y después finalizará la aplicación.
  - Volver a la pantalla principal. Si no se desea en realidad abandonar la aplicación, se puede volver a la pantalla principal de la misma.

En el entorno gráfico de la aplicación, la opción "Salir" se encuentra en la pantalla principal:

Figura 28- Delimitación gráfica en pantalla inicial de funcionalidad Salir

Una vez seleccionado el elemento, la aplicación desplegará la siguiente pantalla:

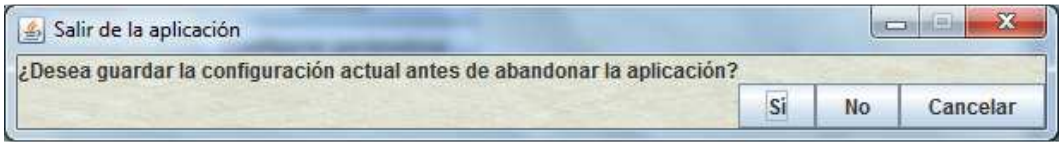


Figura 29- Pantalla Salir de la aplicación

Al elegir la opción "Sí" o la opción "No" la aplicación se ejecutará acorde a lo elegido y se cerrará.

Si, por el contrario, se elige la opción "Cancelar", la ventana emergente se cerrará, pero la pantalla principal de la aplicación permanecerá abierta.

2.4.1 Buscar páginas web similares

Se procede ahora al análisis detallado de la funcionalidad desarrollada en el núcleo del Trazador de Pistas, que consiste en un algoritmo personalizado de un Sistema de Recomendación basado en Contenido, partiendo de las características expuestas en apartados anteriores de esta memoria para su desarrollo.

A nivel de aplicación, implementar un Sistema de Recomendación se traduce en las siguientes acciones: conectar con la URL proporcionada, extraer la información de la misma, realizar el análisis sintáctico del texto para eliminar palabras sin significado semántico, contabilizar las palabras más relevantes, almacenarlas para su uso, establecer el juego de palabras más relevante teniendo en cuenta las palabras guardadas de búsquedas anteriores y las palabras escogidas en la búsqueda actual, y lanzar al buscador web para obtener los resultados que ofrecer al usuario.

Se dividen todas las tareas necesarias en diferentes bloques de actuación, según la funcionalidad que estén cubriendo, que se proceden a explicar detenidamente.

2.4.1.1 Obtener contenido de página web

Obtener el contenido de la página web que el usuario ha introducido en el sistema es el primer paso para poder evaluar el tipo de información existente y poder ofrecer páginas web similares. Para ello, se necesitan resolver los siguientes pasos:

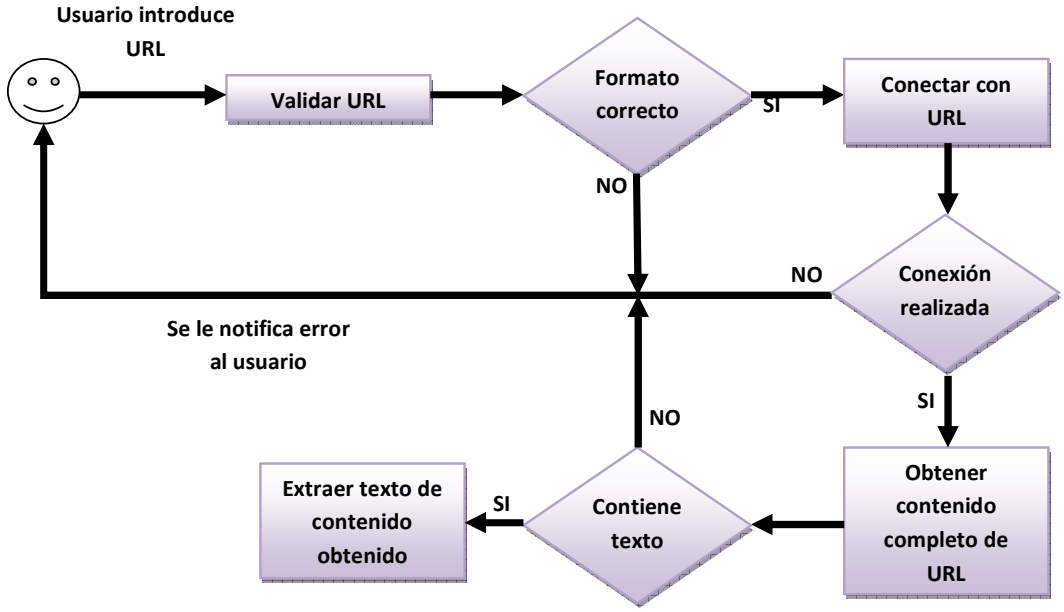


Figura 30- Pasos necesarios para obtener el contenido de una página web

Validar URL:

Cuando un usuario comienza a utilizar la herramienta, una vez que ha sido abierta y la pantalla principal desplegada, primero debe introducir la dirección URL de una página web que desee en el área de texto habilitado para ello y, o bien pulsar el botón *Enter* del teclado, o bien pulsar el botón *Buscar* habilitado para ello.

Dado que una dirección URL tiene ciertas características específicas, el sistema debe validar internamente que el formato introducido por el usuario es el correcto, es decir:

- Debe comenzar por *http://* o *https://*
- Debe coincidir con el siguiente patrón Java, que corresponde al formato típico de una URL:

```
^[A-Za-z][A-Za-z0-9+.-]{1,120}:[A-Za-z0-9/](([A-Za-z0-9$_.+!*,;/?:@&~=-])|%[A-Fa-f0-9]{2}){1,333}(#([a-zA-Z0-9][a-zA-Z0-9$_.+!*,;/?:@&~=%-]{0,1000}))?$
```

Si no es así, se le notificará al usuario para que verifique la URL que ha introducido.

Realizar conexión:

Una vez comprobada la URL, se intenta establecer conexión con la misma para extraer su contenido. Esta tarea se ha implementado utilizando la librería JSoup, que proporciona los métodos necesarios para ello.

```
doc = Jsoup.connect(url).userAgent("Mozilla").timeout(30000).get();//intenta conectar por 30seg
```

Figura 31- Fragmento código conexión a URL con JSoup

Para que la conexión sea reconocida, es necesario indicarle un cliente de conexión. En este caso se ha establecido Mozilla por defecto, pero podría haberse utilizado cualquier otro, pues no afecta en la conexión.

Adicionalmente, se estable un tiempo inicial de conexión de 30 segundos, para evitar que si se produce algún problema de red, la aplicación permanezca continuamente intentando conectarse a Internet y bloquee el proceso indefinidamente.

Cabe mencionar también, que se ha implementado un mecanismo de conexión que ofrece tres reintentos más si la primera conexión es fallida, aumentando en 10 segundos el tiempo de conexión en cada reintento. De esta manera, si se ha producido un fallo momentáneo de red, o alguna otra causa fortuita por la que la conexión no se ha podido realizar a la primera, la aplicación sigue intentándolo antes de reportarle el error al usuario.

Si finalmente la conexión no es posible, se le notifica al usuario para que esté al tanto del fallo producido.

Obtener contenido completo URL:

Una vez establecida la conexión, JSoup nos devuelve automáticamente el contenido completo que forma la URL con la que se ha conectado, esto es, el código fuente HTML completo, incluyendo *Header*, *Body* y metadatos, en formato Document, lo que quiere decir que ya está estructurado en Nodos que forman una estructura de árbol para poder ser recorrida.

Extraer texto del contenido obtenido:

Sin embargo, puesto que el interés del Trazador de Pistas se centra en conocer el contenido semántico de la página web, no es necesario almacenar la información completa que JSoup nos proporciona, sino que bastará con almacenar los elementos de tipo texto existentes en el código HTML, que son los elementos que aportan la información deseada en este caso. Para ello, se utiliza de nuevo la librería JSoup, que proporciona un método que devuelve todo el texto combinado del elemento seleccionado en el árbol y todos sus hijos, en este caso el cuerpo del HTML. Además también normaliza los espacios en blanco.

```
text[1] = doc.body().text();
```

Figura 32- Fragmento código JSoup para obtener texto

Existe la posibilidad de que la URL introducida no devuelva ningún elemento de tipo texto, bien porque la página no contenga información, bien porque la dirección introducida provoque una redirección hacia otra página web. En ese caso, la aplicación mostrará un mensaje indicativo al usuario, para avisarle de que es necesario introducir la URL final de la página que está visitando.

Adicionalmente, se extrae también el título de la página web para almacenar el texto en un fichero que lleve su nombre, y así tener control sobre el elemento seleccionado.

Para poder utilizar este título, es necesario normalizarlo, esto es, filtrar las palabras que lo forman para eliminar caracteres susceptibles de malformación en el nombre del fichero:

- caracteres de puntuación:

~!¡@#$%^&*()_+`={}[]|:;<>,./?¿ "\'

- caracteres de longitud igual a uno
- espacios en blanco, tabuladores, saltos de línea, retornos de carro.
- vocales acentuadas:

áàäéèëîïíóòöúùüÁÀÄÉÈËÎÏÍÓÒÖÚÙÜ

Por tanto, esta fase finaliza con la escritura en el directorio establecido para ello de un fichero cuyo nombre corresponde al título de la página web y cuyo contenido se corresponde con todos los elementos de tipo texto que contiene el cuerpo del código HTML de la página.

Puesto que es un fichero que contiene texto plano, la extensión del mismo será ".txt", y además, dentro de la estructura de directorios, se almacenará en la carpeta con la fecha correspondiente al momento de la búsqueda, para mantener organizada cada búsqueda que se realice.

2.4.1.2 Parsear texto y almacenar palabras relevantes

Una vez que se ha obtenido todo el texto, el siguiente paso consiste en parsearlo para extraer las palabras que proporcionan información semántica, y, una vez obtenidas, contabilizarlas para determinar cuáles se consideran las más relevantes, y que serán por tanto tenidas en cuenta.

De esta manera, se implementa el *perfil* del documento que se mencionaba en el **apartado 1.4.1** en relación a los Sistemas de Recomendación Basados en Contenido.

Los pasos necesarios en este bloque son:

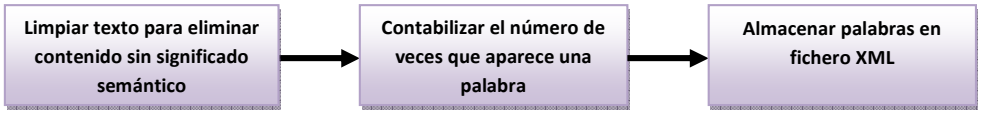


Figura 33- Pasos necesarios para obtener las palabras más relevantes de una página web

Parseo de texto para limpieza semántica:

Como se observa en la figura anterior, en primer lugar es necesario diferenciar entre palabras con contenido semántico relevante, como sustantivos y adjetivos, y palabras que, semánticamente hablando, no aportan información, como artículos determinados e indeterminados, pronombres personales etcétera, que en un Sistema de Recomendación son llamadas *stop words*, como se mencionó en el apartado introductorio.

Para ello, el Trazador de Pistas cuenta con un fichero llamado *dictionary.txt*, donde se ha confeccionado una lista con las palabras que deben ser descartadas. Esta lista está formada por los siguientes grupos de palabras:

- Artículos determinados e indeterminados.
- Preposiciones.
- Adverbios de cantidad, temporales, lugar, modo, afirmación, negación, orden, duda, interrogativos/exclamativos y relativos.

- Pronombres personales, demostrativos, posesivos, indefinidos e interrogativos.
- Letras del alfabeto
- Números del 0 al 9.

Para todos estos grupos de palabras, cuando ha sido posible se ha incluido el singular, el plural, su femenino y su masculino. Además, si la palabra original lleva alguna tilde, se ha incluido también la palabra sin tilde.

De esta manera, se pretende abarcar el rango máximo posible de *stop words* a tener en cuenta por el algoritmo del trazador.

Además, se tendrán aquí también en cuenta las opciones marcadas en el panel de configuración, pues dependiendo de ello, el algoritmo deberá seleccionar unas palabras u otras, elaborando un *perfil* diferente de la página web en cada caso. El algoritmo tendrá en cuenta:

- si es necesario normalizar cada palabra para evitar el uso de tildes y distinción entre mayúscula y minúscula,
- si el usuario desea tener en cuenta los caracteres numéricos, o por el contrario omitirlos,
- y por último, si el usuario desea utilizar los lemarios precargados con palabras contenidas propias del idioma español, restringiendo la búsqueda de palabras a las almacenadas en ellos, o si prefiere contabilizar todas las palabras que aparezcan, incluyendo así posibles palabras pertenecientes a otros idiomas, o neologismos.

Este proceso se llevará a cabo en 3 partes:

- Primero se eliminan cadenas de texto que correspondan con direcciones URL, sustituyéndolas por espacios en blanco. Esto se realiza aplicando los filtros siguientes:

http://www\\.[a-z0-9]+\\.[a-z0-9]+
http://[a-z0-9]+\\.[a-z0-9]+

- Después se eliminan todos los caracteres de puntuación, caracteres de longitud igual a uno y espacios en blanco, siguiendo el mismo proceso que se ha mencionado anteriormente para el caso del título de la página web. Esto es:
  - caracteres de puntuación:

~!¡@#$%^&*()_+`={}[]|:;<>,./?¿ "\'

  - espacios en blanco, tabuladores, saltos de línea, retornos de carro.

- Por último, se realiza una limpieza semántica del contenido restante. Para ello, se recorre cada palabra y se analiza:
  - Si se ha marcado la opción “Normalizar palabra” en el menú de configuración, es necesario entonces en la palabra seleccionada sustituir los caracteres que contengan tildes por esos mismos caracteres sin acento, y además se sustituyen también los caracteres en mayúscula por caracteres en minúscula.
    Los caracteres con acentos que se sustituyen son:

áàäéèëîïóòöúùüÁÀÄÉÈËÎÏÓÒÖÚÙÜ

- El siguiente paso es comprobar si la palabra se trata de un número. Si es así, se tendrá en cuenta la opción seleccionada "Tener en cuenta caracteres numéricos":
  - Si el usuario ha seleccionado esta opción, esta palabra se almacena como una de las relevantes a tener en cuenta.
  - Si el usuario no ha seleccionado esta opción, esta palabra descartará y no se almacenará. El algoritmo procederá a analizar la siguiente palabra disponible.
- Si la palabra no es un número, se comprueba ahora si pertenece al grupo de palabras que se delimitó en el fichero *dictionary.txt* y que se corresponden con palabras que deben ser descartadas por el sistema. Se vuelve a comprobar además que la palabra esté formada por, al menos, dos caracteres:
  - Si el texto corresponde con una de las palabras de dicho fichero, o si sólo está compuesta por un carácter, entonces el sistema descartará dicha palabra y procederá a analizar la siguiente.
- Si la palabra no ha sido descartada, y en el caso de que el usuario haya indicado en el panel de configuración que desea utilizar los lemarios incorporados en el Trazador de Pistas, mediante la opción "Utilizar diccionarios precargados", entonces es el momento de comprobar si pertenece a uno de ellos.
  - Si la opción no está marcada, se almacena la palabra directamente y se procede a analizar la palabra siguiente.
  - Si la opción está marcada, se comprueba si pertenece a alguno de los lemarios precargados.
    - Si es así, se almacena la palabra.
    - Si no es así, se comprueba si la palabra puede ser un plural de alguna existente, si su terminación es “s” o “es” y se vuelve a comprobar si el singular existe en el lemario. Si existe se almacena la palabra, y si no existe, se descarta y se procede a analizar la siguiente palabra.

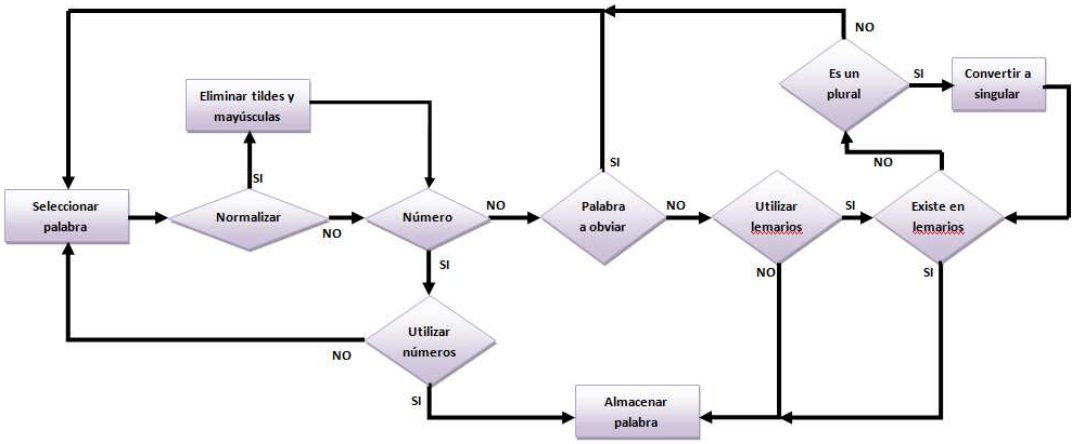


Figura 34- Algoritmo que limpieza semántica palabras

Contabilizar palabras:

Una vez que se han obtenido las palabras que poseen significado a nivel semántico, es necesario establecer cuáles de ellas son las que el sistema va a seleccionar como palabras cla-

ve que indican inequívocamente la materia sobre la que versa la página web, que constituirán el *perfil* de la página web.

Para ello, tal y como se ha explicado anteriormente, el método más directo es contar el número de veces que ha aparecido cada palabra en la página web, pues se entiende que en un texto que trate sobre una determinada materia, el conjunto de palabras que más se utilicen serán aquéllas que tengan relación directa con la materia del texto, y además, tendrán relación semántica entre ellas. Por ejemplo, en un texto donde se hable de gastronomía, las palabras más repetidas pueden ser *comer, cocinar, restaurante, plato, bebida, comida*, etc.

Para contar las palabras de cada texto se ha creado una matriz de dos columnas donde en la primera columna se almacena la palabra, y en la segunda un número que se va incrementando a medida que la palabra vuelve a aparecer en el texto. En Java se trata un objeto de tipo *Map< String, Integer >*.

```
public static  Map< String, Integer > countWords(String line){
  Map< String, Integer > map = new HashMap <String, Integer>();

  StringTokenizer tokenizer = new StringTokenizer( line );
  String word ="";
  int count =0;

// procesamiento del texto de entrada

  while ( tokenizer.hasMoreTokens() ){ // mientras haya más entrada
     word = tokenizer.nextToken().toLowerCase(); // obtiene una palabra

  // si el mapa contiene la palabra

     if ( map.containsKey( word ) ){ // esta la palabra en el mapa?

        count = map.get( word ); // obtiene la cuenta actual
        map.put( word, count + 1 ); // incrementa la cuenta

     } // fin de if

     else

        map.put( word, 1 ); // agrega una nueva palabra con una cuenta de 1 al mapa

  } // fin de while


  return map;
}
```

Figura 35- Fragmento código Java almacenar palabras más número de veces que se repiten

En este caso, se vuelven a tratar todas las palabras en minúsculas, pues puede que el usuario no haya determinado normalizarlas previamente, y se estarían contabilizando palabras iguales como palabras diferentes.

Almacenar palabras en fichero XML:

Una vez se tienen recogidas todas las palabras con significado semántico que aparecen en el texto, y se ha contabilizado el número de veces que aparecen en el mismo, se procede a almacenar esta información en un fichero XML con la siguiente estructura:

```
<entry>
  <string>universidad</string>
  <int>20</int>
</entry>
<entry>
  <string>campus</string>
  <int>16</int>
</entry>
```

Figura 36- Fragmento archivo XML almacena palabras

Como se puede observar, cada nodo del fichero XML representa una entrada de la matriz que se ha descrito anteriormente, donde el primer hijo de este nodo corresponde a la palabra almacenada y el segundo hijo del nodo corresponde al número de veces que se ha repetido en el texto.

Para convertir el objeto *Map< String, Integer >* a un fichero XML, se ha utilizado la librería ***XmlXstream***, que proporciona métodos adecuados para ello.

```
// convert to XML
   XStream xStream = new XStream();
   xStream.alias("map", java.util.Map.class);
   xml = xStream.toXML(map);
```

Figura 37- Fragmento código XmlXstream obtener XML desde Map

2.4.1.3 Seleccionar palabras para realizar búsqueda y generar historial

Una vez que se ha almacenado en un fichero la relación de palabras que forman el texto, con el número de veces que aparecen, es necesario implementar un algoritmo que las procese junto con la información almacenada de búsquedas anteriores, si existe.

De esta manera, se desarrolla un Sistema de Recomendación basado en Contenido, mediante el método de *máquina de aprendizaje*, como se expuso en el apartado introductorio, pues no sólo se tendrá en cuenta la página web actual, sino que el resultado final que se lanzará al buscador habrá sopesado también el contenido guardado, fruto de combinar búsquedas anteriores.

La fórmula que se utilizará para implementar el Sistema de Recomendación es la siguiente:

$$Historial = \bigcup_{n=1}^{20}\{max(\ INTR(Historial, Perfil)\ +\ Perfil \notin INTR(Historial, Perfil) + Historial \notin INTR Historial, Perfil$$

Donde,

$$INTR(Historial, Perfil) = Historial_n \cap Perfil_n\ ,$$

$$Historial\ \equiv Historial[palabra, valor]\ ,$$

$$Perfil \equiv Perfil[palabra, 2 * valor]$$

Esto es, el nuevo historial se compondrá de la unión de las 20 palabras cuyos valores sean los más altos entre los conjuntos formados por la intersección del historial con el perfil de la página en concreto y las palabras pertenecientes al *perfil* y al historial que no están contenidas en dicha intersección.

Como se puede observar, se le ha dado más valor al *perfil* de la búsqueda actual que al contenido existente en el historial. Esto es así porque se entiende que el valor que debe primar en el momento de elegir las palabras más relevantes, debe ser el de la navegación que el usuario esté realizando en ese instante, pues supondrá el interés actual del usuario. Por ejemplo, si un usuario ha buscado previamente varías páginas sobre gastronomía, pero la última búsqueda ha sido sobre deportes, la búsqueda que debe primar será la de deportes, pero se le debe

ofrecer todavía resultados relacionados con gastronomía, pues es la materia sobre la que ha estado más interesado anteriormente.

Antes de comenzar a desgranar el algoritmo que compone el Sistema de Recomendación, es necesario realizar algunos incisos sobre el mismo, y aclarar algunas decisiones tomadas:

- En primer lugar, es necesario matizar que el algoritmo desarrollado para implementar el Sistema de Recomendación está diseñado para procesar varios ficheros a la vez, si bien por las características del Trazador de Pistas implementado los ficheros se procesarán de uno en uno. Sin embargo, si en un futuro se extiende la funcionalidad del mismo y se obtienen las URLs que el usuario visita en tiempo real, sin que el usuario tenga que intervenir en el proceso introduciéndolas él mismo, el algoritmo está preparado para procesar dichos ficheros a la vez.
- Para ello, se ha diseñado una estructura de carpetas donde se almacena la información por fecha, ordenando los elementos a su vez por fecha, para establecer una mayor relevancia si el elemento es más reciente.
- Es por ello que en el desarrollo del algoritmo que sigue, se hablará siempre de directorio de búsquedas recientes, en lugar de un único fichero con la página web solicitada por el usuario.
- Además, resulta interesante mencionar, para clarificar explicaciones, que el nombre del directorio donde se almacenan las búsquedas recientes se llama *daily*, y los ficheros se almacenan en subcarpetas nombradas por la fecha del día en que se almacena la información. El nombre del directorio donde se almacena la lista de palabras que componen el historial del Trazador de Pistas se ha nombrado *processed* y el archivo donde se almacena dicha lista se llama *words.xml*.
- Por otro lado, a la hora de decidir el número de palabras con el que se trabaja por página web, es decir, las palabras que compondrán finalmente el *perfil* de la misma¸ se ha establecido que se tendrán en cuenta las 5 primeras palabras que hayan sido más repetidas en la página web. Esta decisión se ha tomado al comprobar que una de las estrategias de posicionamiento ***SEO*** (*Search Engine Optimization*) en Internet consiste en establecer un juego de palabras clave, normalmente 4 o 5, que se repetirán con más frecuencia que el resto dentro de la página web que se quiere promocionar en un buscador (40). Por lo tanto, hoy en día, cualquier página web que se esté visitando, si cuenta con una empresa de marketing móvil y posicionamiento SEO (sector de la tecnología que está en auge, debido a la velocidad de expansión de dispositivos móviles y presencia de empresas en la Red), seguirá esta estrategia.
  Así pues, si se extraen las 5 palabras más repetidas de la página web, se está extrayendo el set de palabras claves que identifican la materia sobre la que trata la página web.
- A su vez, el historial de búsqueda se ha limitado a 20 palabras, pues teniendo en cuenta que se limita cada página web a las 5 palabras más utilizadas, en el escenario más extremo que se podría presentar, que se corresponde con la posibilidad de que el usuario visitara páginas consecutivas que no tuvieran ninguna relación de contenido entre sí, el Trazador de Pistas almacenaría el contenido de hasta 4 páginas diferentes, sin dejar de tenerlas en cuenta como páginas que le han interesado al usuario.
- Sin embargo, el sistema permitirá variar estas decisiones, ampliando el número de palabras que se seleccionan, tanto para el *perfil* como para el historial,

en un caso específico: si existen un número mayor de palabras con el mismo peso.
Es decir, se ampliará el *perfil* de la página web si hay más de 5 palabras que se repiten el mismo número de veces, pues se entiende que todas tienen el mismo valor dentro de la página web.
De igual manera, el número de palabras a almacenar en el historial puede ser ampliado únicamente si, al unir todas las palabras posibles, existe un número mayor de palabras cuyo peso es idéntico, pues se entiende que en ese caso todas ellas tienen el mismo valor para ser tenidas en cuenta.

- De aquí en adelante, para simplificar la explicación del algoritmo, se considerará que las palabras que forman el perfil son 5 y las que forman el historial 20.

Una vez realizados los incisos precisos para comprender el algoritmo, se exponen a continuación cuáles son los pasos básicos que lo componen:

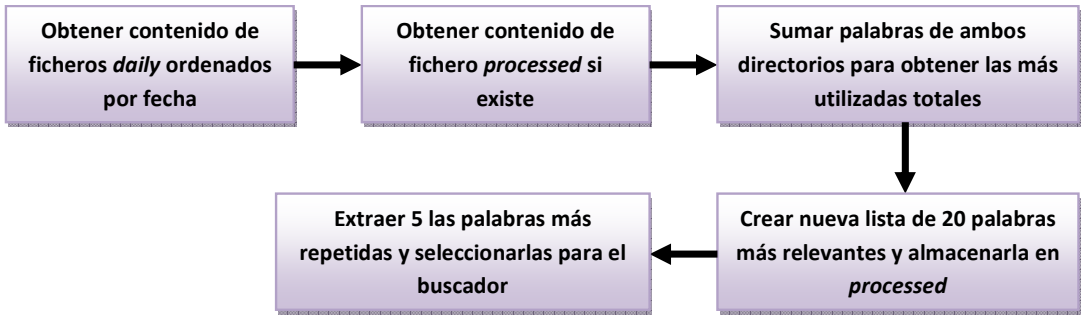
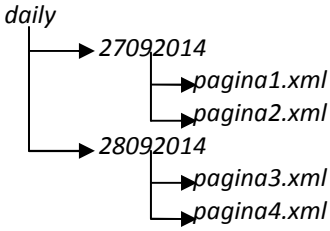


Figura 38- Pasos necesarios para desarrollar el algoritmo de elección de palabras

Obtener contenido de ficheros almacenados en directorio daily:

Como se ha mencionado ya, el directorio *daily* es el directorio donde se almacenan las búsquedas más recientes del usuario, agrupadas en subcarpetas por fecha. La estructura del directorio es el siguiente:



El contenido de este directorio se elimina una vez que el algoritmo es ejecutado, pues en el momento en el que se determinan cuáles serán las palabras que se almacenen como historial, se habrán utilizados los ficheros existentes en el directorio y almacenarlos para volver a utilizarlos supondría duplicar información en la ejecución siguiente, además de un gasto de memoria innecesario.

Para obtener la relación de palabras con la cantidad de veces que han sido empleadas de cada fichero, se utiliza nuevamente la librería *XmlXstream*, que convierte un fichero XML en un objeto *Map* de Java.

Los pasos que se ejecutan para procesar todos los ficheros son:

- Por cada subdirectorio de la carpeta *daily,* se recorren todos los ficheros y se obtienen sus duplas *palabra-repeticiones* ordenados por el número de repeticiones de cada palabra.

- Después, se eligen los 5 primeros elementos y se multiplica el número de repeticiones por 2, para darle mayor valor frente a los elementos ya almacenados anteriormente.
- Con cada *Map* obtenido con las 5 palabras más importantes de cada página web, se crea un *TreeMap* ordenado por fecha, para establecer la preferencia a las palabras más recientes.

Obtener contenido del directorio processed, el historial de búsquedas anteriores:

Las palabras que se almacenan a modo de historial en el Trazador de Pistas se guardan en el directorio *processed* de la aplicación, en un fichero XML bajo el nombre *words.xml.*

Para acceder al contenido del mismo, y manejar sus datos en la aplicación, de nuevo es necesario utilizar la librería de *XmlXstream*.

Si es la primera vez que se ejecuta la aplicación, o si el usuario ha seleccionado la opción "Eliminar datos previos", el fichero no existirá o estará vacío, por lo que el algoritmo creará un *Map* nuevo vacío para trabajar en el Sistema de Recomendación, donde al final del proceso se almacenen los datos resultantes, y una vez hecho esto, se creará o rellenará el fichero con los nuevos datos.

Unir palabras recientes más historial para obtener las nuevas palabras relevantes:

Una vez que se tienen todas las palabras que van a estar implicadas en el proceso, las resultantes de las búsquedas recientes y las almacenadas anteriormente, es necesario sumarlas para obtener las nuevas con mayor peso en el Sistema.

Para ello, se realizan los siguientes pasos:

- Se suman primero las palabras contenidas en las páginas web visitadas recientemente. La manera en la que se procede a ello es la siguiente:
  - Se recorre el TreeMap donde se almacenaron todas las palabras por página web consultada.
  - Se almacena el contenido de la primera página web en un nuevo *Map* donde se irán sumando las palabras del resto de páginas web. Para clarificar el desarrollo de este algoritmo, se llamará de aquí en adelante *resultadoActual* a este objeto *Map*.
  - Por cada página web restante, se recorre el *Map* con sus palabras relevantes almacenado en el *TreeMap* y se compara con el contenido en *resultadoActual.* Si la palabra que se evalúa ya existe, se suma el valor contenido en *resultadoActual* con el nuevo valor, y si no existe, se añade la palabra y su valor a *resultadoActual.*
  - Al terminar de recorrer el *TreeMap* inicial, se dispondrá de un objeto *Map resultadoActual* cuyo contenido es la suma de todas las palabras relevantes de las últimas búsquedas del usuario.
- Con el objeto *resultadoActual* calculado, y las palabras almacenadas en el fichero *words.xml,* que llamaremos ahora *resultadoHistorial*, ya incorporado al proceso, el siguiente paso es sumar ambos grupos de palabras. Para ello, el procedimiento es análogo:
  - Se almacena el contenido de *resultadoHistorial* en un nuevo *Map* que se llamará *resultadoTotal*.

- Se recorrerán todas las palabras contenidas en *resultadoActual,* una por una, y se comprobará si están ya registradas en *resultadoTotal.* Si es así, se sumará el valor almacenado en *resultadoActual* al almacenado en *resultadoTotal*, y si no está contenido, se añadirá la palabra a *resultadoTotal.*

Crear nueva lista de palabras relevantes y almacenarlo en el historial:

Una vez obtenido el resultado de unir todas las palabras involucradas en el proceso, las más relevantes de cada página web reciente visitada y las almacenadas de ejecuciones anteriores, es necesario establecer cuáles serán las 20 palabras más usadas en el conjunto global, es decir, las que constituyen la materia de interés del usuario en su navegación por la Red. Para ello:

- Se ordenan todas las palabras contenidas en *resultadoTotal* por su valor, cuyos primeros elementos serán los que tengan los valores más altos.
- Se recorre *resultadoTotal*, y se seleccionan las 20 primeras palabras.

Completado el proceso, el Trazador de Pistas ya posee un historial actualizado con los intereses del usuario, por lo tanto, se procede a almacenarlo de nuevo en el fichero *words.xml*, con la librería XmlXstream nuevamente.

Obtener palabras para lanzar al buscador:

Si bien se ha conseguido una lista de 20 palabras para trabajar sobre ellas, como se ha explicado anteriormente, al buscador sólo se le ofrecerán 5, esperando que nos devuelva páginas web cuyas palabras clave (las más repetidas dentro de la página) sean las 5 elegidas por el Trazador de Pistas.

Por ello, se recorre de nuevo el *Map* que contiene las palabras que se han almacenado en el historial, se extraen las 5 primeras que poseen el valor más alto y se devuelven a la pantalla gráfica principal de la aplicación, que las manejará para utilizarlas en el buscador y devolver los resultados al usuario.

En este momento, se procede a eliminar el contenido del directorio *daily,* como se explicó anteriormente.

2.4.1.4 Lanzar búsqueda y filtrar resultados

Después de haber elegido qué palabras son las que se consideran más relevantes y resultan más interesantes en términos semánticos para determinar el interés actual del usuario, se procede a conectar con un buscador, en este caso ***Trovator***, para realizar una búsqueda con estas palabras y recoger el resultado ofrecido por el buscador.

El algoritmo implementado en esta fase, intenta obtener los 50 primeros resultados que el buscador ofrece con las palabras clave enviadas, o, en su defecto, si el número es menor de 50, los resultados disponibles ofrecidos.

Una vez obtenido el resultado del buscador, se analizarán los enlaces proporcionados y se determinarán cuáles son los adecuados para ser mostrados al usuario.

El flujo de datos en esta parte del proceso es el siguiente:

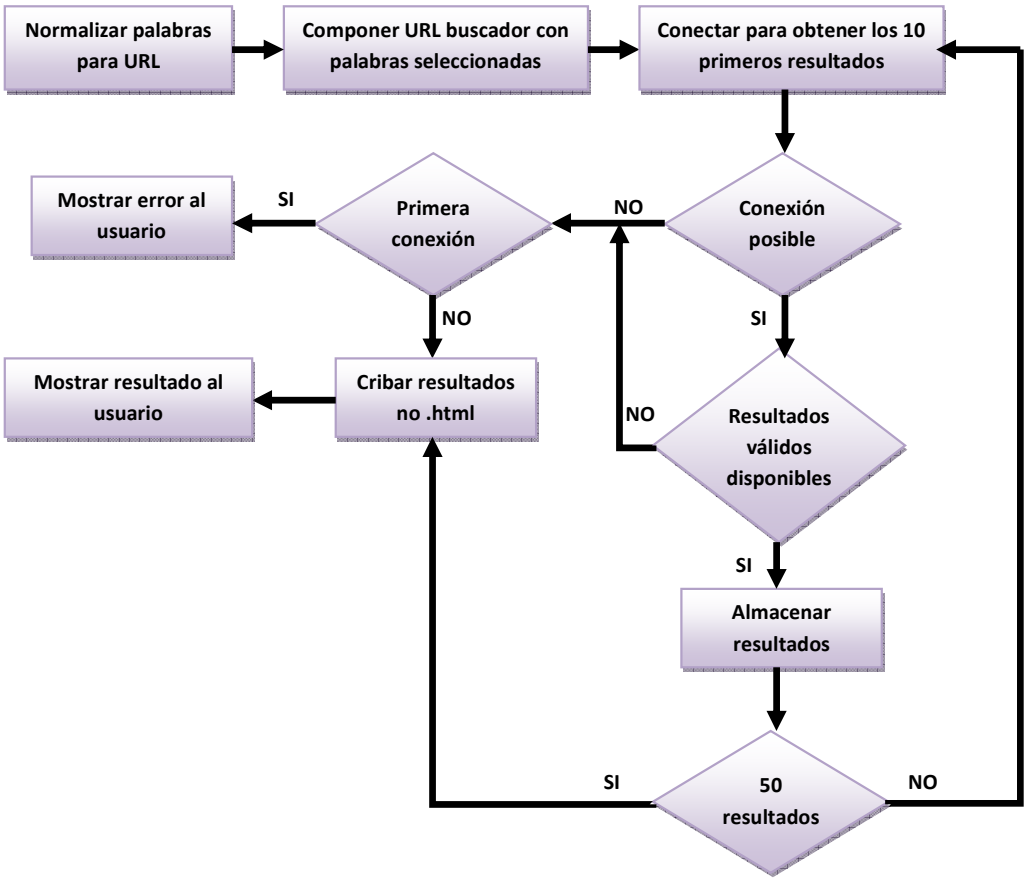


Figura 39- Flujo datos obtención resultados buscador

Normalizar palabras para URL:

En primer lugar, antes de componer la URL del buscador, es necesario comprobar que las palabras seleccionadas no contienen ningún carácter que pueda ocasionar errores en la conexión, por lo que se procede a su comprobación. Los caracteres que es necesario sustituir son:

áàäéèëíìïóòöúùüÁÀÄÉÈËÍÌÏÓÒÖÚÙÜçÇñÑ

Si estos caracteres se encuentran en alguna de las palabras, se procederá a sustituirlos por los siguientes caracteres:

aaaeeeiiiooouuuaaaeeeiiiooouuucCnN

Componer URL del buscador con palabras seleccionadas:

Una vez corregidas las palabras, si ha resultado necesario, el siguiente paso es componer la URL necesaria para realizar la conexión con el buscador.

Antes de desarrollar el funcionamiento del algoritmo, es necesario realizar una puntualización acerca de la manera en la que *Trovator* muestra las páginas web resultantes derivadas de una consulta:

Tanto en el caso de una búsqueda estándar a través de un navegador web, como en el caso de una conexión vía *JSoup* con la URL estándar mencionada en el apartado **2.2.4** de la

memoria, la primera URL de la tabla mostrada a continuación, el buscador sólo devuelve los 10 primeros resultados disponibles.

El resto de resultados se van obteniendo a medida que se realiza la paginación disponible en el navegador web. De igual manera, es necesario modificar la URL inicial para añadirle paginación extra e ir obteniendo los 10 resultados siguientes. Esto es:

http://www.trovator.com/buscador/?fil=no&img=0&leng=&ord=rel&q=*palabra1+palabra2+palabra3+palabra4+palabra5*	Primeros 10 resultados
http://www.trovator.com/buscador/?fil=no&img=0&leng=&ord=rel&q=*palabra1+palabra2+palabra3+palabra4+palabra5*&np=i	Siguientes 10 resultados (i=1…n)

Donde *i* es el número de página que se va avanzando.

Dado que el interés del Trazador de Pistas es ofrecer hasta 50 resultados, es necesario utilizar la segunda URL, variando el valor de *i* de 1 a 5, y conectarse hasta 5 veces con el buscador, para obtener todos los enlaces a las páginas web necesarios.

Conexión con el buscador y obtención de resultados devueltos:

Se utilizará *JSoup* para realizar la conexión con el buscador, de igual manera que se realizó al inicio del proceso para conectar con la página web que el usuario introduce en el sistema.

En este caso también se realizarán hasta un máximo de 3 reintentos aumentando el tiempo de espera en 10 segundos, si el primer intento de conexión resulta fallido tras esperar 30 segundos, antes de finalizar la conexión.

El proceso de obtención de las páginas web resultantes necesarias es el siguiente:

- Se intenta conectar con el buscador para obtener los 10 primeros resultados.
  - Si la conexión resulta errónea, o el buscador no devuelve ningún resultado, se finaliza el proceso y se muestra un mensaje de error al usuario.
  - Si la conexión se realiza correctamente y el buscador devuelve las primeras 10 páginas web, éstas se almacenan.
- Se intenta conectar 4 veces más con el buscador para obtener los siguientes resultados de 10 en 10.
  - Si alguna de las conexiones resulta errónea, se termina el proceso de conexión únicamente, pero se continúa la ejecución del algoritmo con las páginas web que se hayan obtenido hasta el momento, pues existe la posibilidad de que el juego de datos propuesto al buscador no devuelva la cantidad de páginas web sugeridas que se desea.
  - Si cada conexión se va realizando correctamente, obteniendo resultados, éstos se van almacenando conjuntamente.
  - Si cada conexión se realiza correctamente, pero el buscador ya no devuelve más resultados, se finaliza el proceso el proceso de obtención de páginas similares, pero se continúa la ejecución del algoritmo con las páginas web que se hayan obtenido hasta el momento, pues existe la posibilidad de que el juego de datos propuesto al buscador no devuelva la cantidad de páginas web sugeridas que se desea.

Como se ha expuesto en el apartado 2.2.4 de esta memoria, la estructura del código fuente que proporciona *Trovator* es la siguiente:

Dentro del elemento *Body* del código HTML proporcionado, la parte en la que se muestran los enlaces sugeridos se encuentra dentro del elemento "**bl_res**", que a su vez contiene un elemento "**a[href]**" por cada página web sugerida, y que a su vez contiene un atributo "**abs:ref**" donde guarda la dirección URL completa de la página web. Es decir:

<bl_res>

→<a[href] →< abs:ref URLpágina1/>

Por lo tanto, para obtener el enlace deseado, es necesario acceder al atributo abs:ref de cada elemento.

Dado que existe la posibilidad de que la página de resultados no devuelva ningún valor, o el valor que devuelva únicamente sugiera realizar otra búsqueda en el buscador, el Sistema de Recomendación implementado, filtra estas posibilidades para, por un lado no generar error si no existe ninguna URL disponible, y, por otro lado, no mostrar enlaces a la misma página del buscador.

Cribado de resultados obtenidos:

En el momento que se obtienen todos los resultados que el buscador ofrece al Trazador de Pistas, se evalúan uno por uno para comprobar si los enlaces obtenidos son efectivamente enlaces a páginas web, o son enlaces a otro tipo de contenido, como archivos Word o PDF, en cuyo caso de desestimarán y no serán ofrecidos al usuario.

Además, es necesario tener en cuenta que el número de enlaces para mostrar almacenados no debe superar el número máximo de enlaces a mostrar al usuario (50) o el número máximo de enlaces que el usuario ha establecido a través de la pantalla de configuración, seleccionando un valor en el parámetro "Número de links mostrados por búsqueda".

Por lo tanto, una vez obtenidos todos los enlaces iniciales, se recorren uno a uno y se almacenan únicamente aquéllos en los que se comprueba que su extensión no es ".doc" o ".pdf", mientras no se supere el límite de enlaces a mostrar establecido.

El resultado obtenido, compondrá el juego de enlaces que la aplicación muestra al usuario como resultado de búsqueda similar a la página web que él introdujo.

Página de resultados:

La aplicación abre una nueva ventana de resultados cada vez que todo el proceso se ha ejecutado correctamente y se obtienen enlaces similares para mostrar al usuario. Si durante todo el proceso de ha producido algún error, o la búsqueda no ha producido ningún enlace similar, se mostrará una ventana emergente para notificar el error, en lugar de dicha ventana de resultados.

A su vez, como ya se ha mencionado, si la búsqueda ha producido un número menor de resultados del número establecido en los parámetros de configuración, se mostrará un mensaje indicativo al final de la página con el siguiente texto: "*No hay más resultados que mostrar*".

Esta ventana de resultados contendrá una lista con todos los enlaces disponibles, sobre los que el usuario podrá clicar, desencadenando las siguientes acciones:

- Si el usuario no ha activado dicha opción en el menú de configuración, la única opción que sucederá será la siguiente:

    - La URL seleccionada se cargará automáticamente en la ventana de la pantalla principal para ser utilizada como nueva búsqueda si así lo desea el usuario, que deberá pulsar de nuevo el botón "Buscar" para iniciar el proceso completo otra vez.
- Si el usuario ha activado la opción "Mostrar URL seleccionada en Navegador Web", al clicar sobre cualquier enlace, además de la acción anterior, sucederá también la siguiente:
    - Se abrirá el navegador web que esté predefinido por defecto en el Sistema Operativo del equipo con la página web seleccionada.
- En ambos casos, una vez realizadas las acciones oportunas, esta ventana de resultados se cerrará automáticamente.

2.4.2 Configurar parámetros

Una vez que se ha explorado la funcionalidad principal del Trazador de Pistas implementado, se detallarán a continuación en profundidad todas las opciones disponibles en la pantalla de configuración de la aplicación.

Como se ha explicado anteriormente, a pesar de que se han definido tres partes diferenciadas para completar la funcionalidad de un Trazador de Pistas, el interés principal de este proyecto es el desarrollo de la funcionalidad central, que analiza, mediante un Sistema de Recomendación, las páginas web en las que el usuario tiene interés para proporcionar páginas similares a las suyas.

Es por ello que se han implementado algunas funciones extra que permitan al usuario, si lo desea, configurar una serie de parámetros para poder realizar diferentes estudios acerca del uso del lenguaje en la Red.

Los parámetros de configuración existentes en el sistema se pueden agrupar en dos partes, parámetros referidos a la personalización de la herramienta, y referidos al análisis semántico y ortográfico de la Red.

Por un lado, parámetros que permiten al usuario personalizar la apariencia del Trazador de Pistas. Estos parámetros son:

- **Número de links mostrados por búsqueda**: el usuario podrá especificar cuántos links desea que la aplicación le muestre en cada nueva búsqueda, en un rango establecido de 1 a 50. De esta manera podrá decidir si las primeras opciones que el trazador le ofrece le resultarán adecuadas, o si por el contrario prefiere analizar un mayor número de links antes de decantarse por alguno.
  Por defecto, la aplicación le mostrará al usuario 5 enlaces.
- **Mostrar URL seleccionada en Navegador Web:** la aplicación ofrece la opción de desplegar la URL seleccionada en el navegador establecido por defecto en el Sistema Operativo, de manera que el usuario pueda utilizar el Trazador de Pistas y navegar por la Red al mismo tiempo.
  Esta opción está desactivada por defecto.

Por otro lado, parámetros que permiten al usuario realizar búsquedas similares con resultados diferentes, pudiendo analizar contenido semántico y ortográfico en la Red:

- **Normalizar palabras:** Si está opción está marcada, el sistema no distinguirá entre el uso o no tanto de mayúsculas/minúsculas, ni de tildes. Para ello, se le aplicará a cada palabra un análisis donde se sustituyan vocales con algún tipo de acentuación por sus vocales correspondientes sin acentuar, y se considerarán todas las letras minúsculas, transformando las mayúsculas a minúsculas.

Esta opción estará activada por defecto en el Trazador, considerando la posibilidad de que dentro de un mismo texto, el autor haya escrito varias veces una misma palabra, cometiendo en alguna de las ocasiones un error ortográfico, que provoque que el Trazador considere la misma palabra como dos distintas.

- **Utilizar diccionarios precargados**: el Trazador de Pistas implementado cuenta con tres lemarios distintos precargados, que pueden ser utilizados, si el usuario así lo indica para discernir cuáles son las palabras que se considerarán que poseen significado semántico para el ámbito del proyecto. Eso quiere decir que si esta opción está activada, el trazador buscará la palabra que esté evaluando en cada momento dentro de uno de estos lemarios. Si la encuentra, considerará que es una palabra apta, si no, la descartará.
  Si esta opción no se activa, una vez que la palabra que se esté analizando haya superado los filtros previos, donde se determina si es una palabra que carece de importancia semántica para el Trazador, será directamente almacenada como válida. Esto quiere decir que, a pesar de que el Trazador implementado está pensado para el idioma español, de esta forma podrán ser incluidas palabras pertenecientes a otros idiomas, pues el Trazador no las cotejará contra los lemarios en idioma castellano.
  Los lemarios existentes se han confeccionado a partir de los lemarios disponibles en el sitio web "*Lemarios y listas de palabras del español*" (41), adaptándolos a las necesidades del proyecto, como por ejemplo eliminando sufijos y prefijos y adaptándolos a un formato válido para ser utilizado dentro del mismo.
  Se utilizan tres lemarios en el Trazador:
    - Lemario general: contiene todas las palabras que componen el castellano, sustantivos, adjetivos, verbos, etc.
    - Lemario nombres propios: contiene una lista de nombres propios comunes en España.
    - Lemario apellidos: formado por una lista de apellidos comunes en España.

  Además, se combina esta opción con la opción anterior que normaliza palabras. Para ello, se han normalizado estos tres lemarios mediante el mismo proceso por el que se normalizan las palabras analizadas en el sistema, de manera que si el usuario selecciona ambas opciones, la palabra normalizada se cotejará frente a los lemarios normalizados, mientras que si la opción "Normalizar palabras" no está disponible, los lemarios que se emplearán serán los originales, con distinción entre mayúsculas y minúsculas y acentuación en las palabras.
- **Tener en cuenta caracteres numéricos:** el Trazador también puede configurarse para tener en cuenta caracteres numéricos, si se selecciona esta opción. De esta manera, al extraer el texto de la página web solicitada, se tendrá en cuenta contenido numérico, como por ejemplo fechas.
  Por defecto, esta opción permanece inhabilitada en el Trazador de Pistas.

Una vez que el usuario selecciona o deselecciona alguna de las opciones, éstas se guardarán en el sistema cuando el usuario pulse el botón "Guardar cambios". Este botón sobrescribirá en el sistema los nuevos valores de las variables de configuración. Además volverá a cargar los lemarios con los que comparar el contenido de las páginas web en función de la variable "Normalizar palabras", de manera que si esta opción está seleccionada, los diccionarios que se cargan también serán normalizados, y, por el contrario, si la opción está desactivada, los lemarios serán los originales, para que el resultado sea consistente.

Después de haber realizado todos los cambios que considere oportuno en la configuración del sistema, y haber guardado los cambios, para regresar a la pantalla principal, es necesario que el usuario pulse el botón “Volver”, que cerrará la pantalla de configuración. De igual manera. También es la acción que necesita realizar si desea cerrar el menú de configuración sin haber realizado ningún cambio.

Por último, se ha incorporado un botón que permite resetear los datos almacenados en el Trazador de Pistas, llamado “Eliminar datos previos”.

Los elementos que resetea este botón son

- El archivo *words.xml* que se mencionó anteriormente, en el que se almacenan las palabras que se seleccionan a modo de historial.
- El directorio de trabajo donde se almacenan todas las palabras seleccionadas como texto de cada página web.

Así, si el usuario configura la aplicación de manera distinta, o si en un mismo equipo trabajan varios usuarios sobre la herramienta, se puede elegir que el Sistema de Recomendación comience su aprendizaje de cero, o combinar la información nueva con la existente y observar cómo va cambiando el proceso de aprendizaje.

Todas estas opciones proporcionan un abanico de escenarios que permitirán al usuario evaluar distintos resultados partiendo de un mismo punto, pues los resultados mostrados serán diferentes en función de los ajustes que se realicen en el algoritmo, como se podrá observar en los distintos casos de uso que se expondrán en apartados posteriores.

2.4.3 Salir de la aplicación

La última funcionalidad implementada consiste en gestionar la salida del usuario de la aplicación.

Tal y como se mencionó anteriormente, al acceder a la pantalla que gestiona la salida del usuario, se le ofrecen tres posibilidades, que responden a la pregunta que el sistema lanza al usuario al abrir esta ventana, *"¿Desea guardar la configuración actual antes de abandonar la aplicación?"*:

- Una de las opciones consiste en cancelar la acción y volver a la pantalla principal de la aplicación. Esta opción se corresponde con el botón “Cancelar” y la acción que se realizará será la pantalla de salida.
- Otra posibilidad es abandonar la aplicación sin guardar los posibles cambios que se hayan establecido en la configuración. Este caso, se perderán dichos cambios, y únicamente se conservará el historial de palabras guardadas por la aplicación. Esta acción se lleva a cabo al pulsar el botón “No”, que cierra tanto la ventana actual de salida como la ventana principal de la aplicación.
- La última posibilidad es abandonar la aplicación, pero guardando los posibles cambios que se hayan realizado en la configuración. Se corresponde con el botón “Sí” de la pantalla. En este caso, las nuevas opciones se almacenarán en el fichero de configuración *project.properties*, que ya se ha mencionado anteriormente, y del cual se obtienen todos los valores por defecto al inicializar la aplicación. Esto supone sobrescribir los valores por defecto con los que se instala aplicación y que la siguiente vez que se ejecute la misma, los valores predefinidos sean éstos que se guardan ahora. Finalmente, una vez realizados estos cambios, la aplicación se cerrará, de igual manera que en el caso anterior.

2.5 Arquitectura

Se presenta a continuación la arquitectura que se ha seguido para implementar el Trazador de Pistas.

Se ha diseñado la arquitectura de este proyecto dividiéndola en tres módulos, que se corresponden con las tres partes que forman el Trailblazer: obtención del contenido de la página web, extracción de información relevante y obtención de contenido similar. De esta manera, si uno de los módulos se amplía o se modifica, sólo es necesario realizar cambios en dicha parte, manteniendo el resto de la aplicación intacta sin cambios.

Para mantener la organización dentro del proyecto y facilitar el acceso a los diferentes módulos, se ha estructurado la organización de directorios de la siguiente manera:

- El directorio raíz se llama ***Trailblazer***, y contiene todos los elementos necesarios para ejecutar la aplicación.
- El fichero ***project.properties*** contiene todas las propiedades que deben ser usadas en la aplicación, tanto estructura de directorios, como nombre de los lemarios a utilizar y valores por defecto de los parámetros de configuración.
- El fichero ***log4j.properties*** contiene los parámetros de configuración del log que genera la aplicación.
- El directorio ***dictionary*** contiene todos los lemarios y fichero de *stop words* necesarios para la ejecución de la aplicación.
- En el directorio ***files*** será donde se almacenen los archivos con el texto de cada página web primero, en el directorio ***xhtml-files,*** el perfil final de cada página web, en el directorio ***xml-files/daily*** y el historial del Sistema de Recomendación, en el directorio ***xml-files/processed***.
- El directorio ***img*** contiene la imagen que forma el fondo de pantalla de la aplicación.
- El directorio ***lib*** almacena las librerías externas necesarias para la aplicación.

- El directorio ***log*** almacenará los fichero de log que la aplicación vaya generando a medida que se ejecute, donde se registran todos los eventos que suceden en la aplicación.

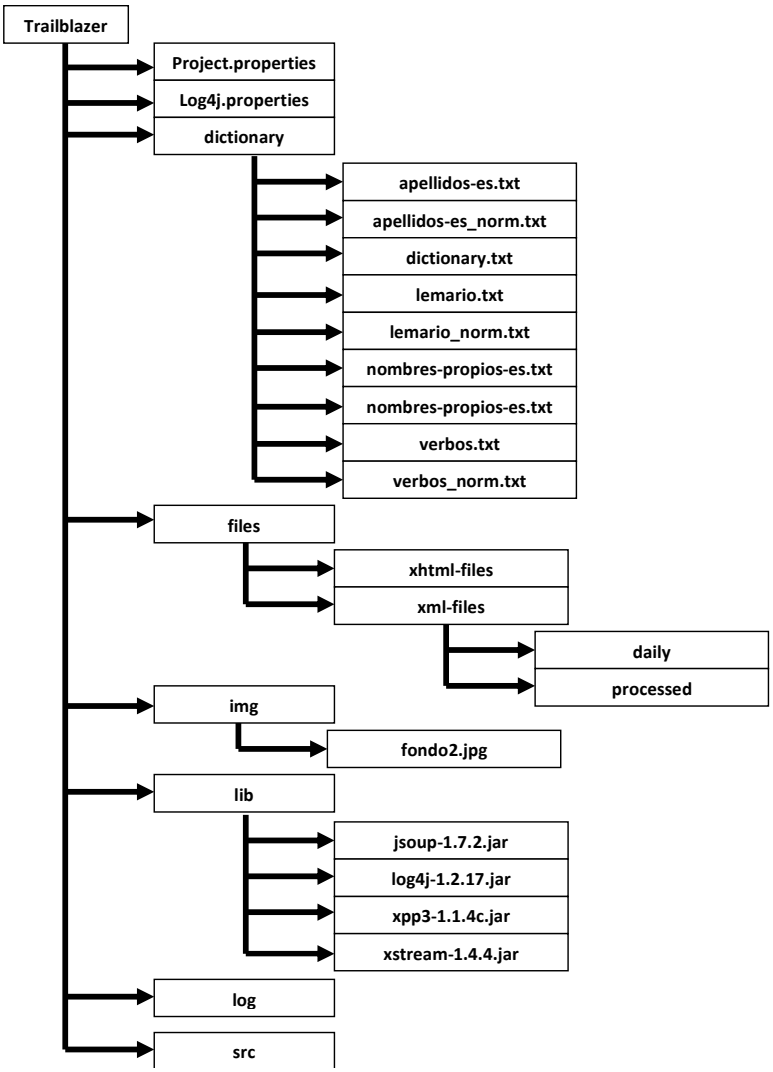


Figura 40- Organización directorios proyecto

El directorio ***src*** contiene el código implementado para este proyecto. La estructuración de carpetas se corresponde con la estructuración de paquetes de la aplicación.

Se expone a continuación cómo se han establecido los paquetes de la aplicación, y la funcionalidad que implementan:

- La clase ***RunApp*** es la encargada de iniciar la aplicación.
- Cada directorio representado en la siguiente figura, se corresponde con un paquete Java:
- ***src.semantic***: paquete que contiene las clases necesarias para realizar conexión con URL, extraer el contenido de una página web y almacenarlo en el directorio correspondiente.
    - ***ParseJSoup:*** contiene los métodos necesarios para conectar con URL, y extraer texto y resultados del mismo.
    - ***ParserDocument:*** clase que se encarga de realizar limpieza semántica del texto de una página web, esto es, eliminar palabras sin contenido semántico y recopilar las palabras que finalmente serán tenidas en cuenta.
    - ***ProcessFiles:*** clase que gestiona el contenido de los directorios donde se almacena la información.

  - ***ProcessWords:*** clase que contiene métodos necesarios para gestionar palabras, eliminar caracteres no válidos, normalizar palabras, validar URLs, validar formato para directorios, etc.
  - ***XMLXtream:*** clase mediante la que se gestiona la conversión de objetos Java a ficheros XML y viceversa.
- ***src.algorithm.:*** contiene la lógica del Sistema de Recomendación.
  - ***ProcessAlgorithm:*** implementa el algoritmo donde se seleccionan las palabras que componen el perfil de la página web, las que formarán el historial y las que se lanzarán al buscado
- ***src.navigator:*** paquete que gestiona los resultados obtenidos por el buscador.
  - ***SearchNavigator:*** gestiona la conexión y obtención de páginas similares a través del buscador. Se selecciona aquí las URLs que se mostrarán al usuario.
- ***src.props:*** paquete que gestiona la lectura y escritura, tanto en el fichero project.properties como dentro de la aplicación, de los parámetros de configuración disponibles y las rutas donde se almacenará el contenido.
  - ***ManageProperties:*** contiene métodos para cargar los parámetros de configuración del sistema desde el fichero de propiedades a la aplicación y viceversa, como para actualizar estos valores cuando se modifican desde la aplicación.
- ***src.swing:*** paquete donde se implementa la interfaz de usuario.
  - ***DisplayApp:*** clase que gestiona la pantalla principal.
  - ***DisplayConfig:*** clase que gestiona la pantalla de configuración.
  - ***DisplayExit:*** clase que gestiona la pantalla de salida.
  - ***DisplayLink:*** clase que gestiona la pantalla de resultados similares.
- ***src.exceptions:*** paquete que contiene las excepciones personalizadas para la aplicación.
  - ***LinkNavigatorException:*** excepción específica para el manejo de errores al obtener información del buscador.
  - ***ProcessURLException:*** excepción específica para el manejo de errores al realizar una conexión con una URL.
- ***src.commons:*** paquete que contiene funciones genéricas***.***
  - ***CommonsUtils***: contiene método genérico para eliminar contenido dentro de directorios.

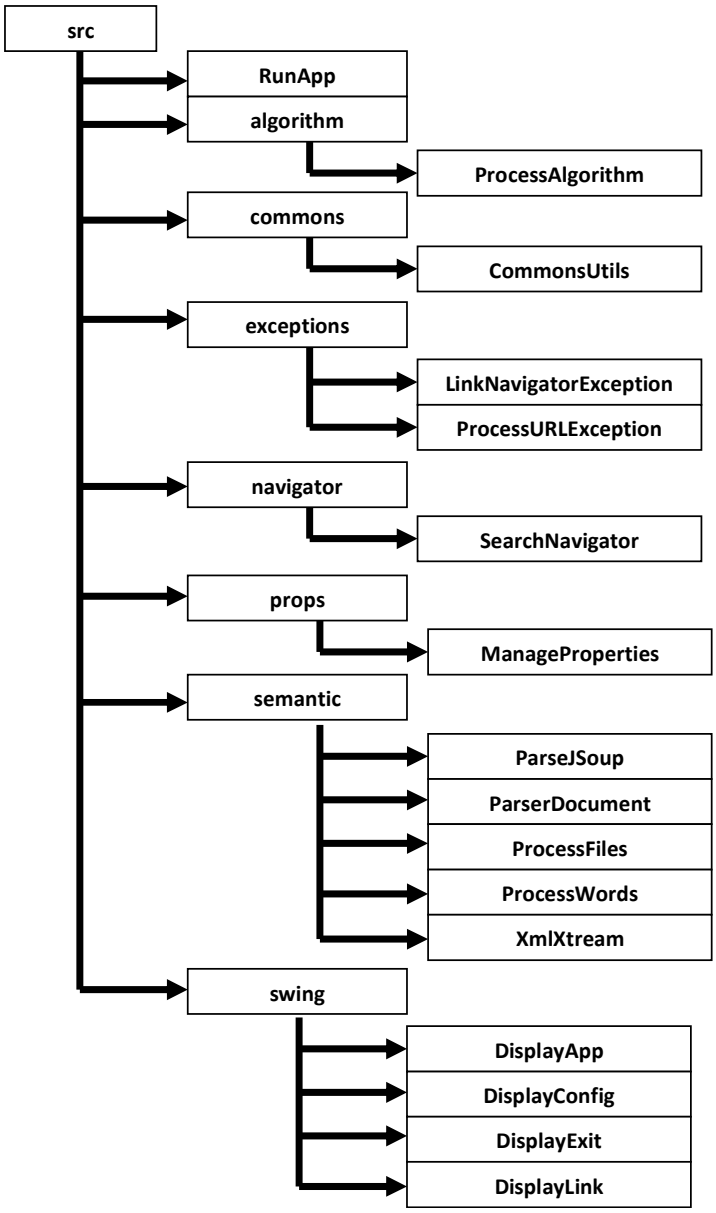


Figura 41- Organización directorios código proyecto

Se presenta a continuación el diagrama UML del proyecto, donde se observan las relaciones entre clases y los métodos y atributos que contiene cada una:

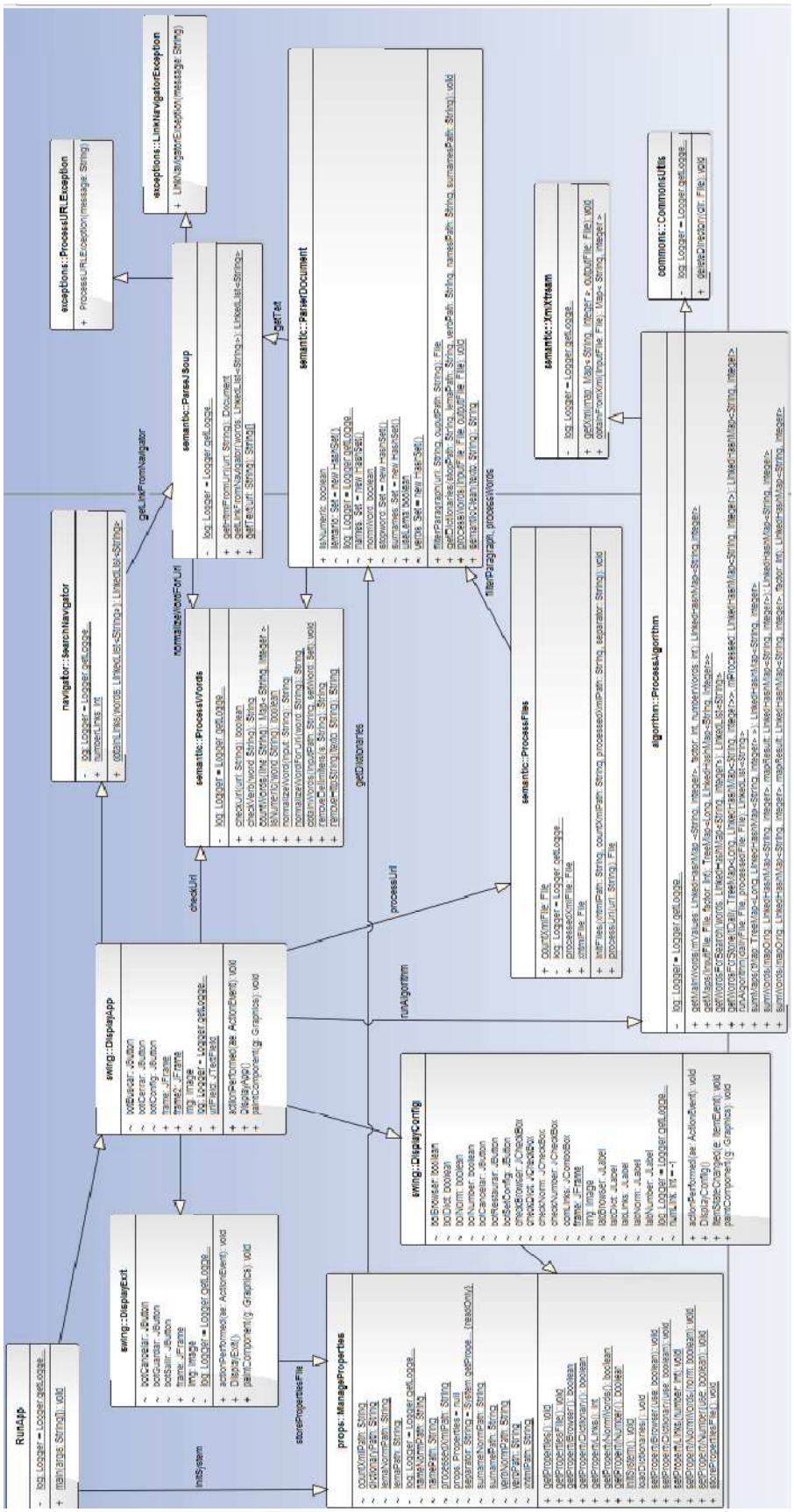


Figura 42- Diagrama UML del código del proyecto

2.6 Casos de uso / Pruebas

Una vez que se ha desgranado el funcionamiento completo del Trazador de Pistas, explicando cómo funciona el núcleo del mismo, es decir, el Sistema de Recomendación, y detallando las opciones de configuración posibles, se procede a enumerar los distintos casos de uso disponibles y los resultados obtenidos al llevarlos a cabo para probar el Trazador de Pistas.

Por cada caso de uso se expondrá una explicación sobre el escenario que lo forma, los pasos de ejecución a seguir, los resultados obtenidos, tanto en el Sistema de Recomendación (perfil de la página web e historial) como en el Trazador de Pistas (páginas web sugeridas) y las conclusiones acerca del caso de uso.

2.6.1 Secuencia ejecución Trazador con páginas similares

Escenario:

Partiendo desde cero en el Trazador de Pistas, esto es, con la configuración inicial por defecto que establece la aplicación y ningún dato almacenado en el historial, se realiza una secuencia de 5 búsquedas de páginas web de contenido similar.

La configuración de los parámetros será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí
- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón “Configurar parámetros”.
   2.1. Dentro de panel de configuración, pulsar botón “Eliminar datos previos”.
   2.2. Pulsar botón “Volver”.
3. Introducir en el cuadro de texto de búsqueda la URL: ***http://uc3m.es/Inicio***.
4. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
5. Comprobar resultados.
6. Seleccionar enlace sugerido: ***http://www.uax.es***
7. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
8. Comprobar resultados.
9. Seleccionar enlace sugerido: ***http://www.um.es/actualidad/prensa/noticia.php?id=1479651***
10. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
11. Comprobar resultados.
12. Seleccionar enlace sugerido: ***http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm***
13. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
14. Comprobar resultados.
15. Seleccionar enlace sugerido: ***http://www.ulpgc.es/index.php?pagina=campusvirtual&ver=inicio***
16. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
17. Comprobar resultados.

Resultados de ejecución:

Primera búsqueda ejecutada en paso 4 (***http://uc3m.es/Inicio***):

- Perfil de la página web: **universidad(10), campus(10), informacion(8), alumno(6), lunes(6), carlos(6), antiguo(6), octubre(6), madrid(6), aula(6)**.
- Palabras almacenadas como historial: **campus(10), universidad(10), informacion(8), aula(6), madrid(6), octubre(6), antiguo(6), carlos(6), lunes(6), alumno(6)**
- Palabras seleccionadas para lanzar búsqueda: **campus, universidad, informacion, aula, madrid**
- Links mostrados al usuario:
  http://www.upsa.es/facultades/detalle-facultad/index.php?idCtro=2506
  http://www.uc3m.es/
  http://www.uax.es/
  http://www.educaweb.com/centro/udima-universidad-distancia-madrid-18579/
  http://www.emes.es/Actualidad/Noticias/Noticia/uc3m/tabid/581/itemid/4749/type/noticia/Default.aspx

Segunda búsqueda ejecutada en paso 6 (***http://www.uax.es/***):

- Perfil de la página web: **campus(32), estudiante(22), grado(20), centro(18), servicio(16), universidad(16)**
- Palabras almacenadas como historial: **campus(42), universidad(26), estudiante(22), grado(20), centro(18), servicio(16), informacion(8), alumno(6), lunes(6), carlos(6), antiguo(6), octubre(6), madrid(6), aula(6)**
- Palabras seleccionadas para lanzar búsqueda: **campus, universidad, estudiante, grado, centro**
- Links mostrados al usuario:
  http://www.um.es/
  http://www.um.es/actualidad/prensa/noticia.php?id=1479651
  http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm
  http://www.upm.es/institucional/FuturosEstudiantes/Centros/CampusCiudadUniversitaria/RectoradoA
  http://enciclopedia.us.es/index.php/Bogot%C3%A1_(Cundinamarca)

Tercera búsqueda ejecutada en paso 9
(***http://www.um.es/actualidad/prensa/noticia.php?id=1479651)***:

- Perfil de la página web: **prensa(10), actualidad(8), universidad(6), agenda(6), noticia(6), actividad(6)**
- Palabras almacenadas como historial: **campus(42), universidad(32), estudiante(22), grado(20), centro(18), servicio(16), prensa(10), actualidad(8), informacion(8), actividad(6), noticia(6), agenda(6), aula(6), madrid(6), octubre(6), antiguo(6), carlos(6), lunes(6), alumno(6)**
- Palabras seleccionadas para lanzar búsqueda: **campus, universidad, estudiante, grado, centro**
- Links mostrados al usuario:
  http://www.um.es/
  http://www.um.es/actualidad/prensa/noticia.php?id=1479651
  http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm
  http://www.upm.es/institucional/FuturosEstudiantes/Centros/CampusCiudadUniversitaria/RectoradoA
  http://enciclopedia.us.es/index.php/Bogot%C3%A1_(Cundinamarca)

Cuarta búsqueda ejecutada en paso 12

(***http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm***):

- Perfil de la página web: **campus(68), sebastian(40), san(40), beca(26), grado(16)**
- Palabras almacenadas como historial: **campus(110), san(40), sebastian(40), grado(36), universidad(32), beca(26), estudiante(22), centro(18), servicio(16), prensa(10), informacion(8), actualidad(8), alumno(6), lunes(6), carlos(6), antiguo(6), octubre(6), madrid(6), aula(6), agenda(6), noticia(6), actividad(6)**
- Palabras seleccionadas para lanzar búsqueda: **campus, san, sebastian, grado, universidad**
- Links mostrados al usuario:
  http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm
  http://www.upm.es/institucional/FuturosEstudiantes/Centros/CampusMontegancedo/ETSIngenierosInformaticos
  http://www.ulpgc.es/index.php?pagina=campusvirtual&ver=inicio
  http://es.wikipedia.org/wiki/Universidad_de_Sevilla
  http://www.emagister.com/master/master-derecho-civil-tematica-763_3.htm

Quinta búsqueda ejecutada en paso 15
(***http://www.ulpgc.es/index.php?pagina=campusvirtual&ver=inicio***):

- Perfil de la página web: **palma(16), canaria(14), mapa(12), servicio(12), gran(10), examen(10)**
- Palabras almacenadas como historial: **campus(110), sebastian(40), san(40), grado(36), universidad(32), servicio(28), beca(26), estudiante(22), centro(18), palma(16), canaria(14), mapa(12), examen(10), gran(10), prensa(10), actualidad(8), informacion(8), actividad(6), noticia(6), agenda(6), aula(6), madrid(6), octubre(6), antiguo(6), carlos(6), lunes(6), alumno(6)**
- Palabras seleccionadas para lanzar búsqueda: **campus, sebastian, san, grado, universidad**
- Links mostrados al usuario:
  http://www.deusto.es/DEUSTO/MapaWebHomeUD_cast.htm
  http://www.upm.es/institucional/FuturosEstudiantes/Centros/CampusMontegancedo/ETSIngenierosInformaticos
  http://www.ulpgc.es/index.php?pagina=campusvirtual&ver=inicio
  http://es.wikipedia.org/wiki/Universidad_de_Sevilla
  http://www.emagister.com/master/master-derecho-civil-tematica-763_3.htm

Conclusiones:

Se observa cómo el Trazador de Pistas va evolucionando dentro de un mismo ámbito, en este caso el campus universitario. Sin embargo, el peso que tenga cada palabra dentro del perfil que lo va componiendo, también es relevante para componer el historial, así, se observa que las palabras “san sebastian” tienen más peso en el perfil de la página web de la Universidad de Deusto que las palabra “madrid” en el perfil de la página web de la Universidad Carlos III, por lo que finalmente adquieren más relevancia a nivel global dentro del historial.

2.6.2 Secuencia ejecución Trazador con páginas no similares

Escenario:

Partiendo desde cero en el Trazador de Pistas, esto es, con la configuración inicial por defecto que establece la aplicación y ningún dato almacenado en el historial, se realiza una secuencia de 5 búsquedas de páginas web de contenido diferente entre sí.

La configuración de los parámetros será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí
- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón "Configurar parámetros".
   2.1. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   2.2. Pulsar botón "Volver".
3. Introducir en el cuadro de texto de búsqueda la URL: ***http://es.wikipedia.org/wiki/N%C3%BAmero_at%C3%B3mico***.
4. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
5. Comprobar resultados.
6. Volver a pantalla principal e introducir en el cuadro de texto de búsqueda la URL: ***http://webosfritos.es.***
7. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
8. Comprobar resultados.
9. Volver a pantalla principal e introducir en el cuadro de texto de búsqueda la URL: ***http://www.vogue.es/moda***.
10. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
11. Comprobar resultados.
12. Volver a pantalla principal e introducir en el cuadro de texto de búsqueda la URL: ***http://uc3m.es/Inicio***
13. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
14. Comprobar resultados.
15. Volver a pantalla principal e introducir en el cuadro de texto de búsqueda la URL: ***http://www.colorear.info/Dibujos-infantiles/***
16. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
17. Comprobar resultados.

Resultados de ejecución:

Primera búsqueda ejecutada en paso 3 (***http://es.wikipedia.org/wiki/N%C3%BAmero_at%C3%B3mico***):

- Perfil de la página web: **numero(30), atomico(20), proton(20), atomo(16), electron(14), elemento(14), pagina(14)**
- Palabras almacenadas como historial: **numero(30), proton(20), atomico(20), atomo(16), pagina(14), elemento(14), electron(14)**
- Palabras seleccionadas para lanzar búsqueda: **numero, proton, atomico, atomo, pagina**
- Links mostrados al usuario:
  http://enciclopedia.us.es/index.php/Grupo_carboxilo
  http://es.wikipedia.org/wiki/Modelo_at%C3%B3mico_de_Bohr
  http://www.filosofia.org/enc/cc1/cc1001.htm
  http://www.ecured.cu/index.php/Reactividad
  http://html.rincondelvago.com/geologia-historica.html

Segunda búsqueda ejecutada en paso 6 (***http://webosfritos.es***):

- Perfil de la página web: **receta(36), frito(12), cuchara(10), pan(8), verde(8)**

- Palabras almacenadas como historial: **receta(36), numero(30), atomico(20), proton(20), atomo(16), electron(14), elemento(14), pagina(14), frito(12), cuchara(10), verde(8), pan(8**
- Palabras seleccionadas para lanzar búsqueda: **receta, numero, atomico, proton, atomo**
- Links mostrados al usuario:
  http://es.wikipedia.org/wiki/Modelo_at%C3%B3mico_de_Bohr
  http://www.ecured.cu/index.php/Reactividad
  http://enciclopedia.us.es/index.php/Grupo_carboxilo
  http://html.rincondelvago.com/geologia-historica.html
  http://www.nuevatribuna.es/articulo/america-latina/bombas-y-default/20140801140124105671.html

Tercera búsqueda ejecutada en paso 9 (***www.vogue.es/moda***):

- Perfil de la página web: **espia(18), paris(18), dia(14), mario(12), moda(10), tendencia(10)**
- Palabras almacenadas como historial: **receta(36), numero(30), proton(20), atomico(20), paris(18), espia(18), atomo(16), dia(14), pagina(14), elemento(14), electron(14), mario(12), frito(12), tendencia(10), moda(10), cuchara(10), pan(8), verde(8)**
- Palabras seleccionadas para lanzar búsqueda: **receta, numero, proton, atomico, paris**
- Links mostrados al usuario:
  http://es.wikipedia.org/wiki/Elemento_qu%C3%ADmico
  No hay más resultados para mostrar

Cuarta búsqueda ejecutada en paso 12 (***http://uc3m.es/Inicio***):

- Perfil de la página web: **universidad(10), campus(10), informacion(8), alumno(6), lunes(6), carlos(6), antiguo(6), octubre(6), madrid(6), aula(6)**
- Palabras almacenadas como historial: **receta(36), numero(30), atomico(20), proton(20), espia(18), paris(18), atomo(16), electron(14), elemento(14), pagina(14), dia(14), frito(12), mario(12), campus(10), universidad(10), cuchara(10), moda(10), tendencia(10), informacion(8), verde(8), pan(8)**
- Palabras seleccionadas para lanzar búsqueda: **receta, numero, atomico, proton, espia**
- Links mostrados al usuario:
  http://www.cubarepresorid.com/2012/02/punto-cu-esta-les-va-por-zapata-tamayo.html
  No hay más resultados para mostrar

Quinta búsqueda ejecutada en paso 15
(***http://www.colorear.info/Dibujos-infantiles/***):

- Perfil de la página web: **colorear(42), dibujo(40), imagen(8), imprimir(8), gato(4), dibujar(4), pintar(4), top(4), carnaval(4), pagina(4), infantil(4), lobo(4)**
- Palabras almacenadas como historial: **colorear(42), dibujo(40), receta(36), numero(30), proton(20), atomico(20), pagina(18), paris(18), espia(18), atomo(16), dia(14), elemento(14), electron(14), mario(12), frito(12), tendencia(10), moda(10), cuchara(10), universidad(10), campus(10)**
- Palabras seleccionadas para lanzar búsqueda: **colorear, dibujo, receta, numero, proton**

- Links mostrados al usuario:
  http://actividades.parabebes.com/dibujo-colorear-princess-29_ef258.html
  http://www.tarotsms.es/recetas.htm
  http://www.yodibujo.es/index.php
  http://www.encuentos.com/embarazo/alimentacion-dietas-y-recetas-de-cocina-para-embarazadas/
  http://es.wikipedia.org/wiki/Citoplasma

Conclusiones:

Como se observa en la evolución de las búsquedas, al realizar búsquedas tan dispares de contenido semántico entre sí, el historial va almacenando palabras relativas a todas ellas, si bien aquéllas cuyo peso es menor, en las últimas búsquedas terminan desapareciendo del historial, mientras que aquéllas cuyo peso es elevando dentro del perfil de la página web, como es la palabra *proton*, mantienen su vigencia durante todo el proceso de búsqueda.

Se aprecia por tanto, cómo el Sistema de Recomendación va realizando un aprendizaje, a pesar de tener que manejar contenido tan dispar entre sí.

En cuanto a los resultados ofrecidos por el Trazador de Pistas, se observa cómo al introducirle páginas que no poseen ninguna relación entre sí, el set de palabras clave a lanzar al buscador contiene palabras que distan mucho entre sí a nivel semántico, y por lo tanto, el buscador no es capaz de encontrar contenido en la Red suficiente que abarque dicho set de palabras.

2.6.3 Comparativa página web filtrando contenido/sin filtrar contenido:

Escenario:

Se prueba ahora la diferencia de resultados obtenidos sobre una misma página web cuando se restringe el uso de contenido mal formado o neologismos o permitiendo cualquier tipo de palabra. Para ello se utilizarán las opciones “Normalizar palabras” y “Utilizar diccionarios precargados”. Se partirá de la configuración por defecto del Trazador de Pistas, se realizará una búsqueda, se modificará el valor del parámetro a probar, se restaurarán los datos almacenados en la aplicación y se volverá a realizar la misma búsqueda.

Se ha utilizado un blog de moda como ejemplo, por el tipo de lenguaje que utiliza y los comentarios que los lectores escriben.

La configuración de los parámetros inicial será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí
- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

El cambio de configuración que se realizará será:

- Normalizar palabras: no
- Utilizar diccionarios precargados: no

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón “Configurar parámetros”.
   2.1. Dentro de panel de configuración, pulsar botón “Eliminar datos previos”.
   2.2. Pulsar botón “Volver”.
3. Introducir en el cuadro de texto de búsqueda la URL:
   http://time-for-fashion.blogs.elle.es/2014/10/03/aw-2014-15-trends-sweatpants/.

4. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
5. Comprobar resultados.
6. Volver a pantalla principal y pulsar botón “Configurar parámetros”.
   6.1. Desmarcar opción “Normalizar palabras”.
   6.2. Desmarcar opción “Utilizar diccionarios precargados”.
   6.3. Pulsar botón “Guardar cambios”.
   6.4. Dentro de panel de configuración, pulsar botón “Eliminar datos previos”.
   6.5. Pulsar botón “Volver”.
7. Introducir en el cuadro de texto de búsqueda la URL: ***http://time-for-fashion.blogs.elle.es/2014/10/03/aw-2014-15-trends-sweatpants/***.
8. Pulsar botón “Buscar” o tecla “Enter” en el teclado.
9. Comprobar resultados.

Resultados de ejecución:

Primera búsqueda ejecutada en paso 3 (***http://time-for-fashion.blogs.elle.es/2014/10/03/aw-2014-15-trends-sweatpants/***):

- Perfil de la página web: **blog(42), octubre(40), septiembre(22), moda(22), pantalon(16)**
- Palabras almacenadas como historial: **blog(42), octubre(40), septiembre(22), moda(22), pantalon(16)**
- Palabras seleccionadas para lanzar búsqueda: **blog, octubre, moda, septiembre, pantalon**
- Links mostrados al usuario:
  http://www.blogymoda.com/moda/27801/victoriabeckham/nuevayork/primaveraverano2013
  http://lepisma.liblit.com/
  http://lepisma.liblit.com/2014/09/29/rayco-pulido-nela/
  http://operasiempre.lacoctelera.net/post/2006/09/12/aquien-es-mejor-reina-la-noche-
  http://antonsaavedra.wordpress.com/2014/07/02/futbol-y-movimiento-obrero/?replytocom=1014

Segunda búsqueda ejecutada en paso 7 (***http://time-for-fashion.blogs.elle.es/2014/10/03/aw-2014-15-trends-sweatpants/***):

- Perfil de la página web: **octubre(40), style(34), the(22), septiembre(22), moda(22), blogs(22)**
- Palabras almacenadas como historial: **octubre(40), style(34), the(22), septiembre(22), moda(22), blogs(22)**
- Palabras seleccionadas para lanzar búsqueda: **octubre, style, blogs, moda, septiembre**
- Links mostrados al usuario:
  http://www.telva.com/blogs/leccion-maquillaje/2014/07/15/elige-el-maquillaje-que-va-con-tu-tono.html
  http://www.laverdad.es/murcia/comarcas/?buscaedicion=murcia&reducido=1&pag=2
  http://www.rosapocacosa.com/search/label/concurso%20blogs
  http://www.forospanama.com/calendar.php?s=8c48a29e5af5bb27ab4a7aef05f18a67
  http://www.mujerhoy.com/belleza/blog-maquillaje-estrellas/diez-trucos-para-maquillaje-801385052014.html

Conclusiones:

Se observa cómo la normalización y el uso de diccionarios modifican los resultados, tanto del perfil de palabras, como del historial y las páginas web similares ofrecidas, pues no se restringe el uso de palabras referentes a otros idiomas.

2.6.4 Comparativa página web usando lemarios/sin usar lemarios:

Escenario:

Se prueba ahora la diferencia de resultados obtenidos sobre una misma página web seleccionando la opción "Utilizar diccionarios precargados" y deseleccionando esta opción, para profundizar en el uso de lemarios en castellano para limitar las búsquedas. Para ello, se partirá de la configuración por defecto del Trazador de Pistas, se realizará una búsqueda, se modificará el valor del parámetro a probar, se restaurarán los datos almacenados en la aplicación y se volverá a realizar la misma búsqueda.

La página web que se ha seleccionado es un artículo de moda sobre la Fashion Week de Milán, para enfatizar el uso de palabras provenientes de otros idiomas y neologismos en el castellano.

La configuración de los parámetros inicial será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí
- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

El cambio de configuración que se realizará será:

- Utilizar diccionarios precargados: no

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón "Configurar parámetros".
   2.1. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   2.2. Pulsar botón "Volver".
3. Introducir en el cuadro de texto de búsqueda la URL: ***http://www.telva.com/2014/09/18/pasarelas/1411039062.html***.
4. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
5. Comprobar resultados.
6. Volver a pantalla principal y pulsar botón "Configurar parámetros".
   6.1. Desmarcar opción "Utilizar diccionarios precargados".
   6.2. Pulsar botón "Guardar cambios".
   6.3. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   6.4. Pulsar botón "Volver".
7. Introducir en el cuadro de texto de búsqueda la URL: ***http://www.telva.com/2014/09/18/pasarelas/1411039062.html***.
8. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
9. Comprobar resultados.

Resultados de ejecución:

Primera búsqueda ejecutada en paso 3 (***http://www.telva.com/2014/09/18/pasarelas/1411039062.html***):

- Perfil de la página web: **verano(66), primavera(66), boda(26), milan(24), otoño(18), madrid(18)**

- Palabras almacenadas como historial: **primavera(66), verano(66), boda(26), milan(24), madrid(18), otoño(18)**
- Palabras seleccionadas para lanzar búsqueda: **primavera, verano, boda, milan, madrid**
- Links mostrados al usuario:
  http://www.telva.com/
  http://telva.feedsportal.com/c/33597/f/588978/s/3f1ee5cb/sc/26/l/0L0Stelva0N0C20A140C10A0C0A40Cnovias0C14124133710Bhtml/story01.htm
  http://elpais.com/tag/confeccion/a/
  http://teleprograma.fotogramas.es/tele-corazon/2014/noticias/noticias-2-de-junio/silvestre
  http://cultura.elpais.com/tag/italia/a/

Segunda búsqueda ejecutada en paso 7 (***http://www.telva.com/2014/09/18/pasarelas/1411039062.html***):

- Perfil de la página web: **verano(66), primavera(66); fashion(52), week(50), boda(26)**
- Palabras almacenadas como historial: **verano(66), primavera(66); fashion(52), week(50), boda(26)**
- Palabras seleccionadas para lanzar búsqueda: **primavera, verano, fashion, week, boda**
- Links mostrados al usuario:
  http://www.telva.com/
  http://telva.feedsportal.com/c/33597/f/588978/s/3f1ee5cb/sc/26/l/0L0Stelva0N0C20A140C10A0C0A40Cnovias0C14124133710Bhtml/story01.htm
  http://blogdelatele.blogspot.com/2014/09/kendall-jenner-modelo-revelacion-en-la.html
  http://elpais.com/tag/confeccion/a/
  http://www.revistaohlala.com/1731627-lo-mejor-de-paris-fashion-week

Conclusiones:

Una vez más, se observa cómo la utilización o no de lemarios precargados puede resultar interesante, o limitar el significado semántico de la página web que se evalúa, dependiendo del ámbito en el que esté enmarcada. Por tanto, resulta interesante que el usuario tenga la opción de poder configurar ciertos parámetros para ayudar al Sistema de Recomendación en su aprendizaje, en función de los intereses del mismo.

2.6.5 Comparativa página web usando caracteres numéricos:

Escenario:

Se prueba ahora la diferencia de resultados obtenidos sobre una misma página web seleccionando la opción "Tener en cuenta caracteres numéricos" y deseleccionando esta opción. Para ello, se partirá de la configuración por defecto del Trazador de Pistas, se realizará una búsqueda, se modificará el valor del parámetro a probar, se restaurarán los datos almacenados en la aplicación y se volverá a realizar la misma búsqueda.

Se escoge un blog que enseña una receta como muestra, pues contienen cantidades de los ingredientes y comentarios de los lectores.

La configuración de los parámetros inicial será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí

- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

El cambio de configuración que se realizará será:

- Tener en cuenta caracteres numéricos: sí

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón "Configurar parámetros".
   2.1. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   2.2. Pulsar botón "Volver".
3. Introducir en el cuadro de texto de búsqueda la URL: ***http://webosfritos.es/2014/07/bizcocho-de-yogur/***.
4. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
5. Comprobar resultados.
6. Volver a pantalla principal y pulsar botón "Configurar parámetros".
   6.1. Marcar opción "Tener en cuenta caracteres numéricos".
   6.2. Pulsar botón "Guardar cambios".
   6.3. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   6.4. Pulsar botón "Volver".
7. Introducir en el cuadro de texto de búsqueda la URL: ***http://webosfritos.es/2014/07/bizcocho-de-yogur/***.
8. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
9. Comprobar resultados.

Resultados de ejecución:

Primera búsqueda ejecutada en paso 3 (***http://webosfritos.es/2014/07/bizcocho-de-yogur/***):

- Perfil de la página web: **responder(162), julio(148), bizcocho(74), receta(50), gracias(44)**
- Palabras almacenadas como historial: **responder(162), julio(148), bizcocho(74), receta(50), gracias(44)**
- Palabras seleccionadas para lanzar búsqueda: **responder, julio, bizcocho, receta, gracias**
- Links mostrados al usuario:
  http://www.recetascocina.cl/recetas/pie-de-limon.html
  http://fronterad.com/?q=julio-cortazar-y-paris-%E2%80%98ultimo-round%E2%80%99-pasaje-al-centro-mandala
  http://www.gastronomiaycia.com/2014/05/17/brownie-cheescake/
  http://www.saludynutricion.es/
  http://es.wikipedia.org/wiki/Sildenafilo

Segunda búsqueda ejecutada en paso 7 (***http://webosfritos.es/2014/07/bizcocho-de-yogur/***):

- Perfil de la página web: **2014(170), responder(162), julio(148), bizcocho(72), receta(50)**
- Palabras almacenadas como historial: **2014(170), responder(162), julio(148), bizcocho(72), receta(50)**
- Palabras seleccionadas para lanzar búsqueda: **2014, responder, julio, bizcocho, receta**
- Links mostrados al usuario:
  http://www.rosapocacosa.com/2013/06/sabado-noche.html

http://www.balovega.com/
http://www.balovega.com/2013/09/el-espiritu-de-lazarillo-en-salamanca.html
http://www.balovega.com/2013/01/un-angel-como-regalo-de-los-magos.html
http://www.20minutos.es/noticia/2160711/0/

Conclusiones:

Se observa nuevamente cómo la inclusión o no de ciertos parámetros de configuración, en este caso el uso de caracteres numéricos, afecta tanto en el Sistema de Recomendación, pues el perfil de la página web cambia, como en el resultado que muestra el Trazador de Pistas, pues al cambiar el perfil, y por tanto, el historial, las palabras que se lanzan al buscador son diferentes y generan un resultado diferente. En este caso, al ser la página a analizar un blog cuyas entradas y comentarios están etiquetados con la fecha en la que se realizan, 2014 es una de las palabras relevantes que componen el perfil, aunque no proporciona información semántica acerca del contenido de la entrada del blog.

2.6.6 Modificar parámetros personalización

Escenario:

Se prueban ahora los parámetros de configuración que permiten al usuario personalizar la apariencia de los resultados mostrados por el Trazador de Pistas. Estos parámetros son "Número de links mostrados" y "Mostrar URL seleccionada en navegador web". Para ello, se configurarán inicialmente estos valores a unos valores distintos a los precargados inicialmente en la aplicación, y después se realizará una búsqueda.

La configuración de los parámetros inicial será:

- Número de links mostrados: 5
- Normalizar palabras: sí
- Utilizar diccionarios precargados: sí
- Tener en cuenta caracteres numéricos: no
- Mostrar URL seleccionada en navegador web: no

El cambio de configuración que se realizará será:

- Número de links mostrados: 50
- Mostrar URL seleccionada en navegador web: sí

Pasos a seguir:

1. Abrir aplicación.
2. Pulsar botón "Configurar parámetros".
   2.1. Dentro de panel de configuración, pulsar botón "Eliminar datos previos".
   2.2. Marcar opción "Mostrar URL seleccionada en navegador web".
   2.3. Desplegar menú "Número de links mostrados" y seleccionar 50.
   2.4. Pulsar botón "Guardar cambios."
   2.5. Pulsar botón "Volver".
3. Introducir en el cuadro de texto de búsqueda la URL:
4. ***http://es.wikipedia.org/wiki/N%C3%BAmero_at%C3%B3mico***
5. Pulsar botón "Buscar" o tecla "Enter" en el teclado.
6. Comprobar resultados.
7. Seleccionar un resultado.

Resultados de ejecución:

Links mostrados al usuario:
http://enciclopedia.us.es/index.php/%C3%81tomo
http://www.filosofia.org/enc/cc1/cc1001.htm

http://es.wikipedia.org/wiki/Modelo_at%C3%B3mico_de_Bohr
http://www.ecured.cu/index.php/Reactividad
http://html.rincondelvago.com/geologia-historica.html
http://www.ib.edu.ar/index.php/el-balseiro/historia-del-ib/informes-sobre-las-actividades-en-la-isla-huemul.html
http://curiosoperoinutil.com/esencia-de-cpi/
http://www.cmo.org.mx/captacionreactivos/
http://enciclopedia.us.es/index.php/N%C3%BAcleo_at%C3%B3mico
http://es.wikipedia.org/wiki/F%C3%ADsica_nuclear
http://es.wikipedia.org/wiki/Carbono-12
http://es.wikipedia.org/wiki/Constante_de_Rydberg
http://es.wikipedia.org/wiki/Ion
http://es.wikipedia.org/wiki/Fisi%C3%B3n_nuclear
http://es.wikipedia.org/wiki/Resonancia_magn%C3%A9tica_nuclear
http://es.wikipedia.org/wiki/Elemento_qu%C3%ADmico
http://es.wikipedia.org/wiki/Radiaci%C3%B3n
http://es.wikipedia.org/wiki/Radiaci%C3%B3n_ionizante
http://es.wikipedia.org/wiki/Fl%C3%BAor
http://es.wikipedia.org/wiki/Sistema
http://es.wikipedia.org/wiki/Cloro
http://es.wikipedia.org/wiki/Sal_(qu%C3%ADmica)
http://es.wikipedia.org/wiki/Fuerza
http://es.wikipedia.org/wiki/Unidad_de_medida
http://enciclopedia.us.es/index.php/Is%C3%B3topo_estable
http://enciclopedia.us.es/index.php/Iridio
http://enciclopedia.us.es/index.php/Is%C3%B3topo
http://enciclopedia.us.es/index.php/N%C3%BAclido
http://enciclopedia.us.es/index.php/RMN_en_el_analisis_de_la_estructura_qu%C3%ADmica
http://enciclopedia.us.es/index.php/Grupo_carboxilo
http://www.juegosflasher.com/juego-flash-MIDFIELD-MASTER.html
http://www.rcci.net/globalizacion/2005/fg542.htm
http://www.cihpress.com/2014/08/se-declara-en-huelga-de-hambre.html
http://www.monografias.com/trabajos95/proyectos-investigacion-cultural/proyectos-investigacion-cultural.shtml
http://www.facebookespanol.info/
http://www.cientec.or.cr/ciencias/AMfisica/index.html
http://industrial.udima.es/?author=2
http://relatos.leergratis.com/jose-alvarez-lopez-por-guillermo-borioli.html
http://socialesmoriles.blogspot.com.es/2012/09/tema-22-primera-cruzada-los-cruzados.html
http://librodenotas.com/faq
http://clio.rediris.es/tiemposmodernos/articulos/Numero1-2000-ISSN-1139-6237/MILENIOS.htm
http://jamillan.com/histoint.htm
http://cocinadesara.blogspot.com.es/
http://es.doblaje.wikia.com/wiki/Benjam%C3%ADn_Rivera
http://ayerbereccoarquitectos.blogspot.com.es/p/blog-page_26.html
http://www.ugr.es/%7Epwlac/G19_05DemetrioE_Brisset_Martin.html
http://www.observatorioprecios.es/p/metodologia.html

http://www.acedis.com/Master-en-Prevencion-de-Riesgos-Laborales-y-Experto-en-Energias-Renovables-1_1_72.html
http://www.lagallinapintadita.com/p/de-cuentos.html
http://angeldelalamo.blogspot.com/2006/06/fraunhofer-y-las-lneas-oscuras.html

Además, la URL seleccionada se abre en el navegador:

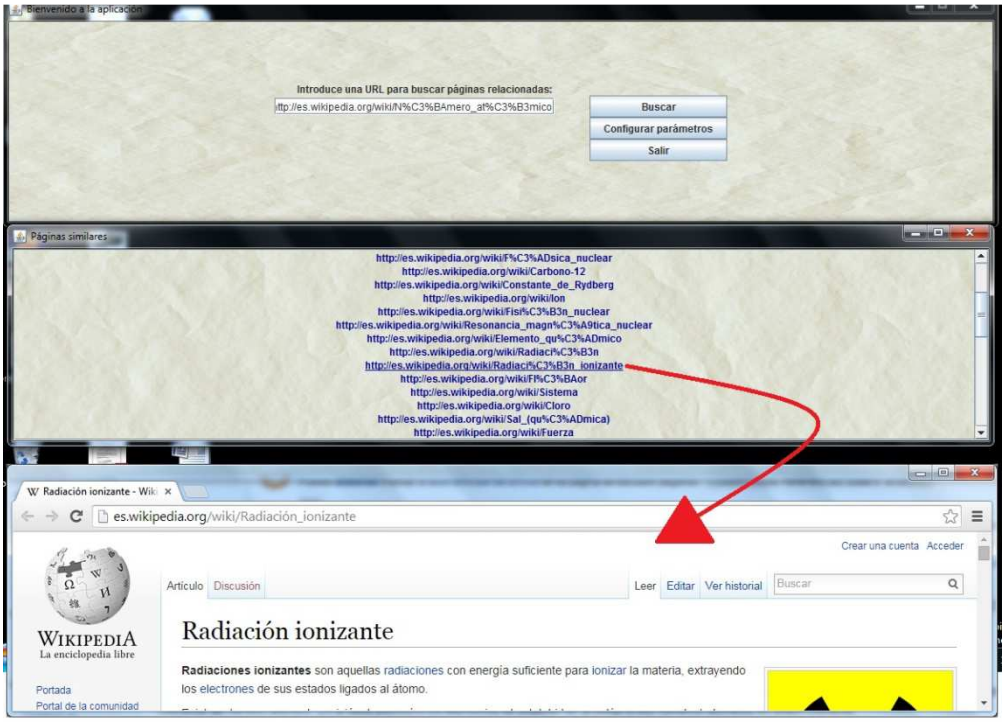


Figura 43- Ejemplo activar opción abrir URL en navegador web

Conclusiones:

Es posible adaptar los parámetros de salida del Trazador de Pistas, en función del uso y el interés del usuario. Si éste desea realizar navegaciones en la Red a través del Trazador, esta funcionalidad se encuentra disponible, así como poder obtener una visión más amplia o más sencilla de los posibles resultados ofrecidos a partir de una página web seleccionada.

3 Conclusiones y trabajos futuros

Después de finalizar el desarrollo del Trazador de Pistas, se exponen a continuación las reflexiones acerca de la consecución de los objetivos iniciales del proyecto y los resultados obtenidos. También se indicarán posibles líneas de trabajo futuras para la ampliación del proyecto.

3.1 Conclusiones

El propósito inicial de este proyecto era construir un Trazador de Pistas a partir de un Sistema de Recomendación basado en Contenido, para facilitar la navegación por la Red a determinados tipos de usuarios, con un manejo menos avanzado de la tecnología.

Para ello, se diferenciaron tres módulos que componían el Trazador de Pistas, que consistían en la obtención de la página web que el usuario visita, el análisis del contenido y extracción de la información relevante, y por último, la búsqueda en la Red de contenido similar para proponerlo al usuario.

Aunque se ha implementado un Trazador de Pistas completo, el objetivo principal de este proyecto era el análisis y la implementación del módulo central del mismo, diseñando un Sistema de Recomendación basado en Contenido personalizado cuyas tareas principales consisten en: análisis de la información, combinación con información previa y la elección de palabras para lanzar al buscador.

Por lo tanto, se puede afirmar que se ha conseguido el propósito de este proyecto, pues se ha desarrollado e implementado un Sistema de Recomendación que se corresponde con el núcleo central del Trazador de Pistas, que ofrece la funcionalidad deseada, y, adicionalmente, se lo ha dotado de diferentes aspectos de configuración, con los que usuarios más avezados en el manejo de Internet pueden evaluar el contenido de tipo semántico en las páginas.

Las conclusiones finales que se obtienen en el proceso de creación de un Trazador de Pistas son:

- Actualmente son necesarias herramientas como la que se ha diseñado en este proyecto que ayuden a los usuarios con el manejo de las nuevas tecnologías, pues existe todavía un porcentaje elevado de usuarios que hoy en día no es capaz de manejarse y navegar por la Red para obtener los resultados deseados cuando busca información. Esto se debe a varios factores:
  - Bien porque han crecido sin acceso a nuevas tecnologías y no están familiarizados con su uso, lo que provoca que su aprendizaje sea muy dificultoso.
  - Bien porque desconocen el funcionamiento de los buscadores y los mecanismos de posicionamiento de páginas web en la red actuales, de manera que, aunque tengan cierto manejo con las nuevas tecnologías para acceder a información básica en la Red, encontrar información más específica deseada se convierta en una difícil tarea.
  - También influye el hecho de que el contenido en la Red a menudo contiene información repetida y de baja calidad, con textos que no están bien etiquetados o errores ortográficos que impiden que los buscadores puedan enlazarlos fácilmente.
- Como se ha probado durante todo el proceso de implementación, desarrollar un Trazador de Pistas requiere de mecanismos que sean capaces de extraer las palabras con relevancia a nivel semántico dentro de una página web, para elaborar el perfil de la misma, y así poder buscar páginas web con contenido simi-

lar, por lo que es necesario tener claro en el momento de parsear un texto cuáles son las palabras que proporcionarán al sistema información semántica sobre la página web y cuáles es necesario eliminar del análisis.

- Combinar conocimientos sobre posicionamiento en Internet (*SEO*) con el algoritmo desarrollado para el Sistema de Recomendación para elegir el número de palabras proporciona al buscador un set de palabras más precisas para obtener enlaces lo más similares posibles y que sean del interés del usuario.
- Resulta necesario e interesante almacenar un historial de palabras relevantes perteneciente a búsquedas anteriores, para observar y tener en cuenta la evolución de los gustos del usuario, y elaborar así un Sistema de Recomendación más riguroso.
- La inclusión de parámetros configurables en el Trazador de Pistas, permite realizar análisis sobre el contenido en la Red que resultan muy interesantes a nivel lingüístico:
    - Primero, porque se puede comprobar cómo la información existente en la red a menudo contiene malformaciones y es de baja calidad, como se mencionaba en el primer punto de este apartado.
    - Y también, porque se puede observar cómo variando los parámetros de entrada del Sistema de Recomendación, éste aprende hacia un camino u otro, lo que proporciona al usuario resultados diferentes partiendo de una misma página web.

3.2 Trabajos futuros

Durante la realización del proyecto, se han ido detectando posibles líneas futuras de trabajo a tener en cuenta para la mejora del Trazador de Pistas implementado y para posibles ampliaciones en proyectos futuros. Se dividirán por tanto los trabajos futuros en dos partes:

Mejoras dentro del ámbito del actual Trazador de Pistas Implementado:

- Como se ha expuesto durante los diferentes capítulos de esta memoria, el Trazador de Pistas implementado está pensado para ser usado con páginas web escritas en idioma español, por lo que una mejora que se podría llevar a cabo sería la ampliación del Trazador a otros idiomas, mediante la inclusión de lemarios y ficheros *stop words* en otros idiomas. Resultaría interesante implementarlo al menos para el idioma inglés, puesto que es el idioma más usado en la Red, según demuestran las estadísticas proporcionadas por ***w3techs***(Web Technology Surveys). (42)
- Otro punto a tener en cuenta para su mejora es el del análisis de los tiempos verbales. El lemario incluido en la herramienta contiene verbos, pero sólo en modo infinitivo y participio, por lo que los demás tiempos verbales conjugados no son computados como palabras válidas. El Sistema de Recomendación que analiza el contenido semántico resultaría más completo si se elaborara una función que fuera capaz de distinguir que una palabra se corresponde con un tiempo verbal específico y contabilizar el infinitivo de dicho verbo, para eliminar palabras redundantes en el perfil de la página web.

Mejoras para la ampliación del actual Trazador de Pistas:

- Según se ha mencionado en varias ocasiones a lo largo de esta memoria, el objetivo final del Trazador de Pistas consistiría en la implementación de un sistema completo en el que el usuario sólo interviniera para seleccionar uno de los enlaces sugeridos por la herramienta, resultando transparente para él la ex-

tracción de las páginas web que visita mientras navega. Por lo tanto, una ampliación de este proyecto consistiría en implementar uno de los mecanismos que se analizaron en el *apartado 2.2.1* (Herramientas analizadas para la obtención del código HTML), que proporcionara al Sistema de Recomendación implementado en este proyecto el contenido de las páginas web, sin que el usuario se viera obligado a introducirlas en la herramienta.

- Por otro lado, si se llevara a cabo la mejora descrita anteriormente y se ampliara el Trazador a más de un idioma, sería aconsejable modificar el último módulo del Trazador, donde se realiza una consulta al buscador para obtener enlaces similares, pues el buscador utilizado, *Trovator*, está pensado para su uso en el idioma castellano, como se explicó en el *apartado 2.2.3.4,* reemplazándolo por otro buscador más generalista.

4 Presupuesto y tiempo dedicado

En este apartado de la memoria se detallará a continuación la planificación del proyecto, describiendo la lista de tareas que se han realizado hasta el desarrollo completo del Trazador de Pistas, y el tiempo estimado para cada una de ellas. Se incluye un presupuesto orientativo del coste del desarrollo del proyecto.

4.1 Tiempo dedicado

El tiempo efectivo dedicado en total ha sido de 16 meses, desde la definición de objetivos inicial hasta la finalización de la escritura de la memoria.

Se describen a continuación las tareas que se han realizado y el tiempo dedicado:

- Definición de objetivos: delimitar funcionalidad del Trazador de Pistas y sus características generales. Delimitar desarrollo del proyecto a la parte central del Trazador, el algoritmo que obtiene contenido relevante de la página web. Tiempo empleado: 1 mes.
- Diseño funcional del Trazador de Pistas completo: división en 3 módulos del Trazador (obtención contenido, Sistema de Recomendación y obtención resultados similares) y delimitación de objetivos para cada módulo. Tiempo empleado, 1 mes.
- Estudio de tecnologías: estudio de las posibles tecnologías a utilizar para los tres módulos definidos. En los casos donde procede, implementación de pruebas para confirmar o descartar tecnología. Tiempo dedicado: 4 meses.
- Diseño del módulo central del Trazador: diseño del algoritmo del Sistema de Recomendación que desarrolla la funcionalidad de la aplicación. Tiempo dedicado: 2 meses.
- Implementación del Sistema de Recomendación: tiempo dedicado, 4 meses.
- Diseño e implementación de los dos módulos restantes del Trazador de Pistas: se diseñan los módulos para poder obtener una primera aproximación del Trazador completo. Tiempo dedicado: 2 meses.
- Escritura y revisión de la memoria del proyecto: tiempo dedicado, 2 meses.

Se presenta a continuación un diagrama de Gantt con la planificación del proyecto, donde se exponen las tareas descritas con el tiempo asignado para cada una de ellas.

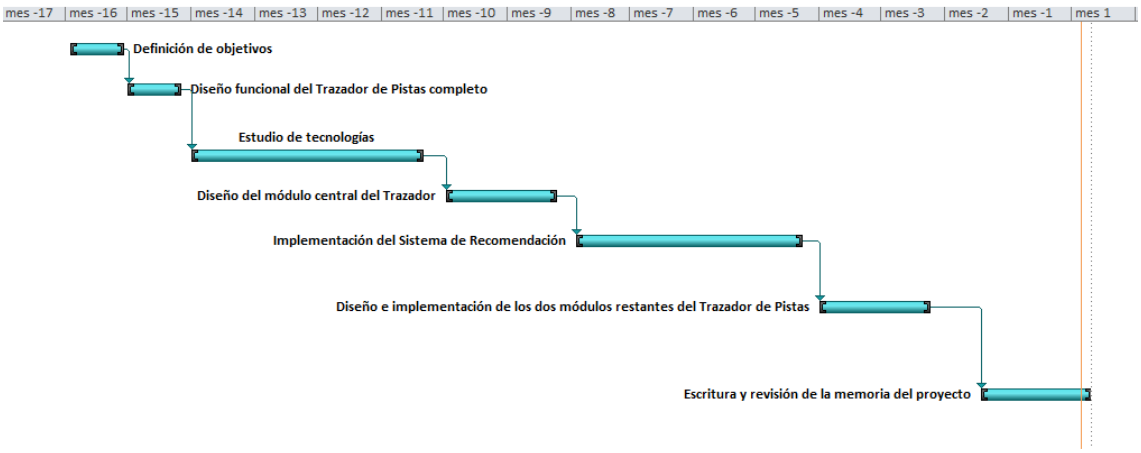


Figura 44- Diagrama de Gantt para planificación proyecto

4.2 Presupuesto

Se detalla a continuación cuál ha sido el coste del proyecto, partiendo de la planificación realizada en el punto anterior.

4.2.1 Costes de personal

Para realizar este cálculo es necesario incluir todas las tareas detalladas en el apartado anterior, teniendo en cuenta que este proyecto ha sido realizado únicamente por una persona, y considerando 21 días laborables al mes con una jornada de trabajo de 8 horas.

Sabiendo que el salario establecido en las plantillas de la universidad para un Ingeniero es de 2694,39€ por hombre y mes, y aplicando un ajuste del 70%, puesto que la autora de este proyecto todavía no está en posesión del título, los costes de personal son:

Nombre de tarea	Duración (meses)	Costes €	Costes Imputables €
Definición de objetivos	1	2.694,39	1.886,07
Diseño funcional del Trazador de Pistas completo	1	2.694,39	1.886,07
Estudio de tecnologías	4	10.777,56	7.544,29
Diseño del módulo central del Trazador	2	5.388,78	3.772,15
Implementación del Sistema de Recomendación	4	10.777,56	7.544,29
Diseño e implementación de los dos módulos restantes del Trazador de Pistas	2	5.388,78	3.772,15
Escritura y revisión de la memoria del proyecto	2	5.388,78	3.772,15
TOTAL	16	43.110,24	**30.177,17**

Tabla 1: Costes de personal

4.2.2 Costes de material

El material necesario para este proyecto ha sido un ordenador portátil personal, adquirido por 799€.

No ha habido costes asociados a software, ya que todas las tecnologías empleadas se han tratado de software libre.

Por lo tanto, sólo es necesario realizar el cálculo del coste de material para el equipo portátil. Para ello, se ha utilizado la siguiente fórmula de amortización extraída del Documento DOUE-L-2004-80145 del Boletín Oficial del Estado.

$$\frac{A \times C \times D}{B}$$

A = período en meses durante el cual está previsto utilizar el material para el proyecto, desde la fecha de entrega.
B = período de depreciación de 60 meses (36 meses en el caso de equipos informáticos cuyo precio sea inferior a 25.000 €);
C = coste del material sin IVA;
D = uso porcentual del material para el proyecto.
El IVA no recuperable abonado por el beneficiario se considerará gasto admisible

Figura 45- Fórmula de amortización del coste

El coste de material es:

Material	Coste €	% uso	Dedicación (meses)	Periodo depreciación	Coste Imputable €
Ordenador portátil	799	100	16	36	**355,11**

Tabla 2: Costes de material

4.2.3 Costes indirectos

Se calculan como el 20% del gasto total (coste de personal más coste de material), y engloban gastos como electricidad, conexión a Internet, etc.

En este caso ascienden a **6.106,45**€.

4.2.4 Coste total

El presupuesto total equivale al sumatorio de los costes directos más los indirectos, en este caso:

Gasto	Coste €
Coste personal	30.177,17
Coste material	355,11
Costes indirectos	6.106,46
Coste total	**36.638,73**

Tabla 3: Coste total

5 Bibliografía

1. Instituto Nacional de Estadística. [En línea] http://www.ine.es/prensa/np738.pdf.

2. Digital Agenda for Europe. [En línea] http://ec.europa.eu/information_society/newsroom/cf/dae/itemdetail.cfm?item_id=10099.

3. **Osti, Marina Vianello.** *El hipertexto entre la utopía y la aplicación: identidad, problemática y tendencias de la Web.* Gijón : TREA, 2004.

4. Hipertexto, el nuevo concepto de documento en la cultura de la imagen. [En línea] http://www.hipertexto.info/documentos/hipertexto.htm.

5. Wikipedia - Memex. [En línea] http://es.wikipedia.org/wiki/Memex.

6. Wikipedia - Trailblazer Project. [En línea] http://en.wikipedia.org/wiki/Trailblazer_Project.

7. History Commons. [En línea] http://www.historycommons.org/entity.jsp?entity=trailblazer_1.

8. Wikipedia - Recommender System. [En línea] http://en.wikipedia.org/wiki/Recommender_system.

9. Jarroba.com. [En línea] http://jarroba.com/que-son-los-sistemas-de-recomendacion/.

10. The Stanford University Infolab. [Online] http://infolab.stanford.edu/~ullman/mmds/ch9.pdf.

11. Economía, empresas y personas de León. [En línea] http://celempresas.blogspot.com.es/2011/06/distribucion-de-cola-larga-long-tail.html.

12. Gecko Plugin API Reference. [En línea] https://developer.mozilla.org/en-US/Add-ons/Plugins/Gecko_Plugin_API_Reference.

13. Mozilla Developer Network - Extensiones. [En línea] https://developer.mozilla.org/es/docs/Extensions.

14. Manifiesto de instalación. [En línea] https://developer.mozilla.org/es/docs/Manifiesto_de_instalaci%C3%B3n.

15. WikiLibros. *Mozilla Firefox.* [En línea] http://es.wikibooks.org/wiki/Mozilla_Firefox/Configuraci%C3%B3n/Crear_una_extensi%C3%B3n_para_Firefox_3.

16. Wikipedia - Dynamic Data Exchange. [En línea] http://es.wikipedia.org/wiki/Dynamic_Data_Exchange.

17. Wikipedia - Java Native Interface. [En línea] http://en.wikipedia.org/wiki/Java_Native_Interface.

18. Best practices for using the Java Native Interface. [En línea] http://www.ibm.com/developerworks/library/j-jni/.

19. Stack Overflow - Disadvantages of using Java Native Interface. [En línea] http://stackoverflow.com/questions/1393937/disadvantages-of-using-java-native-interface.

20. Teldat Router - HTTP Protocol. [En línea] http://www.it.uc3m.es/~teldat/TeldatC/ingles/protocols/Dm737-Iv10_HTTP_Protocol.PDF.

21. Analysis of a Privacy Vulnerability in the OpenID Protocol. [En línea] http://www.it.uc3m.es/muruenya/papers/MCSS10OpenID.pdf.

22. JTidy. [En línea] http://jtidy.sourceforge.net/.

23. Javarevisited - Difference between DOM and SAX Parsers in Java. [En línea] http://javarevisited.blogspot.com.es/2011/12/difference-between-dom-and-sax-parsers.html.

24. Stack Overflow - What is the difference between SAX and DOM? [En línea] http://stackoverflow.com/questions/6828703/what-is-the-difference-between-sax-and-dom.

25. JDOM. [En línea] http://www.jdom.org/docs/faq.html.

26. Easy Java/XML integration with JDOM, Part 1. [En línea] http://www.javaworld.com/article/2076294/java-se/easy-java-xml-integration-with-jdom--part-1.html.

27. Google España. [En línea] https://www.google.es/.

28. Stack Overflow - Can Jsoup simulate a button press? [En línea] http://stackoverflow.com/questions/7508813/can-jsoup-simulate-a-button-press.

29. Hispavista. [En línea] https://www.hispavista.com.

30. Lycos. [En línea] http://www.lycos.com/.

31. Enciclopedia Libre Universal en Español - Trovator. [En línea] http://enciclopedia.us.es/index.php/Trovator.

32. Trovator Buscador Español. [En línea] http://www.trovator.com/.

33. The Java Language Specification, Third Edition. [En línea] http://docs.oracle.com/javase/specs/jls/se5.0/html/j3TOC.html.

34. Java - Características y Ventajas. [En línea] http://www.oracle.com/es/technologies/java/features/index.html.

35. Project Swing (JavaTM Foundation Classes). [En línea] http://docs.oracle.com/javase/1.5.0/docs/guide/swing/.

36. The Java™ Tutorials - About the JFC and Swing. [En línea] http://docs.oracle.com/javase/tutorial/uiswing/start/about.html.

37. jsoup: Java HTML Parser. [En línea] http://jsoup.org/.

38. Apache log4j™ 1.2. [En línea] http://logging.apache.org/log4j/1.2/.

39. XStream. [En línea] http://xstream.codehaus.org/index.html.

40. Posicionamiento SEO para novatos: Las palabras clave y el Posicionamiento Google. [En línea] http://estrategias-marketing-online.com/posicionamiento-seo-para-novatos-las-palabras-clave-y-el-posicionamiento-google/.

41. Lemarios y listas de palabras del español. [En línea] http://olea.org/proyectos/lemarios/.

42. w3techs - Web Technology Surveys. [En línea] http://w3techs.com/.

6 Anexos

Se explicará a continuación cómo instalar el software del Trazador de Pistas y se proporcionará además un manual de usuario para manejar la aplicación.

6.1 Guía de instalación

Junto con la presente memoria, se incorpora el código compilado del proyecto.

Para ejecutar el Trazador de Pistas bastará con descomprimir el fichero ***Trailblazer.rar*** que se adjunta en un directorio con permisos de escritura, acceder a la carpeta Trailblazer que se creará, y hacer doble click sobre el fichero ***arranque.bat.***

Una vez realizados estos pasos, se abrirá una ventana ***cmd*** que ejecutará el código, mostrando la pantalla principal de la aplicación.

Nótese que para poder ejecutar la aplicación, es necesario tener instalado en el equipo una distribución **Java SDK 1.6** o superior.

6.2 Manual de usuario

Trailblazer es una herramienta que permite al usuario navegar fácilmente por la Red. Para ello, se le ha dotado de una interfaz gráfica sencilla e intuitiva. A su vez, el manejo de la herramienta es sencillo y no requiere de muchos conocimientos sobre las nuevas tecnologías, pues está pensada para acceder a todo tipo de público.

Este Trazador de Pistas, no solamente evalúa la página web actual en la que el usuario está interesado, sino que tiene en cuenta las páginas web en las que el usuario ha estado interesado anteriormente, así como la configuración por parte del usuario de ciertos parámetros que le permitirán ajustar los valores de búsqueda, tales como la inclusión o no de caracteres numéricos en la toma de decisiones del Trazador, etcétera.

Se explican a continuación las diferentes pantallas disponibles en la aplicación, junto con el cometido de cada una de ellas.

6.2.1 Pantalla principal

La pantalla principal es la que es desplegada al ejecutar la aplicación, y la que se encarga de gestionar todas las acciones posibles dentro del Trazador de Pistas. Su aspecto es el siguiente:

Figura 46- Pantalla principal de la aplicación

Esta pantalla consta de los siguientes elementos:

- Un ***campo de texto*** donde introducir la dirección o *URL* de la página web de la que se desea buscar páginas web parecidas.

- Un botón ***Buscar*** que activará el mecanismo que detecta la temática de la página web introducida y busca resultados similares.
- Un botón ***Configurar Parámetros*** que al ser pulsado despliega la ventana de configuración, donde el usuario puede personalizar los resultados de la búsqueda.
- Un botón ***Salir*** que permite al usuario abandonar la aplicación.

Introducir una dirección web en el sistema:

Para iniciar la búsqueda de páginas web similares, el usuario debe introducir primero la página web de la que quiere conocer contenido similar. Para ello debe introducir la dirección completa, siguiendo el formato:

http[s]://[www.]paginaejemplo.[com]/[es]/[etc]

Si el formato introducido no es el correcto, la aplicación mostrará un mensaje de error:

Figura 47- Mensaje dirección no válida

Realizar búsqueda:

Una vez que el usuario ha introducido una dirección web válida, para que la aplicación se ejecute deberá pulsar el botón ***Buscar*** o la tecla *Enter*.

Esta acción provocará que el Trazador de Pistas analice la dirección introducida y busque resultados similares. Si todo es correcto, una nueva ventana se desplegará, con los resultados seleccionados por la aplicación que son considerados similares a la dirección web introducida.

Sin embargo, si durante este proceso se produce algún tipo de error, una ventana de error será mostrada para advertir al usuario:

Si no existe acceso a la dirección web introducida, bien porque esté restringida, bien porque el usuario no tenga permiso para conectarse al dominio, la aplicación mostrará el siguiente mensaje de error:

Figura 48- Mensaje no conexión a Red

Si se ha producido un fallo mientras se realizaba la conexión con la dirección web introducida porque su servidor no esté disponible, se mostrará el siguiente mensaje:

Figura 49- Mensaje no conexión con URL página web sugerida

O bien este mensaje, si la conexión no se ha podido realizar con la página web del buscador:

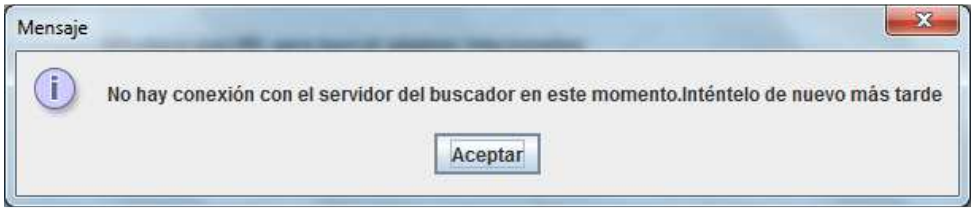


Figura 50- Mensaje no conexión con buscador web

Si la dirección introducida contiene una redirección, la aplicación mostrará el siguiente mensaje de error:

Figura 51- Mensaje URL redirección

Así mismo, si la aplicación no obtiene ningún resultado para mostrar, se mostrará el siguiente mensaje de error:

Figura 52- Mensaje no resultados a mostrar

Si se produce algún error mientras se analiza el contenido de la dirección introducida, la aplicación avisará al usuario de la siguiente manera:

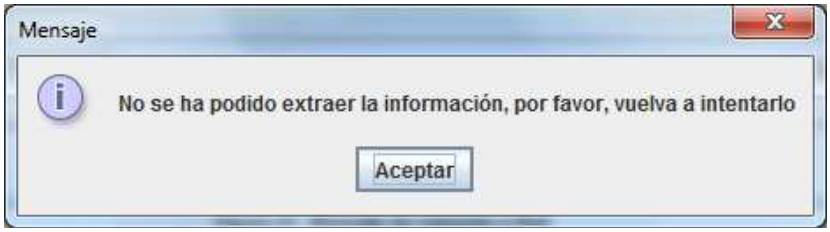


Figura 53- Mensaje error al detectar temática

Y si se produce algún error mientras se buscan resultados similares, la aplicación lo notificará con el siguiente mensaje:

Figura 54- Mensaje error al buscar resultados similares

6.2.2 Pantalla de resultados similares

Si el proceso de búsqueda ha sido correcto, la aplicación mostrará al usuario una nueva ventana que contendrá una lista de resultados similares al que introdujo el usuario:

Figura 55- Pantalla de páginas similares

Si el número de resultados obtenidos por la aplicación es menor que el número de links configurado en la aplicación para mostrar, la pantalla de resultados similares mostrará al final de todos los links disponibles un mensaje de texto indicando este hecho:

Figura 56- Pantalla resultados con menos resultados disponibles de los configurados

Si alguno de los resultados resulta del interés del usuario, bastará con pulsar en él para que la aplicación lo cargue automáticamente en el campo de texto de la pantalla principal. Además, si se ha configurado la aplicación para que el enlace seleccionado se abra en el Navegador Web también, se abrirá otra pantalla adicional en el navegador con el contenido de la dirección seleccionada.

Una vez seleccionado un enlace, esta pantalla se cerrará automáticamente.

Si por el contrario, ninguno de los enlaces mostrados resulta del interés del usuario, éste puede volver a la pantalla principal y realizar una nueva búsqueda. La pantalla de resultados se refrescará automáticamente con los nuevos resultados elegidos para mostrar al usuario.

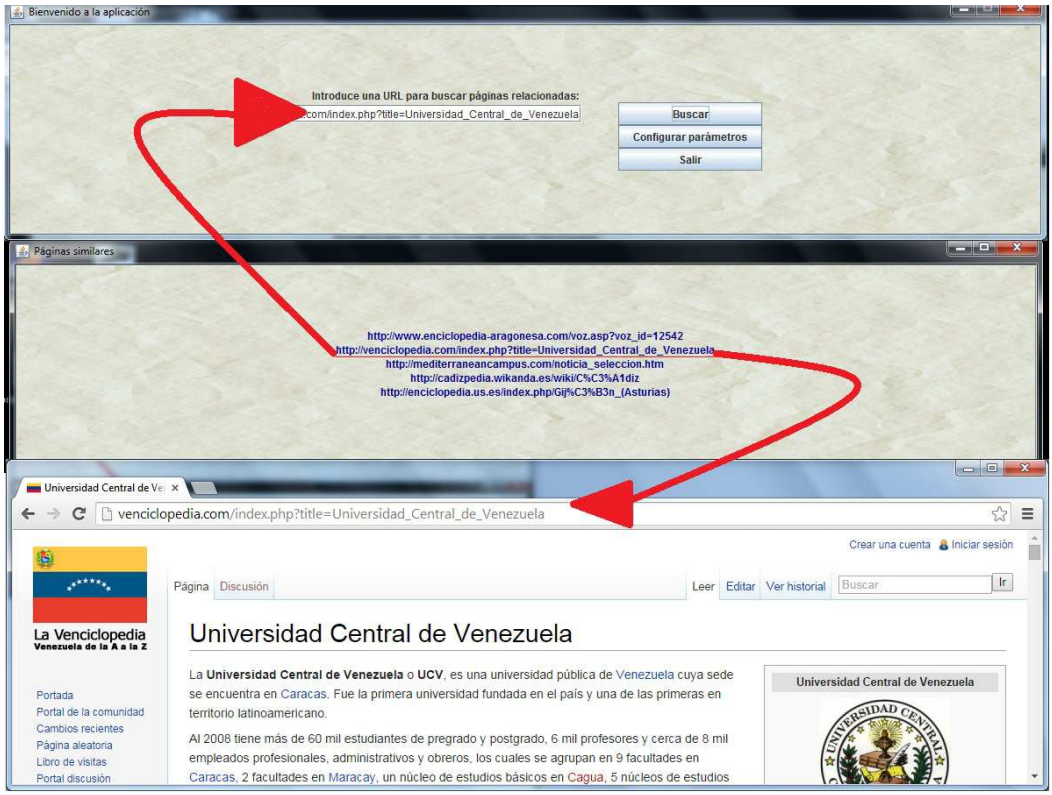


Figura 57- Selección de enlace en página de resultados

6.2.3 *Pantalla de Configuración del Sistema*

La pantalla de Configuración permite al usuario configurar ciertos aspectos del Trazador de Pistas, para influir en el resultado obtenido en la pantalla de páginas similares.

Esta pantalla tiene la siguiente apariencia:

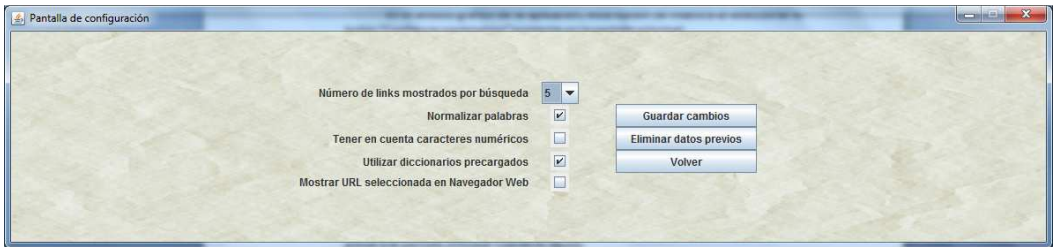


Figura 58- Pantalla Configuración aplicación

Se describen a continuación los parámetros que pueden ser modificados:

- ***Número de links mostrados por búsqueda***: El número de páginas web sugeridas que se mostrará en la aplicación en la pantalla de resultados similares. Por defecto se muestran 5.
- ***Normalizar palabras***: si se desea que la aplicación no tenga en cuenta las tildes y no haga distinciones entre mayúsculas y minúsculas. Por defecto se asumirá esta configuración.
- ***Tener en cuenta caracteres numéricos***: si se desea que los posibles caracteres numéricos que existan en la página web introducida sean tenidos en cuenta para elaborar la temática, o si por el contrario se desea obviarlos. Por defecto se obviarán estos caracteres.

- ***Utilizar diccionarios precargados***: el Trazador de Pistas incorpora diccionarios de palabras en castellano, nombres propios y apellidos para determinar el contenido de la página web. Si no se desea que se tengan en cuenta estos diccionarios, se deberá desmarcar esta opción, que por defecto se encuentra marcada en la aplicación.
- ***Mostrar URL seleccionada en Navegador Web***: al seleccionar esta opción, una vez que se elige un enlace en la página de resultados similares y se selecciona, además de cargar este enlace en la página principal para la siguiente búsqueda, se abrirá en el Navegador Web predefinido por defecto para mostrar su contenido. Esta opción no está activada en el Trazador de Pistas por defecto.

Una vez que se hayan seleccionado los nuevos parámetros de configuración, para guardar los cambios es necesario pulsar el botón "***Guardar cambios***". Estos cambios serán modificados sólo durante la sesión actual, a no ser que sean guardados en la aplicación al salir de la misma, como se indicará en el apartado siguiente.

Existe además otra opción disponible en esta ventana, llamada "***Eliminar datos previos***", que restaura elimina el contenido almacenado en la aplicación relativo a búsquedas anteriores, esto es, el historial de búsquedas que el usuario realizó. Con ello, en las siguientes búsquedas que se produzcan se tendrán en cuenta sólo los datos introducidos a partir de ese momento.

Una vez que se han configurado los parámetros deseados, limpiado datos previos, o simplemente si no se desea realizar ningún cambio, se pulsará el botón "***Volver***", que cerrará esta ventana y regresará al usuario a la pantalla principal.

Si el usuario no desea realizar ninguna modificación, la aplicación se ejecutará con la configuración establecida por defecto.

6.2.4 Pantalla Salir

Cuando el usuario ya no desee seguir utilizando el Trazador de Pistas, puede salir de la aplicación mediante la opción ***Salir*** en la pantalla principal. Se mostrará una ventana nueva con el siguiente aspecto:

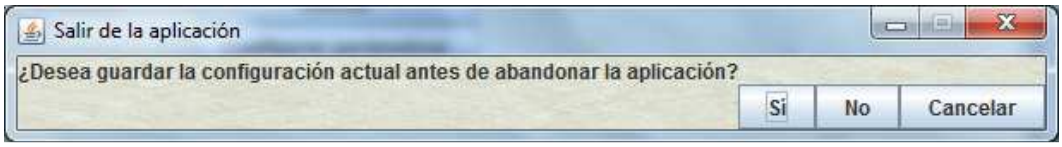


Figura 59- Pantalla Salir de la aplicación

Antes de abandonar la aplicación, el usuario tiene la opción de guardar permanentemente los cambios que haya realizado sobre los parámetros de la ventana de configuración, estableciendo una nueva configuración por defecto, o cerrar la aplicación sin guardar dicha configuración, de manera que la siguiente vez que se ejecute la aplicación se vuelva a mostrar la configuración anterior por defecto. Para ello, se han establecido dos botones:

- El botón ***Sí*** guardará la configuración y después cerrará la aplicación.
- El botón ***No,*** cerrará la aplicación sin guardar cambios.

Si realmente el usuario no desea abandonar la aplicación, deberá pulsar el botón ***Cancelar*** para regresar a la pantalla principal.