Aplicación para la obtención y análisis automático de noticias en el ámbito financiero

Francisco Javier Sanzol Sanz
Grado en Ingeniería Informática

Margarita Hospedales Salomó

Enero de 2014

FICHA DEL TRABAJO FINAL

Título del trabajo:	Aplicación para la obtención y análisis automático de noticias en el ámbito financiero
Nombre del autor:	Francisco Javier Sanzol Sanz
Nombre del consultor:	Margarita Hospedales Salomó
Fecha de entrega (mm/aaaa):	01/2014
Área del Trabajo Final:	Gestión del conocimiento
Titulación:	*Grado en Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

Este trabajo desarrolla el análisis, diseño e implementación de un prototipo, para un sistema de obtención y análisis automático de noticias, estando enfocado a un uso en el ámbito de los mercados financieros. El sistema planteado permitirá al usuario llevar a cabo cuatro grupos de funcionalidades principales. (1) Búsqueda y extracción de información de Internet mediante la suscripción a fuentes de contenidos utilizando *RSS/Atom* o bien mediante el desarrollo de estrategias de *crawling* de enlaces en Internet. (2) Obtención y gestión de datos de mercado, a partir de proveedores de datos en *streaming* o de descarga de datos históricos. (3) Construcción y gestión de documentos mediante las herramientas de indexado y búsqueda que provee el entorno Apache Lucene. (4) El desarrollo de estrategias de análisis de documentos incluyendo el uso de algoritmos de clasificación y agrupamiento.

Abstract (in English, 250 words or less):

This work develops the analysis, design and prototyping, of a system for the automatic extraction and analysis of news from the Internet; the use of the system will be within the scope of the financial markets. The system will allow the user to carry out the following four groups of functionalities: (1) Searching and extracting information from the Internet by using *RSS/Atom* content feed subscriptions or by developing strategies of Web crawling. (2) Subscribing and managing market data from data providers in streaming or by downloading historical data. (3) Building and managing documents by using the indexing and searching tools provided by Apache Lucene. (4) Developing strategies of analysis of documents, including algorithms of clustering and classification.

Palabras clave (entre 4 y 8):

Análisis automático de noticias, RSS/Atom, suscripción de contenidos, Web crawling, Indexado de documentos, mercados financieros.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

El análisis automático de noticias es la disciplina que utiliza técnicas de procesamiento del lenguaje natural para la extracción de atributos cualitativos y/o cuantitativos a partir de historias de textos no estructurados. La posibilidad de transformar textos sin estructura en atributos medibles que puedan ser procesados estadísticamente, permite tratar la información obtenida directamente de medios de comunicación, de una forma computerizada.

En el ámbito financiero, la influencia de las noticias y los artículos de opinión sobre la evolución de los mercados es un hecho bien conocido y que tradicionalmente ha atraído el interés de los inversores para la toma de decisiones [1]. Históricamente, uno de los primeros intentos por sistematizar el análisis de noticias, fue el prototipo desarrollado por Niederhoer, que utilizaba noticias diarias obtenidas en periódicos y las categorizaba en una escala de 19 clases, de positivas a negativas, con el objetivo de correlacionar estas clases con las evoluciones futuras de los mercados [2]. En la actualidad la posibilidad del tratamiento masivo de la información en forma digital unido al aumento de la capacidad computacional a disposición de los usuarios, ha permitido desarrollar sistemas automáticos para el procesamiento de la información, intensificando la experimentación en este campo dentro del contexto de la computación [1]. Normalmente el desarrollo de sistemas de análisis automático de noticias implica la utilización de software y algoritmos en los siguientes dominios:

1. Obtención de información: es responsable de hacer accesible al sistema los documentos *imput* que son posteriormente utilizados como fuente de datos. Puede consistir en componentes que permiten la subscripción a fuentes de noticias como agregadores de fuentes *RSS* o bien rastreadores que buscan información atendiendo a la ejecución de algún algoritmo (*crawlers*).
2. Pre-procesado de la información: consiste en extraer el contenido de los documentos recibidos como *imputs*. Implica tareas de interpretación de texto marcado, identificación de palabras y frases en base a vocabularios, etc…
3. Análisis y generación de conocimiento: es la parte del sistema que se encarga de transformar el *imput* del sistema en algún tipo de medición cualitativa o cuantitativa que permite un tratamiento estadístico de la

información. Se puede decir que da sentido a los datos en un contexto determinado por lo que genera conocimiento.

En los últimos años varias compañías han desarrollado sistemas de información consistentes en el procesado automático de noticias. Como producto al cliente suelen ofrecer contenidos procesados en forma de índices numéricos que permiten interpretar la información en forma cuantitativa. El sistema de Thomson Reuters NewsScope [3], es un *framework* integrado, que procesa noticias en tiempo real obtenidas a partir de agencias de noticias, y genera índices numéricos de eventos en un rango de 0 a 100; cada índice está etiquetado a una o varias temáticas de interés para los inversores. La compañía RavenPack [4], ofrece un sistema RavenPack News Scores que analiza de forma continua información relevante de varias fuentes de información incluyendo agencias de noticias y fuentes contrastadas de Internet (*blogs*, periodicos, etc…), y genera índices emocionales (*sentiment scores*) en tiempo real acerca de mercados, compañías o sectores de inversión. Otras empresas con similares servicios en el mercado son Dow Jones Newswires [5] o Bloomberg [6].

Por su alto coste, estos servicios son de interés para los inversores corporativos, sin embargo resultan poco accesibles al inversor individual y aficionado; normalmente su rentabilización depende de incorporarlos en proyectos que permitan una gran escalabilidad en los volúmenes de inversión. Una alternativa a estos sistemas de información de uso corporativo es el uso de la información libre disponible en Internet y que se ofrece de forma gratuita por medios de comunicación o servicios de análisis y opinión especializados. En este contexto es donde se engloba el objetivo del presente proyecto.

1.2 Objetivos del Trabajo

El objetivo de este proyecto es el prototipado de un sistema automático para la recuperación y análisis de noticias extraídas de Internet en el ámbito de los mercados financieros. Dado el carácter experimental del sistema, uno de los requerimientos más importantes es que sea flexible, en la incorporación para su evaluación, de diferentes tipos de algoritmos tanto en lo referente al componente con responsabilidad en la obtención de contenidos en Internet, así como el componente con responsabilidad en el análisis de la información y la extracción de conocimiento de las noticias.

1.3 Enfoque y método seguido

El sistema a desarrollar aglutina e interconecta una serie de funcionalidades que son comunes a otras aplicaciones ya existentes; algunos ejemplos son suscripción a noticias utilizando *RSS*, *crawling* de contenidos en Internet, suscripción a contenidos en *streaming*, o indexado y gestión de documentos. Para algunas de estas funcionalidades que son de uso común, existen implementaciones bien contrastadas cuya incorporación en el sistema facilita en gran medida el desarrollo. Por este motivo aunque la aplicación se desarrolla *de novo* se tratará de hacer un uso extensivo en la incorporación de

herramientas ya existentes y que en la medida de lo posible serán incorporadas en la forma de librerías externas de desarrollo. Una descripción más detallada de estas relaciones externas con otras aplicaciones se puede ver en el apartado 4.1 de la fase de desarrollo de producto.

1.4 Planificación del Trabajo

Alcance del proyecto

Se prevé que inicialmente este proyecto tenga un carácter de prototipado, cuyo principal objetivo sea el de permitir evaluar el interés de la aproximación planteada. En caso de que la evaluación del prototipo resulte favorable se verá la conveniencia de incorporar los componentes del sistema dentro de otra aplicación de mayor alcance actualmente en desarrollo y que permite la gestión de carteras de inversión y el desarrollo de *trading* algorítmico.

Se prevé que la aplicación pueda desarrollar las siguientes funcionalidades generales:

1. Búsqueda e indexado de *URLs* correspondientes a contenidos Web, principalmente noticias y opinión, en base a criterios establecidos por el usuario.
2. Obtención del contenido de *URLs* , pre-procesamiento del contenido extraído de *URLs* en diferentes formatos y la creación de documentos en un formato estandarizado para la aplicación.
3. Etiquetado de contenidos mediante la identificación de palabras y frases en base a modelos de vocabularios, pre-procesamiento de contenidos mediante identificación de sinonimias, acrónimos, etc…
4. Persistencia de contenidos en formato original así como en formato estructurado para su gestión en una base de datos relacional.
5. Análisis de contenidos para la extracción de información y conocimiento, en base al análisis conjunto de documentos y de datos de mercado.

Para desarrollar las funcionalidades anteriores se plantea la composición del sistema en los cuatro subsistemas siguientes:

1. Subsistema para la búsqueda y extracción de información de Internet: este componente de la aplicación será el responsable de los algoritmos para la búsqueda de *URLs* en base a los criterios establecidos por el usuario, así como para la obtención de contenidos de los *URLs* de interés.
2. Subsistema de pre-procesamiento y gestión de contenidos: este componente será el encargado de las tareas de filtrado de contenidos repetitivos, limpieza de contenidos normalmente marcados con formatos HTML o XML, etiquetado de contenidos mediante la identificación de palabras y frases, identificación de sinonimias, etc…

3. Subsistema para la obtención y gestión de datos de mercado, a partir de proveedores de datos en *streaming* o datos históricos.
4. Subsistema de análisis de documentos: este componente será el responsable de la ejecución de los algoritmos para el análisis de la información y la extracción de conocimiento. Fundamentalmente, deberá permitir la ejecución de algoritmos de minería de datos y aprendizaje computacional, sobre los datos.

Tareas a realizar, entregables y fechas de entrega

Tarea a realizar		Inicio	Entrega	Entregables
T1	PEC1: Planificación y definición de la propuesta	16/09/2013	02/10/2013	—
T2	Definición del proyecto	16/09/2012	22/09/2013	—
T3	Búsqueda de documentación	23/09/2012	26/09/2013	—
T4	Planteamiento de objetivos	27/09/2012	28/09/2013	—
T5	Desarrollo del Plan de Trabajo	29/09/2012	30/09/2013	—
T6	Redacción del documento de la PEC1	27/09/2012	02/10/2013	—
T7	Entrega PEC1	02/10/2013	02/10/2013	Documento PEC1
T8	PEC2: Análisis funcional	03/10/2013	31/10/2013	—
T9	Análisis de requisitos	03/10/2013	08/10/2013	—
T10	Desarrollo de diagramas de casos de uso	09/10/2013	09/10/2013	—
T11	Análisis del modelo estático	10/10/2013	14/10/2013	—
T12	Análisis del modelo lógico de datos	12/10/2013	16/10/2013	—
T13	Análisis del modelo dinámico	14/10/2013	19/10/2013	—
T14	Desarrollo de diagramas de clases	20/10/2013	25/10/2013	—
T15	Redacción del documento de la PEC2	26/10/2013	31/10/2013	—
T16	Entrega PEC2	31/10/2013	31/10/2013	Documento PEC2
T17	PEC3: Diseño	01/11/2013	29/11/2013	—
T18	Diseño de componentes	01/11/2013	06/11/2013	—
T19	Diseño de la arquitectura del sistema	03/11/2013	12/11/2013	—
T20	Diseño de la interfaz de usuario	13/11/2013	20/11/2013	—
T21	Desarrollo de diagramas de diseño	21/11/2013	24/11/2013	—
T22	Redacción del documento de la PEC3	25/11/2013	29/11/2013	—
T23	Entrega PEC3	29/11/2013	29/11/2013	Documento PEC3

T24	PEC4: Memoria, producto, presentación	30/11/2013	08/01/2014	—
T25	*Implementación del diseño*	30/11/2013	03/01/2014	—
T26	*Desarrollo de pruebas*	09/12/2013	03/01/2014	—
T27	*Redacción de la memoria*	15/12/2013	08/01/2014	—
T28	*Desarrollo de la presentación*	04/01/2014	08/01/2014	—
T29	Entrega PEC4	08/01/2014	08/01/2014	- Memoria final - Presentación - Aplicación
T30	*Debate y Defensa del TFG*	13/01/2014	27/01/2014	—
T31	*Defensa*	13/01/2013	27/01/2014	Respuestas a posibles preguntas

Diagrama de Gantt

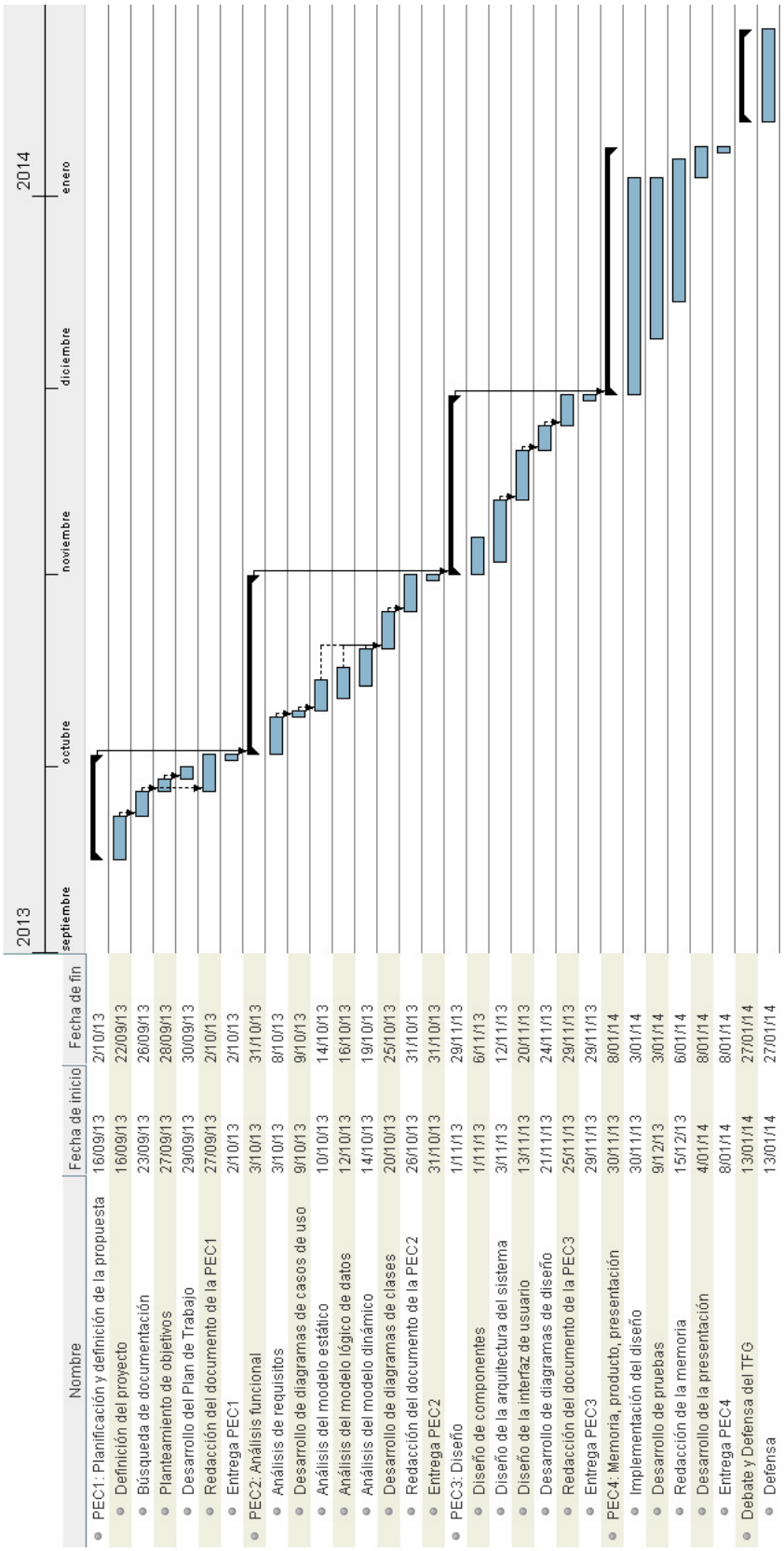


Figura 1. Diagrama de Gantt con la planificación temporal del proyecto.

1.5 Breve sumario de productos obtenidos

Los productos obtenidos durante el desarrollo del proyecto son:

1. Memoria: contiene los antecedentes, plan de proyecto y ejecución del plan de trabajo.
2. Prototipo del sistema: es una implementación inicial de la aplicación propuesta para su evaluación.
3. Presentación: documento de video con diapositivas y capturas de pantalla para la defensa del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria se desarrolla en tres capítulos principales donde se describe la ejecución del plan de trabajo:

1. Análisis: durante la fase de análisis de desarrolla el modelo de dominio y los casos de uso y requisitos del sistema.
2. Diseño: en la fase de diseño se define el diseño de la arquitectura del sistema, el diseño de componentes, la interfaz de usuario y el modelo de datos.
3. Producto: se describen las decisiones tomadas durante el proceso de implementación del prototipo del sistema.

2. Ejecución del plan de trabajo — Análisis

2.1. Análisis de las funcionalidades del sistema

El sistema a desarrollar tiene como objetivo la obtención y gestión de contenidos (principalmente noticias) en Internet para su posterior análisis y extracción de conocimiento. Inicialmente la aplicación se plantea como el prototipo de un sistema en tres capas (presentación, capa de negocio y persistencia de datos). Se pretende realizar un diseño flexible que permita adaptar el sistema a diferentes tecnologías de implementación; tanto en la forma de una aplicación local como una aplicación en forma de objetos distribuidos, aplicación basada en servicios Web, etc… De forma genérica se atribuyen cuatro funcionalidades principales al sistema:

1. Obtención de contenidos de Internet. La obtención de contenidos utiliza dos estrategias, la suscripción utilizando los protocolos RSS y Atom, y el rastreo de enlaces en Internet utilizando estrategias de *crawling*.
2. Gestión de documentos.
3. Datos de mercado.
4. Análisis de documentos y extracción de conocimiento.

El diagrama siguiente presenta estas funcionalidades de alto nivel en estructura de paquetes así como sus interdependencias:

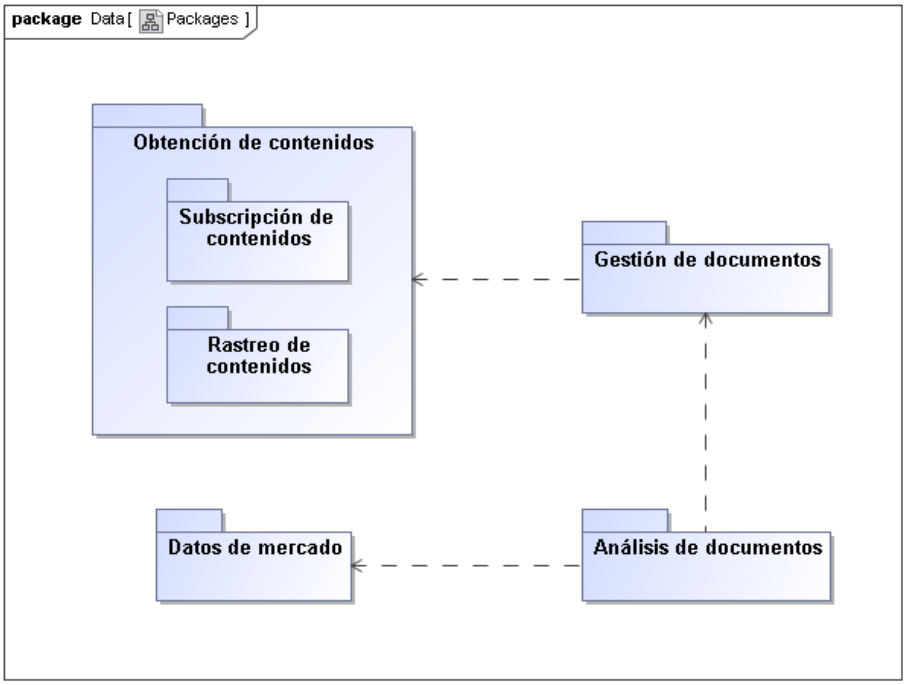


Figura 2. Funcionalidades de alto nivel del sistema estructuradas en paquetes y sus dependencias.

Obtención de contenidos de Internet

Las funcionalidades asociadas a la obtención de contenidos permitirán al usuario automatizar el proceso de obtención de información en Internet principalmente en forma de noticias. El sistema permitirá utilizar dos estrategias. Por un lado el usuario podrá subscribirse a proveedores de contenidos en formato *RSS* y *Atoms*. La segunda estrategia permitirá al usuario desarrollar estrategias de rastreado automático de sitios Web utilizando algoritmos de *crawling*.

- ***Subscripción de contenidos RSS y Atom***

El usuario podrá crear y gestionar sus propias subscripciones a fuentes de información en formato *RSS* y *Atom*. Utilizando las subscripciones creadas podrá programar la lectura de entradas de forma automática, utilizando un lector de subscripciones. El sistema recibirá como *imput* las entradas provenientes de los proveedores, y tendrá que llevar a cabo el análisis de las entradas, la extracción del enlace (*URL)* asociado, y por último acceder al documento Web y extraer su contenido de interés.

- ***Rastreado de contenidos***

El usuario podrá utilizar estrategias de rastreado de enlaces (crawling) de Internet para encontrar *URLs* a contenidos Web de interés. Las estrategias estarán ejecutadas por un rastreador y tendrán una serie de parámetros que podrá configurar el usuario. Para ello el usuario podrá crear y gestionar sus propias estrategias de rastreo. Una estrategia de rastreo partirá de una semilla de rastreo que consiste básicamente en una colección de *URLs* sobre las que se iniciará el rastreo de nuevos contenidos. Esta parte del sistema se encarga de la búsqueda de contenidos en Internet mediante el uso de técnicas de rastreo (*crawling*).

Gestión de contenidos:

Las estrategias de obtención de contenidos descritas en el apartado anterior, resultan en una colección de *URLs* de contenidos en Internet y sus documentos originales asociados. El usuario podrá llevar a cabo la construcción de documentos en un formato estándar en base a esos contenidos para su tratamiento y gestión. El sistema incorpora las siguientes funcionalidades para la gestión de contenidos.

- ***Construcción de documentos***

El usuario podrá crear documentos en base a contenidos Web asociados a una *URL*. Mientras que los contenidos en Internet corresponderán a información en diferentes formatos, la creación de documentos implica la transformación del contenido a un formato estándar basado en campos para que sean tratables mediante la lógica de datos de la aplicación. La construcción de documentos implica analizar los contenidos obtenidos en sitios Web y la extracción de los datos basada en campos: *autor*, *título*, *cuerpo*, *url*, etc… El usuario podrá consultar y gestionar los documentos creados, mediante su exploración, visualización, borrado, revisión, etc…

- ***Indexado de documentos***

El indexado es el proceso por el cual un documento es dividido en una colección de palabras, las cuales son analizadas para su normalización y eliminación de términos poco informativos. El usuario podrá crear y gestionar índices para la organización de documentos. El indexado es la herramienta principal para poder buscar y extraer información de documentos.

- ***Búsqueda de documentos:***

El usuario podrá llevar a cabo búsquedas de documentos utilizando una solicitud de búsqueda sobre un índice de documentos. Para ello el usuario podrá crear y gestionar sus propias solicitudes y resultados de búsqueda.

Obtención de datos de mercado

El sistema tendrá inicialmente, únicamente acceso a datos del mercado continuo español. Para ello se desarrollarán dos funcionalidades. Una permitirá al usuario obtener datos históricos de cotizaciones desde el servidor de datos históricos de Yahoo.com finanzas. Con la segunda funcionalidad el usuario podrá conectarse mediante *streaming* al servidor de datos de mercado de cotizaciones.me, que es una fuente gratuita de datos en *streaming*.

Análisis de documentos:

La funcionalidad de análisis de documentos se fundamenta en la posibilidad de construir una representación vectorizada de los documentos de manera que estos puedan ser tratados cuantitativamente por algoritmos de clasificación o agrupamiento. La construcción de una representación vectorizada implica la división del documento en términos (*tokens*), los cuales pueden tener la forma de palabras simples, o bien frases más o menos complejas.

- ***Identificación de documentos repetidos:***

El usuario podrá identificar documentos repetidos, es decir aquellos documentos que a pesar de proceder de *URLs* diferentes poseen contenidos iguales o muy parecidos por tener una procedencia común.

- ***Clasificación de documentos:***

El usuario podrá crear y gestiona estrategias de clasificación de documentos en base a criterios de clasificación personalizados. Utilizando estas estrategias y la representación vectorizada de documentos el usuario podrá clasificar documentos mediante la utilización de diferentes algoritmos de clasificación.

- ***Agrupamiento de documentos:***

Utilizando diferentes algoritmos de agrupación (*clustering*) y la representación vectorizada de documentos el usuario podrá llevar a cabo la agrupación de colecciones de documentos.

2.2. Ejemplos de escenarios de uso del sistema

Con el fin de ilustrar el ámbito de utilización de la aplicación a continuación describo de forma textual dos escenarios de uso que pueden ser comunes al sistema.

1. **Obtención selectiva de contenidos**: Internet ofrece de forma continua la publicación masiva de información cuya búsqueda y selección por parte del usuario consume una gran cantidad de tiempo si se lleva a cabo de forma manual. Suponemos el caso de un usuario Carlos, que antes del comienzo de una sesión de mercado lleva a cabo una lectura lo más exhaustiva posible de las noticias que se han publicado acerca de los valores incluidos en Ibex35 del mercado continuo español. Para ello Carlos implementa una estrategia de suscripción y rastreo de contenidos que además selecciona aquellos que son de interés para el en base a criterios de categorización con otros contenidos que ha analizado previamente de forma histórica. Carlos ejecuta la estrategia de obtención selectiva de contenidos desde el cierre del mercado (17:35), durante toda la noche y hasta una hora antes de la apertura de la sesión siguiente (8:00). Lo que inicialmente podría haber constituido la selección de sitios de interés sobre miles de contenidos, la herramienta le ofrece a Carlos una selección de 50 noticias seleccionadas. Carlos lleva a cavo su tarea de información antes de la apertura del mercado de una manera eficiente.

2. **Evaluación automática de contenidos**: en un segundo escenario Carlos quiere tener constancia de noticias que se van liberando en tiempo real sobre valores incluidos en Ibex35 del mercado continuo español, con el fin de poder saber si la evolución de la opinión de expertos es positiva o negativa. Sin embargo, durante la evolución de la sesión Carlos no puede hacer frente a la lectura de por ejemplo 10 nuevas noticias por minuto. Para ello Carlos desarrolla una estrategia que le permita la obtención de contenidos de interés y además utiliza algoritmos de clasificación de los contenidos que le permitan evaluar lo positivo o negativo de las noticias. Un tratamiento estadístico de las mismas le da a Carlos un pulso de la opinión de los expertos sin necesidad de llevar a cabo la lectura manual de un solo documento.

2.3. Descripción de actores

Usuario: actor que utiliza las funcionalidades básicas de la aplicación, fundamentalmente para la obtención, gestión y análisis de noticias.

Proveedor de contenidos: servidor remoto que inyecta contenidos en formato *RSS* o *Atom*, activando alguna de las funcionalidades del sistema.

Proveedor de datos de mercado: servidor remoto que inyecta datos de mercado en forma de *streaming*, activando alguna de las funcionalidades del sistema.

2.4. Análisis de casos de uso y documentación de requisitos

Se presentan diagramas para los casos de uso principales del sistema. Los requisitos se documentan en forma de historias de usuario utilizando una sintaxis compatible con la API JBehave para que pueda ser utilizados mediante desarrollo guiado por pruebas.

Obtención de contenidos

- ***Subscripción de contenidos RSS y Atom:***

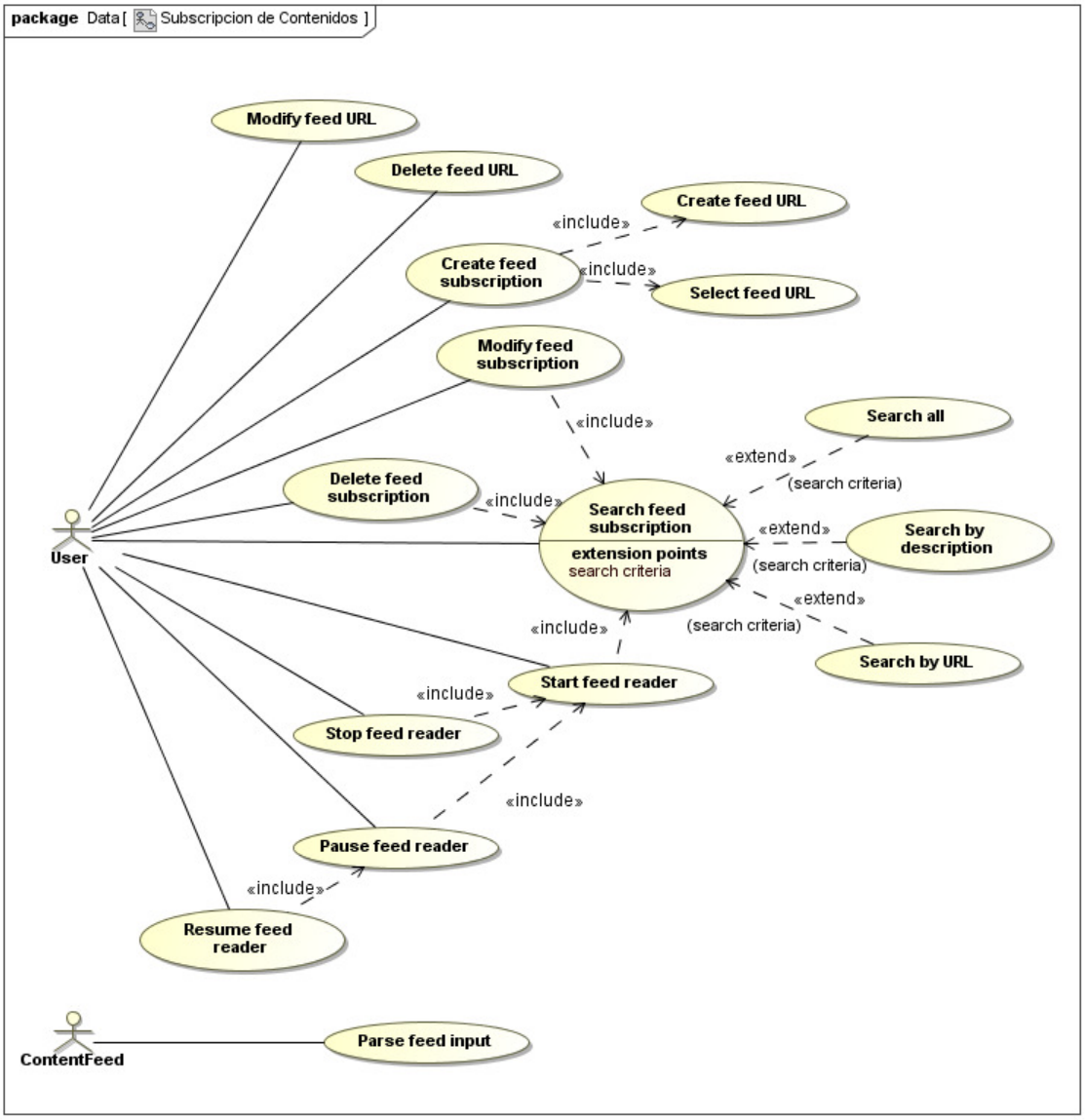


Figura 3. Casos de uso asociados a la suscripción de contenidos RSS y Atom.

Caso de uso:	**Actor:**
Crear nueva subscripción (New subscription)	Usuario

```
User story for the use case "New subscription"

Narrative:
```

```
In order to retrieve content from syndication feeds
As a user of the application
I want to create a new subscription

Scenario: The user creates a new subscription

Given the following subscriptions in the data base
Examples:

When the user creates the subscription [subscription]
Examples:

Then the outcome should be [message]

Examples:
|Subscription|message|
|Subscription1|Subscription created correctly|
|Subscription2|The subscription ID already exist|
|Subscription3|Subscription created correctly|
```

Caso de uso: Modificar una subscripción (Modify subscription)	**Actor:** Usuario

```
User story for the use case "Modify subscription"

Narrative:
In order to maintain updated the information of syndication feeds
As a user of the application
I want to modify a subscription stored in the database

Scenario: The user modifies a subscription

Given the following subscriptions in the database
Examples:

When the user modifies the subscription [subscription]
Examples:

Then the outcome should [message]

Examples:
|Subscription|message|
|Subscription1|Subscription modified correctly|
|Subscription2|The subscription ID does not exit|
```

Caso de uso: Eliminar una subscripción (Delete subscription)	**Actor:** Usuario

```
User story for the use case "Delete subscription"

Narrative:
In order to maintain updated the syndication feed providers
As a user of the application
I want to delete a subscription stored in the database

Scenario: The user deletes a subscription

Given the following subscriptions in the database
Examples:

When the user deletes the subscription [subscription]
Examples:
```

```
Then the outcome should [message]

Examples:
|Subscription|message|
|Subscription1|Subscription deleted correctly|
|Subscription2|The subscription ID does not exit|
```

Caso de uso: Buscar subscripción (Search subscription)	**Actor:** Usuario

```
User story for the use case "Search subscription"

Narrative:
In order to retrieve a syndication feed provider
As a user of the application
I want to search a subscription stored in the database

Scenario: The user searches a subscription

Given the following subscriptions in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the subscription

Given the following subscriptions in the database
Examples:

When the user searches all the subscriptions

Then the outcome should [message]

Examples:
```

Caso de uso: Iniciar lector de subscripción (Start reader)	**Actor:** Usuario

```
User story for the use case "Start reader"

Narrative:
In order to retrieve content from syndication feeds
As a user of the application
I want to run a subscription reader

Scenario: The user starts a subscription reader

Given a reader with subscription [subscription]
And the reader is in status [status]
Examples:
|Subscription|Status|

When the user starts the subscription reader
```

```
Then the outcome should be [message]

Examples:
|Subscription|status|message|
|Subscription1|status1|You have not selected a subscription|
|Subscription2|status2|Reader is already running|
|Subscription3|status3|Reader started|
```

Caso de uso:	**Actor:**
Parar lector (Stop reader)	Usuario

```
User story for the use case "Stop reader"

Narrative:
In order to end receiving content from syndication feeds
As a user of the application
I want to stop a subscription reader

Scenario: The user stops a subscription reader

Given a reader with subscription [subscription]
And the reader is in status [status]
Examples:
|Subscription|status|

When the user stopts the subscription reader

Then the outcome should be [message]

Examples:
|subscription|status|message|
|subscription1|status1|The reader is not running|
|subscription2|status2|The reader has been stopped successfully|
```

Caso de uso:	**Actor:**
Pausar lector (Pause reader)	Usuario

```
User story for the use case "Pause reader"

Narrative:
In order to temporarily stop receiving content from syndication feeds
As a user of the application
I want to pause a subscription reader

Scenario: The user pause a subscription reader

Given a reader with subscription [subscription]
And the reader is in status [status]
Examples:
|subscription|status|

When the user pauses the subscription reader

Then the outcome should be [message]

Examples:
|subscription|status|message|
|subscription1|status1|The reader is not running|
|subscription2|status2|The reader is already paused|
|subscription3|status3|The reader has been paused successfully|
```

Caso de uso:	**Actor:**
Reanudar lector (Resume reader)	Usuario

```
User story for the use case "Resume reader"

Narrative:
In order to restart a temporarily stopped syndication feed
As a user of the application
I want to resume a subscription reader

Scenario: The user resume a subscription reader

Given a reader with subscription [subscription]
And the reader is in status [status]
Examples:
|subscription|status|

When the user resumes the subscription reader

Then the outcome should be [message]

Examples:
|subscription|status|message|
|subscription1|status1|The reader is not running|
|subscription2|status2|The reader is already running|
|subscription3|status3|The reader has been resumed successfully|
```

Caso de uso: Analizar la entrada de una subscripción (Parse feed input)	**Actor:** Proveedor de contenidos

```
User story for the use case "Parse feed input"

Narrative:
In order to provide content to the system
As a syndication content provider of the application
I want the system parses input feeds

Scenario: The syndication content provider submits an input content

Given an active syndication feed reader

When the a content provider submits a content [content_file] in
format [format]
Examples:
|content_file|format|

Then the outcome should be

Examples:
```

- ***Rastreado de contenidos:***

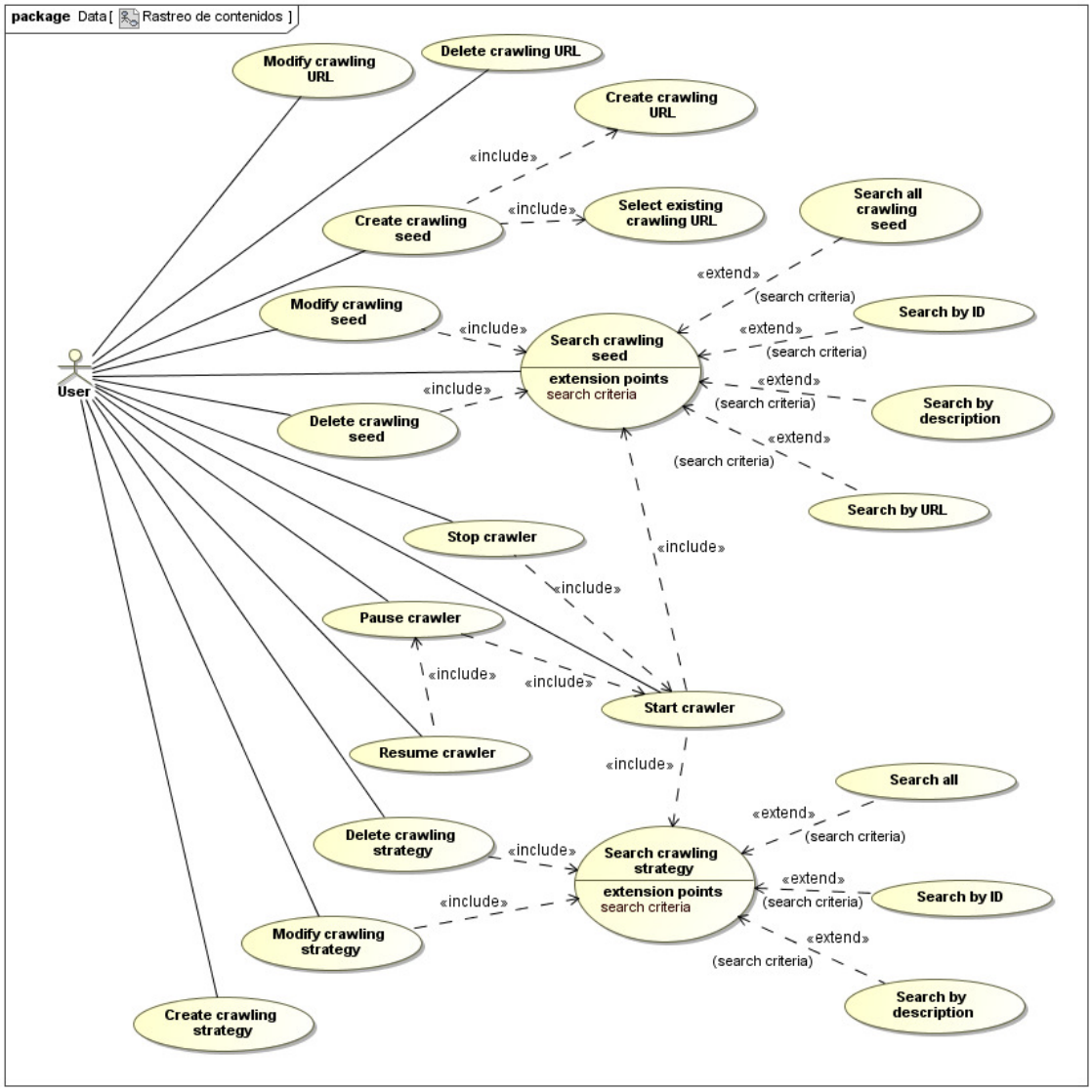


Figura 4. Casos de uso asociados al rastreado de contenidos en Internet (crawling).

Caso de uso:	**Actor:**
Crear nueva semilla de rastreo (New crawling seed)	Usuario

```
User story for the use case "New crawling seed"

Narrative:
In order to crawl content in Internet
As a user of the application
I want to create a new crawling seed

Scenario: The user creates a new crawling seed

Given the following crawling seeds in the database
Examples:

When the user creates a crawling seed [seed]
Examples:

Then the outcome should be [message]
```

```
Examples:
|seed|message|
|seed1|Crawling seed created correctly|
|seed2|The crawling seed ID already exist|
|seed3|Crawling seed created correctly|
```

Caso de uso:	**Actor:**
Modificar una semilla de rastreo (Modify crawling seed)	Usuario

```
User story for the use case "Modify crawling seed"

Narrative:
In order to maintain updated the information of crawling seeds
As a user of the application
I want to modify a crawling seed stored in the database

Scenario: The user modifies a crawling seed

Given the following crawling seeds in the database
Examples:

When the user modifies the crawling seed [seed]
Examples:

Then the outcome should [message]

Examples:
|seed|message|
|seed1|Crawling seed modified correctly|
|seed2|The crawling seed ID does not exit|
```

Caso de uso:	**Actor:**
Eliminar una semilla de rastreo (Delete crawling seed)	Usuario

```
User story for the use case "Delete crawling seed"

Narrative:
In order to maintain updated the crawling seeds
As a user of the application
I want to delete a crawling seed stored in the database

Scenario: The user deletes a crawling seed

Given the following crawling seeds in the database
Examples:

When the user deletes the crawling seed [seed]
Examples:

Then the outcome should [message]

Examples:
|seed|message|
|seed1|Crawling seed deleted correctly|
|seed2|The crawling seed ID does not exit|
```

Caso de uso:	**Actor:**
Buscar semilla de rastreo (Search crawling seed)	Usuario

```
User story for the use case "Search crawling seed"

Narrative:
In order to retrieve a crawling seed
As a user of the application
```

```
I want to search a crawling seed stored in the database

Scenario: The user searches a crawling seed

Given the following crawling seeds in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the crawling seeds

Given the following crawling seeds in the database
Examples:

When the user searches all the crawling seeds

Then the outcome should be

Examples:
```

Caso de uso: Crear nueva estrategia de rastreo (New crawling strategy)	**Actor:** Usuario

```
User story for the use case "New crawling strategy"

Narrative:
In order to crawl content in Internet
As a user of the application
I want to create a new crawling strategy

Scenario: The user creates a new crawling strategy

Given the following crawling strategies in the data base
Examples:

When the user creates a crawling strategy [strategy]
Examples:

Then the outcome should be [message]

Examples:
|strategy|message|
|strategy1|Crawling strategy created correctly|
|strategy2|The crawling strategy ID already exist|
|strategy3|Crawling strategy created correctly|
```

Caso de uso: Modificar una estrategia (Modify crawling strategy)	**Actor:** Usuario

```
User story for the use case "Modify crawling strategy"

Narrative:
In order to maintain updated the information of crawling strategies
As a user of the application
```

```
I want to modify a crawling strategy stored in the database

Scenario: The user modifies a crawling strategy

Given the following crawling strategies in the database
Examples:

When the user modifies the crawling strategy [strategy]
Examples:

Then the outcome should [message]

Examples:
|strategy|message|
|strategy1|Crawling strategy modified correctly|
|strategy2|The crawling strategy ID does not exit|
```

Caso de uso: Eliminar una estrategia (Delete crawling strategy)	**Actor:** Usuario

```
User story for the use case "Delete crawling strategy"

Narrative:
In order to maintain updated the crawling strategies
As a user of the application
I want to delete a crawling strategy stored in the database

Scenario: The user deletes a crawling strategy

Given the following crawling strategies in the database
Examples:

When the user deletes the crawling strategy [strategy]
Examples:

Then the outcome should [message]

Examples:
|strategy|message|
|strategy1|Crawling strategy deleted correctly|
|strategy2|The crawling strategy ID does not exit|
```

Caso de uso: Buscar estrategia de rastreo (Search crawling strategy)	**Actor:** Usuario

```
User story for the use case "Search crawling strategy"

Narrative:
In order to retrieve a crawling strategy
As a user of the application
I want to search a crawling strategy stored in the database

Scenario: The user searches a crawling strategy

Given the following crawling strategies in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
```

```
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the crawling strategies

Given the following crawling strategies in the database
Examples:

When the user searches all the crawling strategies

Then the outcome should be

Examples:
```

Caso de uso: Iniciar rastreador (Start crawler)	**Actor:** Usuario

```
User story for the use case "Start crawler"

Narrative:
In order to crawl content from in the internet
As a user of the application
I want to run a web crawler

Scenario: The user starts a web crawler

Given a crawler with seed [seed] and strategy [strategy]
And the crawler is in status [status]
Examples:

When the user starts the subscription crawler

Then the outcome should be [message]

Examples:
|seed|strategy|status|message|
|seed1|strategy1|status1|You have not selected a seed|
|seed2|strategy2|status2|You have not selected a strategy|
|seed3|strategy3|status3|Crawler is already running|
|seed4|strategy4|status4|Crawler started successfully |
```

Caso de uso: Parar rastreador (Stop crawler)	**Actor:** Usuario

```
User story for the use case "Stop crawler"

Narrative:
In order to end using a web crawler
As a user of the application
I want to stop a crawler

Scenario: The user stops a crawler

Given a crawler with seed [seed] and strategy [strategy]
And the crawler is in status [status]
Examples:

When the user stops the crawler

Then the outcome should be [message]

Examples:
```

```
|seed|strategy|status|message|
|seed1|strategy1|status1|The crawler is not running|
|seed2|strategy2|status2|The crawler stopped successfully|
```

Caso de uso: Pausar rastreador (Pause crawler)	**Actor:** Usuario

```
User story for the use case "Pause crawler"

Narrative:
In order to temporarily stop using a web crawler
As a user of the application
I want to pause a crawler

Scenario: The user pause a crawler

Given a crawler with seed [seed] and strategy [strategy]
And the crawler is in status [status]
Examples:

When the user pauses the crawler

Then the outcome should be [message]

Examples:
|seed|strategy|status|message|
|seed1|strategy1|status1|The crawler is not running|
|seed2|strategy2|status2|The crawler is already paused|
|seed3|strategy3|status3|The crawler paused successfully|
```

Caso de uso: Reanudar rastreador (Resume crawler)	**Actor:** Usuario

```
User story for the use case "Resume crawler"

Narrative:
In order to restart a temporarily stopped crawler
As a user of the application
I want to resume a crawler

Scenario: The user resume a crawler

Given a crawler with seed [seed] and strategy [strategy]
And the crawler is in status [status]
Examples:

When the user resumes the crawler

Then the outcome should be [message]

Examples:
|seed|strategy|status|message|
|seed1|strategy1|status1|The crawler is not running|
|seed2|strategy2|status2|The crawler is already running|
|seed3|strategy3|status3|The crawler resumed successfully|
```

Gestión de documentos

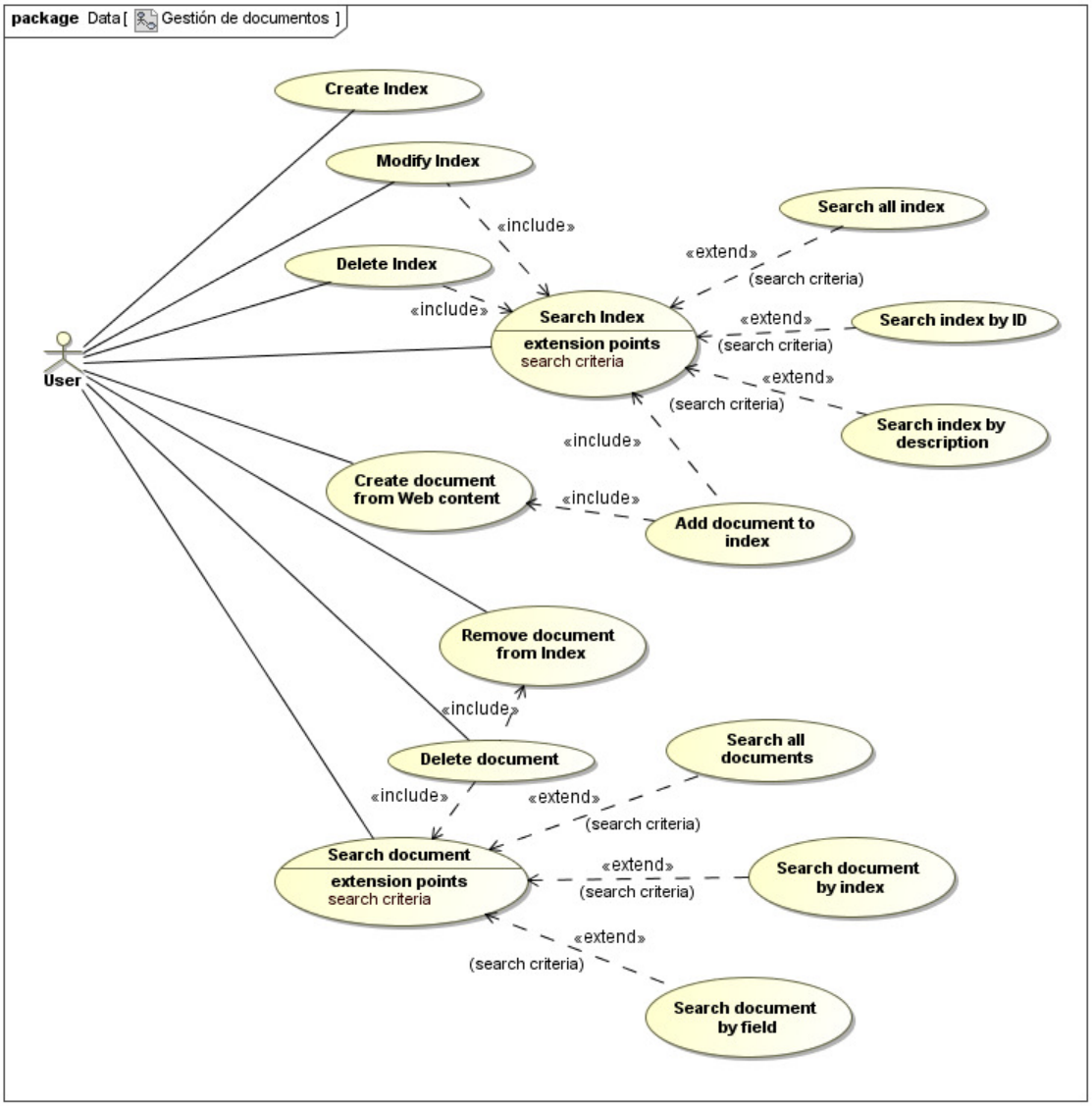


Figura 5. Casos de uso asociados a la gestión de documentos.

Caso de uso: Crear nuevo índice (New index)	**Actor:** Usuario

```
User story for the use case "New index"

Narrative:
In order to manage documents in the application
As a user of the application
I want to create a new index

Scenario: The user creates a new index

Given the following indexes in the database
Examples:

When the user creates a index [index]
Examples:

Then the outcome should be [message]
```

```
Examples:
|index|message|
|index1|Index created correctly|
|index2|The index ID already exist|
|index3|Index created correctly|
```

Caso de uso: Modificar un índice (Modify index)	**Actor:** Usuario

```
User story for the use case "Modify index"

Narrative:
In order to maintain updated indexes
As a user of the application
I want to modify a index in the database

Scenario: The user modifies a index

Given the following indexes in the database
Examples:

When the user modifies the index [index]
Examples:

Then the outcome should [message]

Examples:
|index|message|
|index1|Index modified correctly|
|index2|The index ID does not exit|
```

Caso de uso: Eliminar un índice (Delete index)	**Actor:** Usuario

```
User story for the use case "Delete index"

Narrative:
In order to maintain updated indexes
As a user of the application
I want to delete a index stored in the database

Scenario: The user deletes a index

Given the following indexes in the database
Examples:

When the user deletes the index [index]
Examples:

Then the outcome should [message]

Examples:
|index|message|
|index1|Index deleted correctly|
|index2|The index ID does not exit|
```

Caso de uso: Buscar índice (Search index)	**Actor:** Usuario

```
User story for the use case "Search index"

Narrative:
In order to retrieve an index
```

```
As a user of the application
I want to search an index stored in the database

Scenario: The user searches an index

Given the following indexes in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the subscription

Given the following indexes in the database
Examples:

When the user searches all the indexes

Then the outcome should [message]

Examples:
```

Caso de uso: Añadir documento a índice (Add document to index)	**Actor:** Usuario

```
User story for the use case "Add document to index"

Narrative:
In order to store documents
As a user of the application
I want to create a new document

Scenario: The user adds a new document to an index

Given the index [index]
Examples:

When the user adds the document [document]
Examples:

Then the outcome should be [message]

Examples:
|index|document|message|
|index1|document1|Document created correctly|
|index2|document2|The document already exist|
```

Caso de uso: Eliminar documento de un índice (Remove document from index)	**Actor:** Usuario

```
User story for the use case "Remove document from index"

Narrative:
In order to maintain updated an index of documents
As a user of the application
```

```
I want to remove a document from an index

Scenario: The user remove a document from an index

Given the index [index]
Examples:

When the user removes the document [document]
Examples:

Then the outcome should [message]

Examples:
|index|document|message|
|index1|document1|Document deleted correctly|
|index2|document2|The document does not exit|
```

Caso de uso: Buscar documento (Search document)	**Actor:** Usuario

```
User story for the use case "Search document"

Narrative:
In order to retrieve documents
As a user of the application
I want to search a document

Scenario: The user searches a document

Given the following documents in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the documents

Given the following index [index]
Examples:

When the user searches all the documents

Then the outcome should [message]

Examples:
```

Obtención de datos de mercado

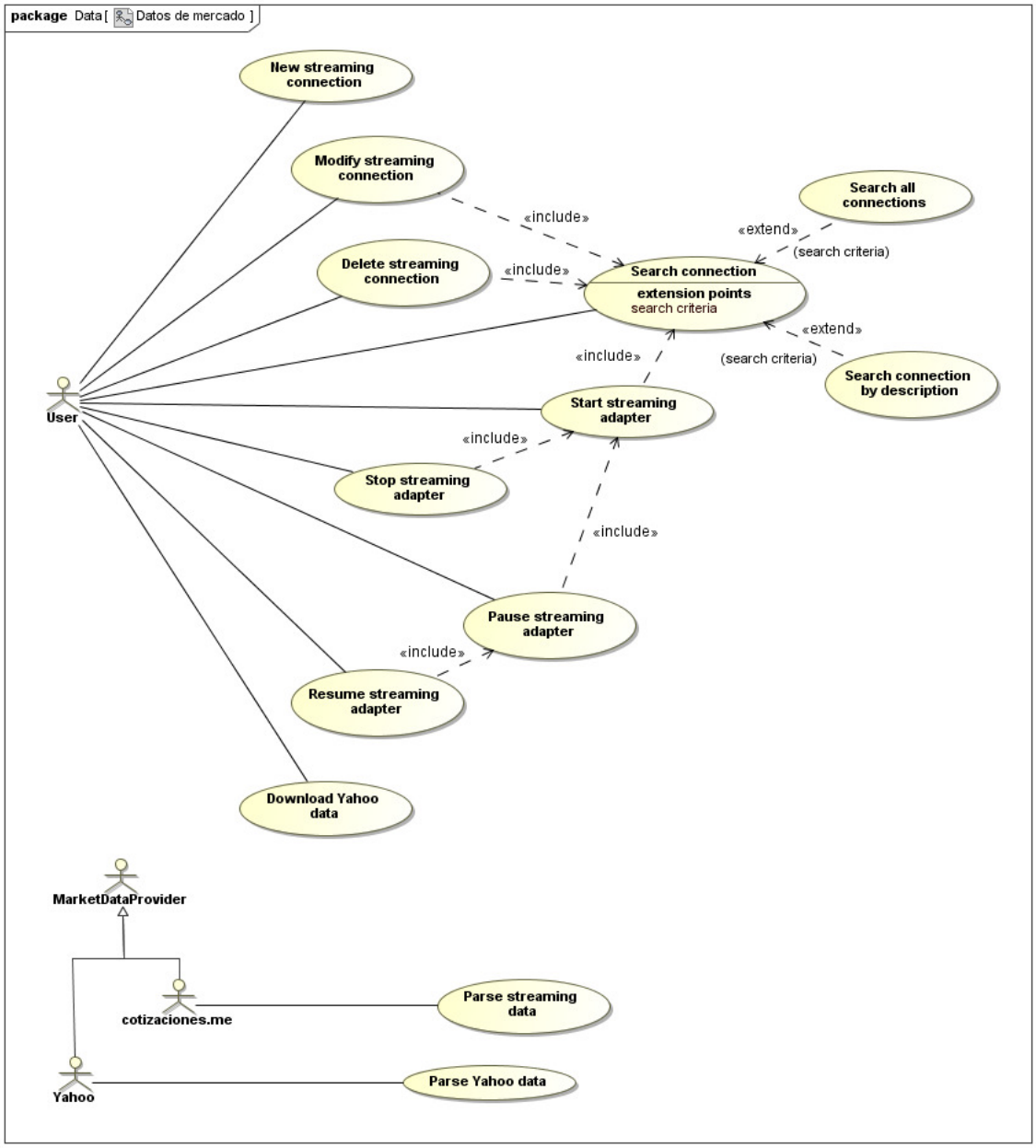


Figura 6. Casos de uso asociados a la obtención de datos de mercado.

Caso de uso: Nueva conexión streaming (New streaming connection)	**Actor:** Usuario

```
User story for the use case "New streaming connection"

Narrative:
In order to retrieve market data from a streaming server
As a user of the application
I want to create a new connection

Scenario: The user creates a new connection

Given the following connections in the database
Examples:
```

```
When the user creates a connection [connection]
Examples:

Then the outcome should be [message]

Examples:
|connection|message|
|connection1|Connection created correctly|
|connection2|The connection already exist|
|connection3|Connection created correctly|
```

Caso de uso:	**Actor:**
Modificar una conexión streaming (Modify streaming connection)	Usuario

```
User story for the use case "Modify streaming connection"

Narrative:
In order to maintain updated the information of streaming connections
As a user of the application
I want to modify a connection stored in the database

Scenario: The user modifies a connection

Given the following connections in the database
Examples:

When the user modifies the connection [connection]
Examples:

Then the outcome should [message]

Examples:
|connection|message|
|connection1|Connection modified correctly|
|connection2|The connection does not exit|
```

Caso de uso:	**Actor:**
Eliminar una conexión streaming (Delete streaming connection)	Usuario

```
User story for the use case "Delete streaming connection"

Narrative:
In order to maintain updated the streaming connections
As a user of the application
I want to delete a connection stored in the database

Scenario: The user deletes a connection

Given the following connections in the database
Examples:

When the user deletes the connection [connection]
Examples:

Then the outcome should [message]

Examples:
|connection|message|
|connection1|Connection deleted correctly|
|connection2|The connection does not exit|
```

Caso de uso:	**Actor:**
Buscar conexión streaming (Search streaming connection)	Usuario

```
User story for the use case "Search streaming connection"

Narrative:
In order to retrieve a connection from the database
As a user of the application
I want to search a connection stored in the database

Scenario: The user searches a connection

Given the following connections in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the connection

Given the following connections in the database
Examples:

When the user searches all the connections

Then the outcome should be

Examples:
```

Caso de uso: Iniciar adaptador streaming (Start streaming adapter)	**Actor:** Usuario

```
User story for the use case "Start streaming adapter"

Narrative:
In order to retrieve market data from a streaming server
As a user of the application
I want to run a streaming adapter

Scenario: The user starts a streaming adapter

Given a streaming adapter [adapter] with subscription [subscription]
And the streaming adapter is in status [status]
Examples:

When the user starts the streaming adapter

Then the outcome should be [message]

Examples:
|adapter|subscription|status|message|
|adapter1|subscription1|status1|You have not selected a streaming
adapter|
|adapter1|subscription1|status2|The streaming adapter is already
running|
|adapter1|subscription1|status3|The streaming adapter started
correctly|
```

Caso de uso: Parar adaptador streaming (Stop streaming adapter)	**Actor:** Usuario

```
User story for the use case "Stop streaming adapter"

Narrative:
In order to end receiving market data from a streaming server
As a user of the application
I want to stop a streaming adapter

Scenario: The user stops a streaming adapter

Given a streaming adapter with connection [connection]
And the streaming adapter is in status [status]
Examples:

When the user stops the streaming adapter

Then the outcome should be [message]

Examples:
|connection|status|message|
|connection1|status1|The streaming adapter is not running|
|connection1|status2|The adapter has been stopped successfully|
```

Caso de uso: Pausar el adaptador streaming (Pause streaming adapter)	**Actor:** Usuario

```
User story for the use case "Pause streaming adapter"

Narrative:
In order to temporarily stop receiving market data
As a user of the application
I want to pause a streaming adapter

Scenario: The user pause a streaming adapter

Given a streaming adapter with connection [connection]
And the reader is in status [status]
Examples:

When the user pauses the streaming adapter

Then the outcome should be [message]

Examples:
|connection|status|message|
|connection1|status1|The streaming adapter is not running|
|connection1|status2|The streaming adapter is already paused|
|connection1|status3|The adapter has been paused successfully|
```

Caso de uso: Reanudar adaptador streamer (Resume streaming adapter)	**Actor:** Usuario

```
User story for the use case "Resume streaming adapter"

Narrative:
In order to restart a temporarily stopped streaming adapter
As a user of the application
I want to resume a streaming adapter

Scenario: The user resume a streaming adapter
```

```
Given a streaming adapter with connection [connection]
And the streaming adapter is in status [status]
Examples:

When the user resumes the streaming adapter

Then the outcome should be [message]

Examples:
|connection|status|message|
|connection1|status1|The streaming adapter is not running|
|connection1|status2|The streaming adapter is already running|
|connection1|status3|The streaming adapter has been resumed successfully|
```

Caso de uso: Descargar datos de mercado de Yahoo (Download Yahoo data)	**Actor:** Usuario

```
User story for the use case "Download Yahoo data"

Narrative:
In order to retrieve historical market data
As a user of the application
I want to download data from Yahoo

Scenario: The user downloads data from Yahoo server

Given the user makes a download request [request]
Examples:

When the user proceeds to download data

Then the outcome should be

Examples:
```

Análisis de documentos:

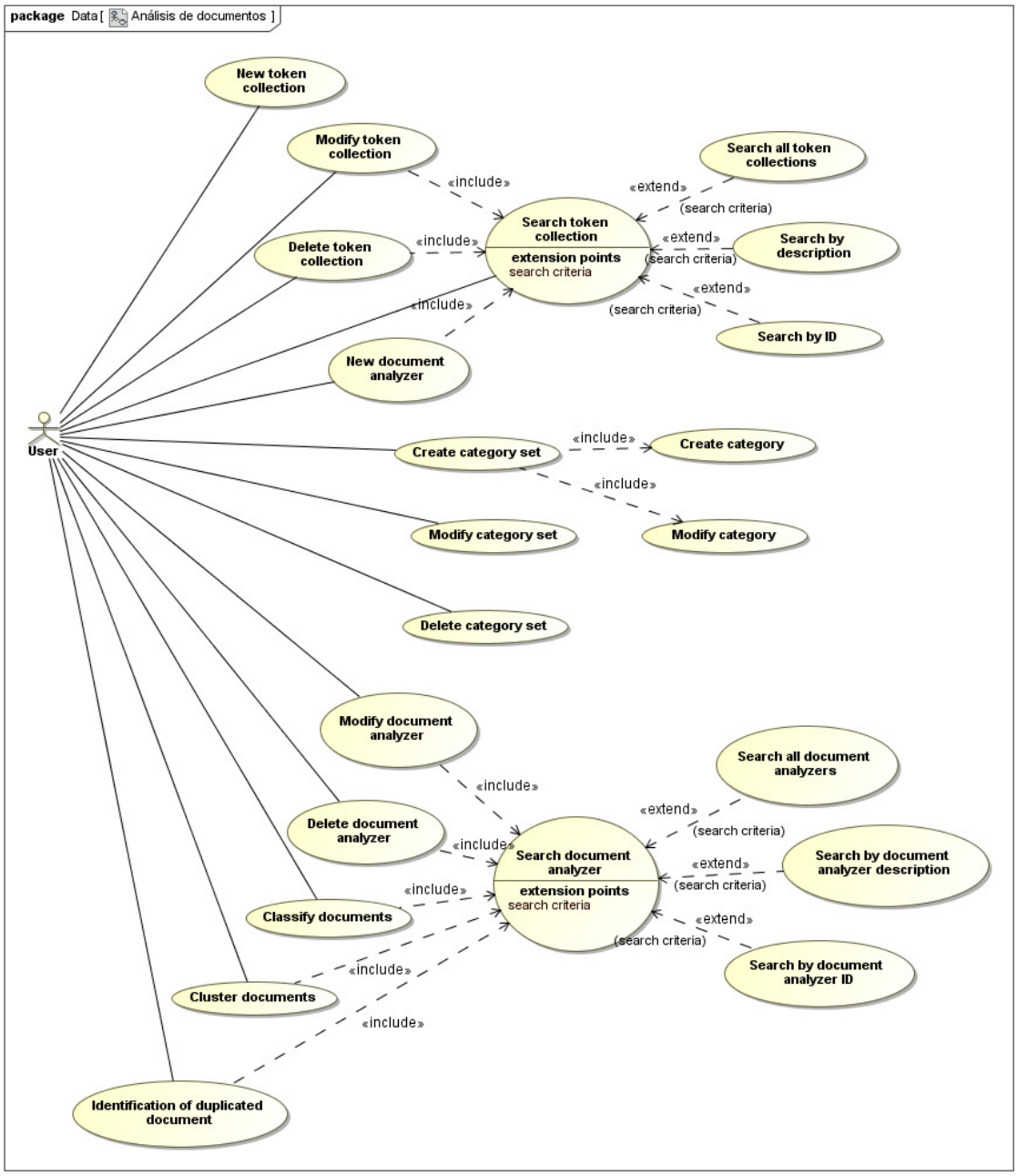


Figura 7. Casos de uso asociados al análisis de documentos.

Caso de uso:	**Actor:**
Crear nueva colección de tokens (New token collection)	Usuario

```
User story for the use case "New token collection"

Narrative:
In order to get a vector representation of a document
As a user of the application
I want to create a new token collection

Scenario: The user creates a token collection

Given the following token collection in the data base
Examples:
```

```
When the user creates the token collection [collection]
Examples:

Then the outcome should be [message]

Examples:
|Collection|message|
|Collection1|Token collection created correctly|
|Collection2|The token collection already exist|
|Collection3|Token collection created correctly|
```

Caso de uso: Modificar una colección de tokens (Modify token collection)	**Actor:** Usuario

```
User story for the use case "Modify token collection"

Narrative:
In order to maintain updated the information of token collection
As a user of the application
I want to modify a token collection stored in the database

Scenario: The user modifies a token collection

Given the following token collections in the database
Examples:

When the user modifies the token collection [collection]
Examples:

Then the outcome should [message]

Examples:
|Collection|message|
|Collection1|Token collection modified correctly|
|Collection2|The token collection does not exit|
```

Caso de uso: Eliminar una colección de tokens (Delete token collection)	**Actor:** Usuario

```
User story for the use case "Delete token collection"

Narrative:
In order to maintain updated the token collections in the database
As a user of the application
I want to delete a token collection stored in the database

Scenario: The user deletes a token collection

Given the following token collections in the database
Examples:

When the user deletes the collection [collection]
Examples:

Then the outcome should [message]

Examples:
|Collection|message|
|Collection1|Token collection deleted correctly|
|Collection2|The token collection does not exit|
```

Caso de uso:	**Actor:**

Buscar una colección de tokens (Search token collection)	Usuario

```
User story for the use case "Search token collection"

Narrative:
In order to retrieve a token Collection from the database
As a user of the application
I want to search a token collection stored in the database

Scenario: The user searches a token collection

Given the following token collections in the database
Examples:

When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the token collection

Given the following token collections in the database
Examples:

When the user searches all the token collections

Then the outcome should be

Examples:
```

Caso de uso: Crear analizador de documentos (New document analyser)	**Actor:** Usuario

```
User story for the use case "New document analyser"

Narrative:
In order to analyse documents
As a user of the application
I want to create a new document analyser

Scenario: The user creates a new document analyser

Given the following document analysers in the database
Examples:

When the user creates the document analyser [analyser]
Examples:

Then the outcome should be [message]

Examples:
|Analyser|message|
|Analyser1|Document analyser created correctly|
|Analyser2|The document analyser already exist|
|Analyser3|Document analyser created correctly|
```

Caso de uso:	**Actor:**

Modificar un analizador de documentos (Modify document analyser)	Usuario

```
User story for the use case "Modify document analyser"

Narrative:
In order to maintain updated the document analyzers in the database
As a user of the application
I want to modify a document analyser stored in the database

Scenario: The user modifies a document analyser

Given the following document analysers in the database
Examples:

When the user modifies the document analyser with [analyser]
Examples:

Then the outcome should [message]

Examples:
|Analyser|message|
|Analyser1|Document analyser modified correctly|
|Analyser2|The document analyser does not exit|
```

Caso de uso: Eliminar un analizador de documentos (Delete document analyser)	**Actor:** Usuario

```
User story for the use case "Delete document analyser"

Narrative:
In order to maintain updated the document analyzers
As a user of the application
I want to delete a document analyzer stored in the database

Scenario: The user deletes a document analyzer

Given the following document analyzers in the database
Examples:

When the user deletes the document analyzer [analyzer]
Examples:

Then the outcome should [message]

Examples:
|Analyser|message|
|Analyser1|Document analyser deleted correctly|
|Analyser2|The document analyser does not exit|
```

Caso de uso: Buscar un analizador de documentos (Search document analyser)	**Actor:** Usuario

```
User story for the use case "Search document analyser"

Narrative:
In order to retrieve a document analyser from the database
As a user of the application
I want to search a document analyser stored in the database

Scenario: The user searches a document analyser

Given the following document analysers in the database
Examples:
```

```
When the user introduces the following search query [query]
Examples:

Then the outcome should [message]

Examples:
|Query|message|
|Query1|There is not matches in the database|
|Query2|This are all the matches in the database|

Scenario: The user searches all the document analysers

Given the following document analysers in the database
Examples:

When the user searches all the document analysers

Then the outcome should be

Examples:
```

2.5. Análisis del modelo del dominio

Obtención de contenidos:

- ***Subscripción de contenidos RSS y Atom***

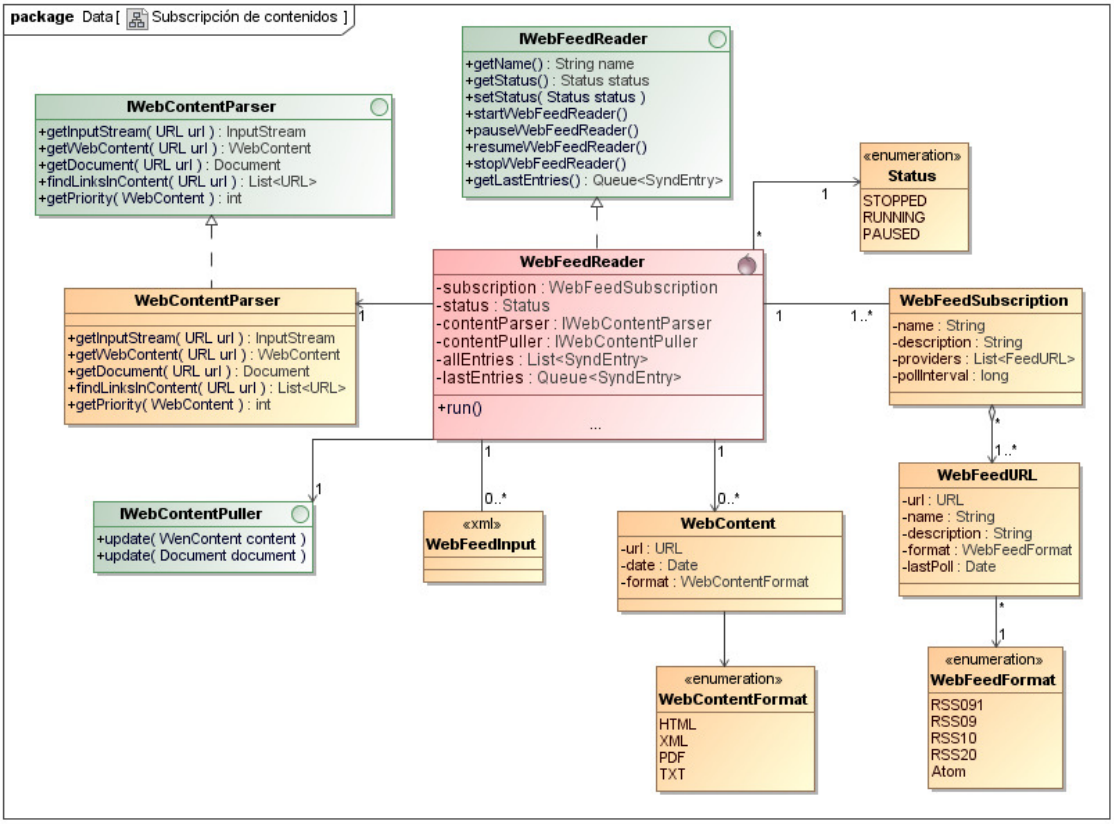


Figura 8. Modelo de dominio asociado a la subscripción de contenidos RSS y Atom.

- ***Rastreado de contenidos:***

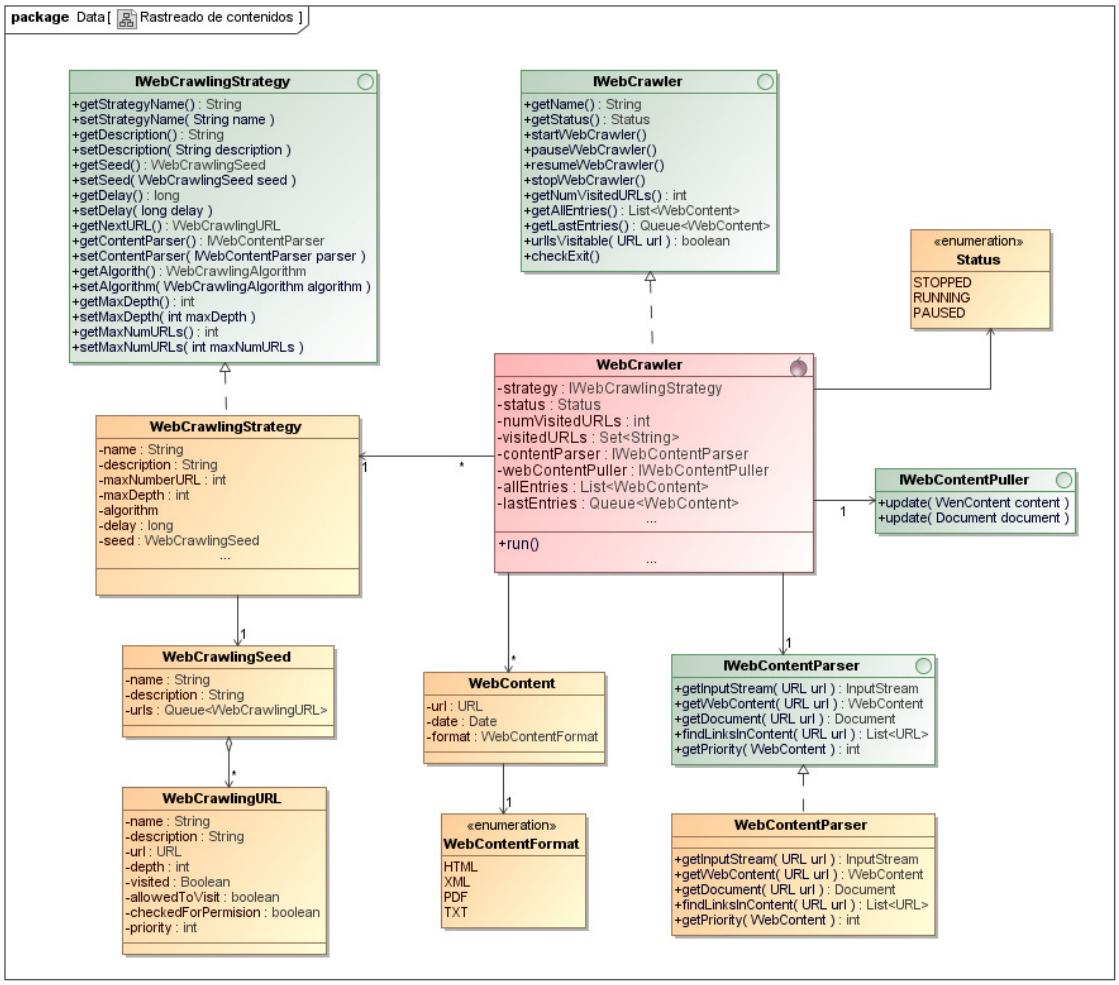


Figura 9. Modelo de dominio asociado al rastreo de contenidos en Internet (crawling).

Gestión de documentos

Para la modelización de contenidos y la creación de documentos se utilizará el marco establecido por el proyecto Apache Lucene [7, 8]. En Apache Lucene un documento es una unidad de almacenamiento de información que permite el indexado y la búsqueda. Un documento es básicamente una correspondencia entre campos (*Field*) y valores (*Value*), donde los campos es metainformación para el marcado y estructurado de los valores dentro del documento.

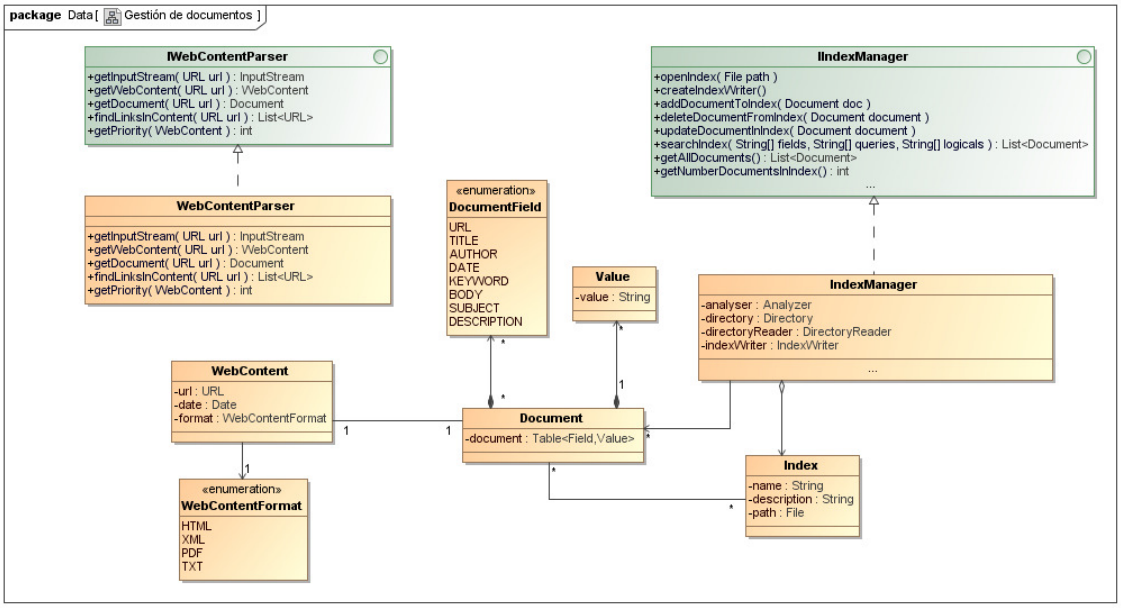


Figura 10. Modelo de dominio asociado a la gestión de documentos.

Datos de mercado

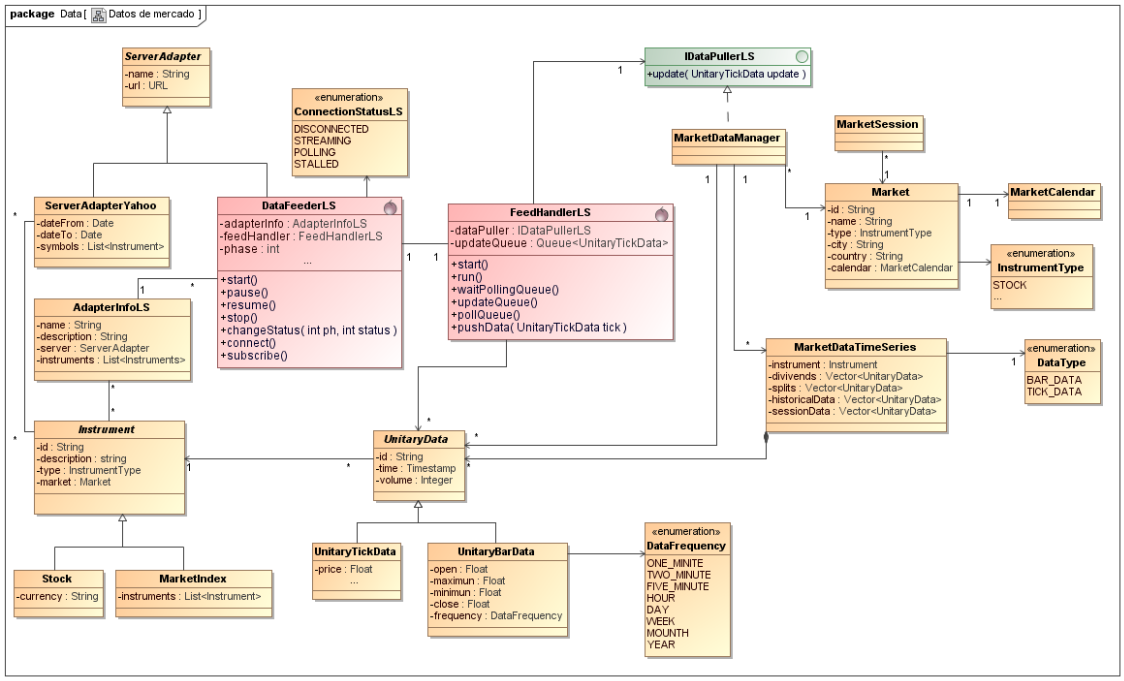


Figura 11. Modelo de dominio asociado a la obtención de datos de mercado.

Análisis de documentos:

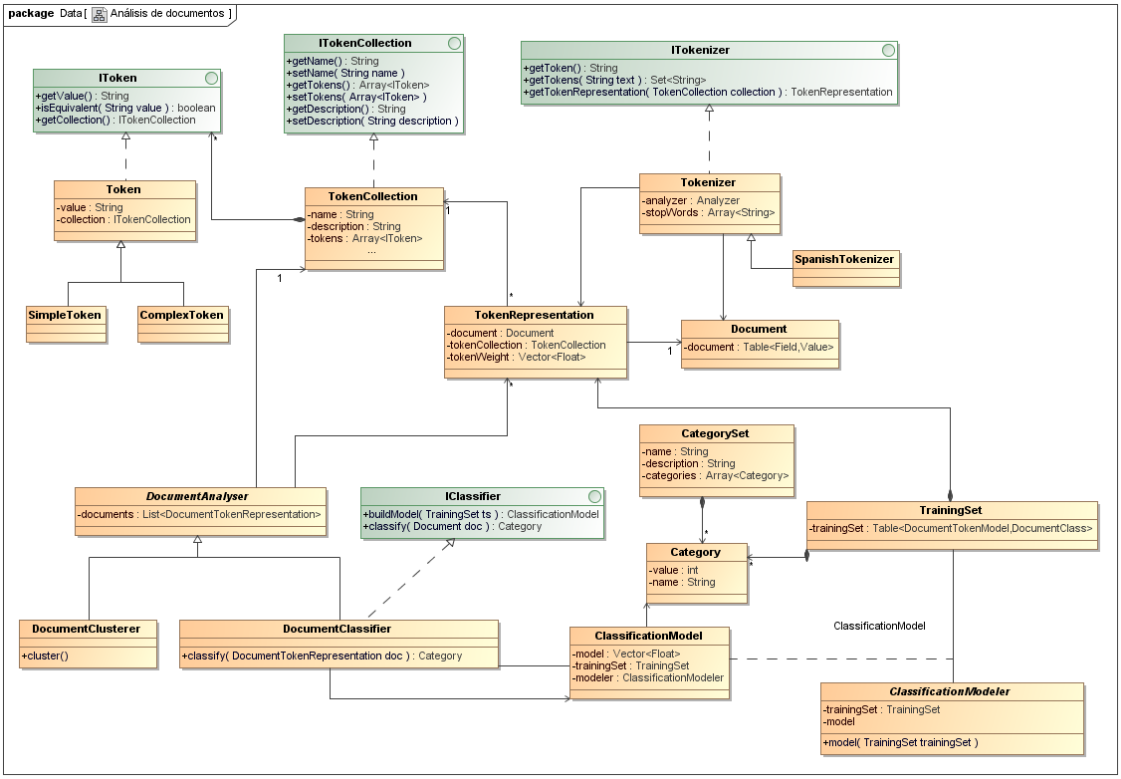


Figura 12. Modelo de dominio asociado al análisis de documentos.

2.6. Glosario del modelo del dominio

Se describen las entidades que forman parte del modelo de dominio asociado a todas las funcionalidades del sistema ordenadas alfabéticamente.

ComplexToken: clase para representar un Token complejo, normalmente una frase o conjunto de frases.

Category: representa una categoría de clasificación.

CategorySet: representa un conjunto cerrado de Category.

DataFrequency: clase enumerativa para representar los diferentes tipos de frecuencia en la que puede existir una dato de mercado.

Document: clase para representar de forma estandarizada para la aplicación un contenido de Internet.

DocumentAnalyser: clase abstracta para implementar un analizador de documentos.

DocumentClassifier: clase que implementa DocumentAnalyser que representa un clasificador de documentos.

DocumentClusterer: clase que implementa DocumentAnalyser que representa un agrupador de documentos.

ITokenizer: interfaz que especifica las operaciones de un Tokenizer

Tokenizer: clase para transformar un documento en un conjunto de Tokens.

SpanishTokenizer: subclse de Tokenizer usada para la tokenización de textos en español.

TokenRepresentation: clase para representar un documento de forma vectorizada utilizando una colección de Tokens.

WebFeedFormat: clase enumeración que representa los formatos en los que pueden enviar contenido los proveedores de contenidos.

WebFeedInput: contenido enviado por un proveedor de contenidos tras una consulta. Puede contener una o varias entradas.

IWebContentParser: interfaz que define las operaciones de un WebContentParser.

WebContentParser: clase para interpretar y analizar un contenido Web.

WebFeedReader: clase controladora que consulta los proveedores de contenidos utilizando una FeedSubscription y procesa los contenidos enviados utilizando un FeedInputParser para obtener una colección de FeedEntry.

IWebContentPuller: Interfaz que implementa la clase utilizada por un obtentor de contenidos Web para gestionar su destino, por ejemplo puede ser implementada por un manejador de persistencia de datos.

Status: clase enumeración que define los estados básicos de algunas clases controladores.

WebFeedSubscription: clase agrupación de una colección de proveedores de contenidos FeedURL.

WebFeedURL: clase que extiende *URL*, y representa una *URL* asociada con una fuente de sindicación de contenidos en formatos *RSS* o *Atom*.

DocumentField: clase para representar los campos que componen un documento.

Index: clase que representa el indexado de un grupo de documentos.

IIndexManager: interfaz que especifica las operaciones de un IndexManager

IndexManager: clase para la gestión de los índices en la base de datos.

Instrument: clase que representa un instrumento financiero.

InstrumentType: clase enumerativa para representar los diferentes tipos de instrumentos financieros.

Market: clase que representa un mercado financiero.

MarketCalendar: clase que representa el calendario de un mercado financiero.

MarketDataManager: clase para la gestión de los datos de mercado.

MarketDataTimeSeries: clase que representa una serie temporal de datos de Mercado para un instrumento financiero.

MarketSession: clase que representa una sesión de negociación de un mercado financiero.

ServerAdapter: clase abstracta para representar el controlador que gestiona la connexion con un servidor de datos de mercado.

DataFeederLS: gestiona la conexion con un servidor *streaming* de datos de mercado de tipo LightStreamer [].

FeedHandlerLS: gestiona la utilización de los datos obtenidos por un DataFeederLS.

ServerAdapterYahoo: implementación de ServerAdapter, para gestionar la conexion con el servidor de datos de Mercado de Yahoo.

SimpleToken: clase para representar un término simple de un documento, normalmente una única palabra.

Stock: clase para representar un instrumento financiero del tipo acción.

IToken: interfaz que especifica las operaciones de un Token.

Token: clase para representar una parte del contenido de un documento, normalmente una palabra o una frase.

TokenCollection: clase para representa una colección de Tokens que permita representar un documento de forma vectorizada.

UnitaryBarData: clase que implementa UnitaryData y que representa la cotización de un valor durante un periodo de tiempo.

UnitaryData: clase abstracta que representa un valor discreto para la cotización de un instrumento financiero en un mercado.

UnitaryTickData: clase que implementa UnitaryData y que representa una transacción de un instrumento financiero.

WebContent: clase para representar un contenido procedente de Internet.

WebContentFormat: clase enumerativa para representar posibles formatos de contenidos obtenidos en Internet.

IWebCrawler: interfaz que especifica las operaciones de un **WebCrawler**.

WebCrawler: clase para rastrear contenidos en Internet mediante estrategias de WebCrawling.

WebCrawlingSeed: clase para representar una semilla utilizada por un WebCrawler para iniciar una estrategia de rastreo.

IWebCrawlingStrategy: interfaz que especifica las operaciones de una WebCrawlingStrategy.

WebCrawlingStrategy: clase para representar una estrategia de rastreo utilizada por un WebCrawler.

WebCrawlingURL: Clase para representar una dirección *URL* utilizada por una entidad WebCrawler.

2.7. Identificación de clases frontera, control y entidad

Obtención de contenidos

- ***Subscripción de contenidos RSS y Atom***

Crear una nueva subscripción:

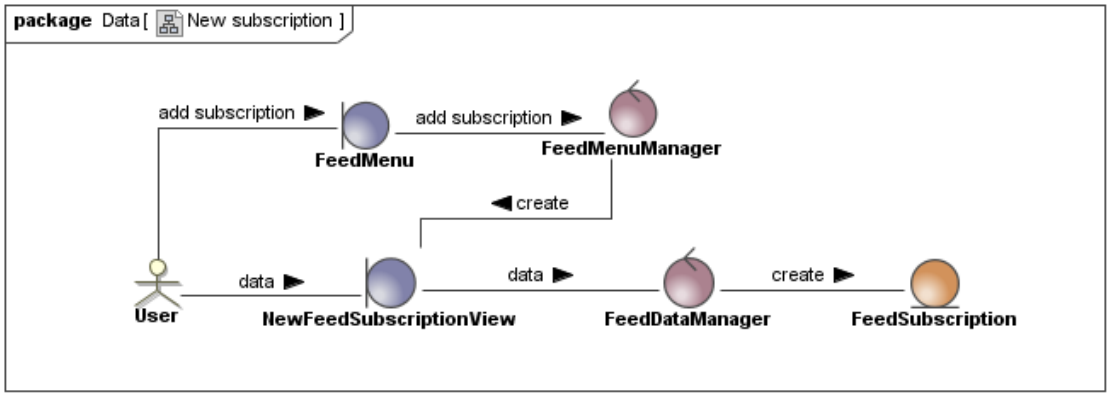


Figura 13. Diagrama de colaboración asociado a la creación una nueva subscripción.

Modificar una subscripción existente:

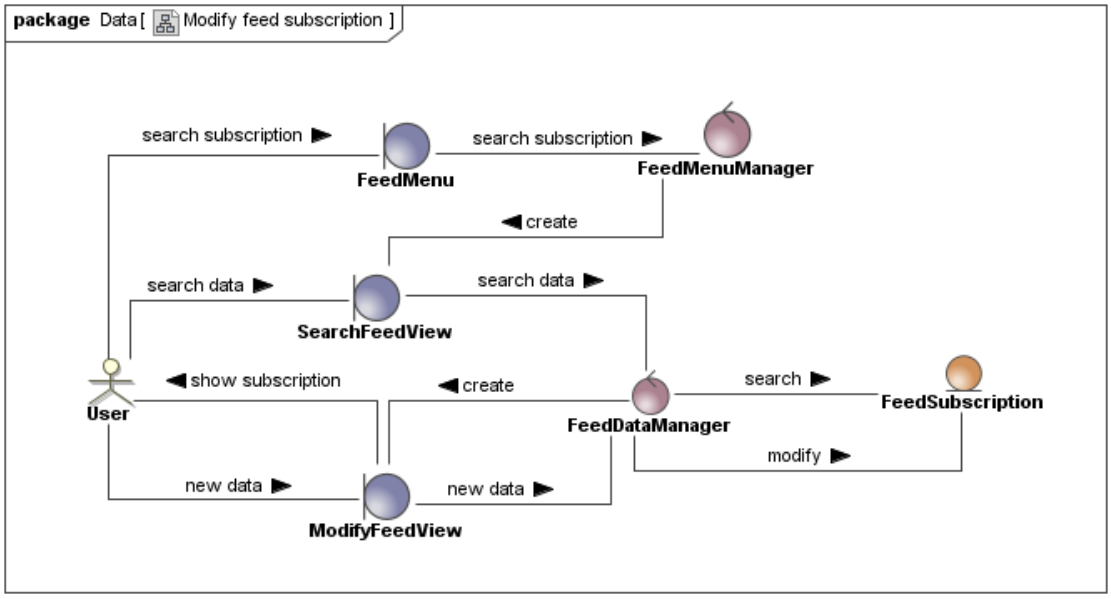


Figura 14. Diagrama de colaboración asociado a la modificación una subscripción existente

Eliminar una subscripción existente:

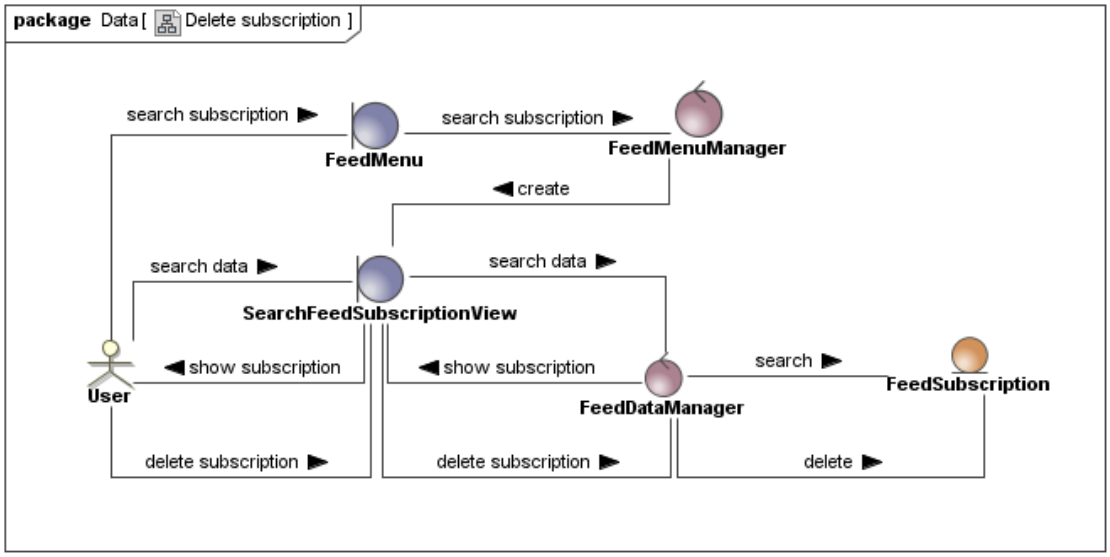


Figura 15. Diagrama de colaboración asociado a la eliminación de una subscripción existente

Funcionalidades del lector de subscripción:

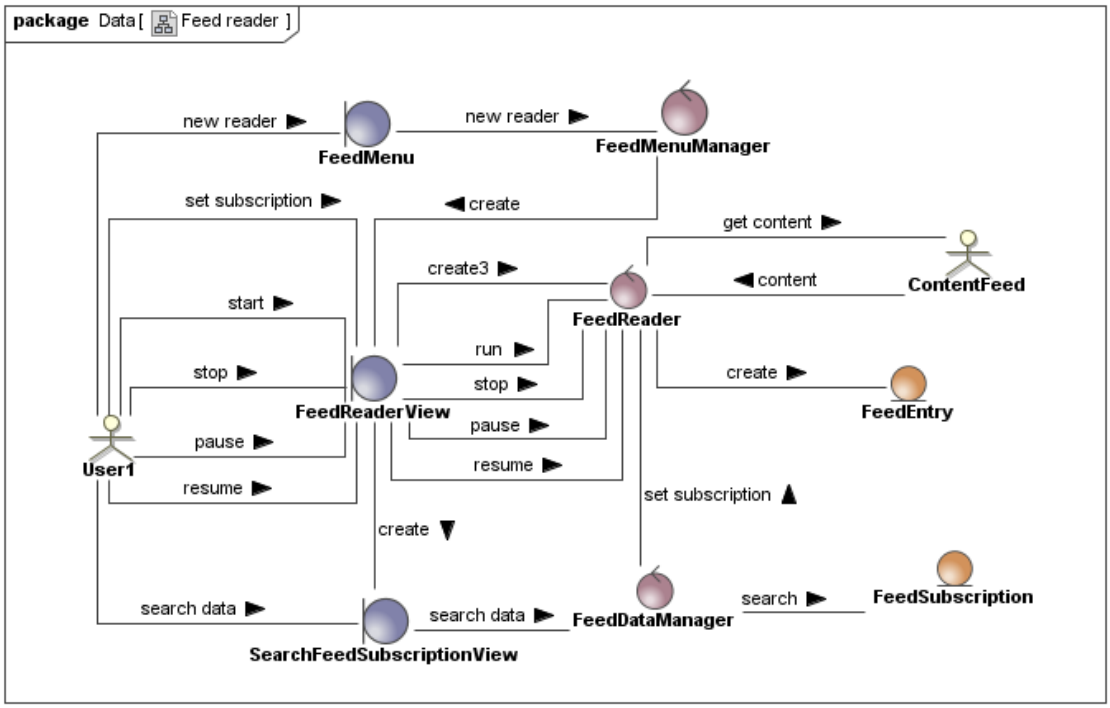


Figura 16. Diagrama de colaboración asociado a las funcionalidades del lector de subscripciones

- ***Rastreado de contenidos***

Crear estrategia de rastreo:

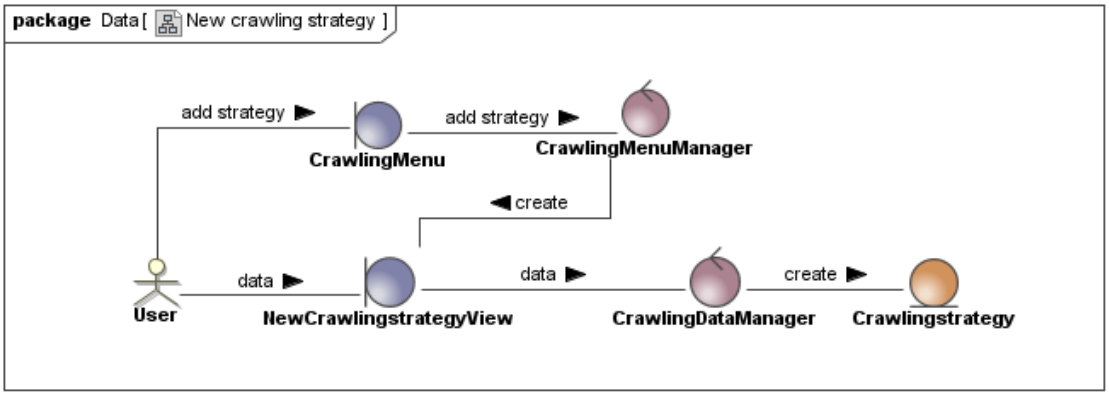


Figura 17. Diagrama de colaboración asociado a la creación de una estrategia de rastreo

Modificar estrategia de rastreo:

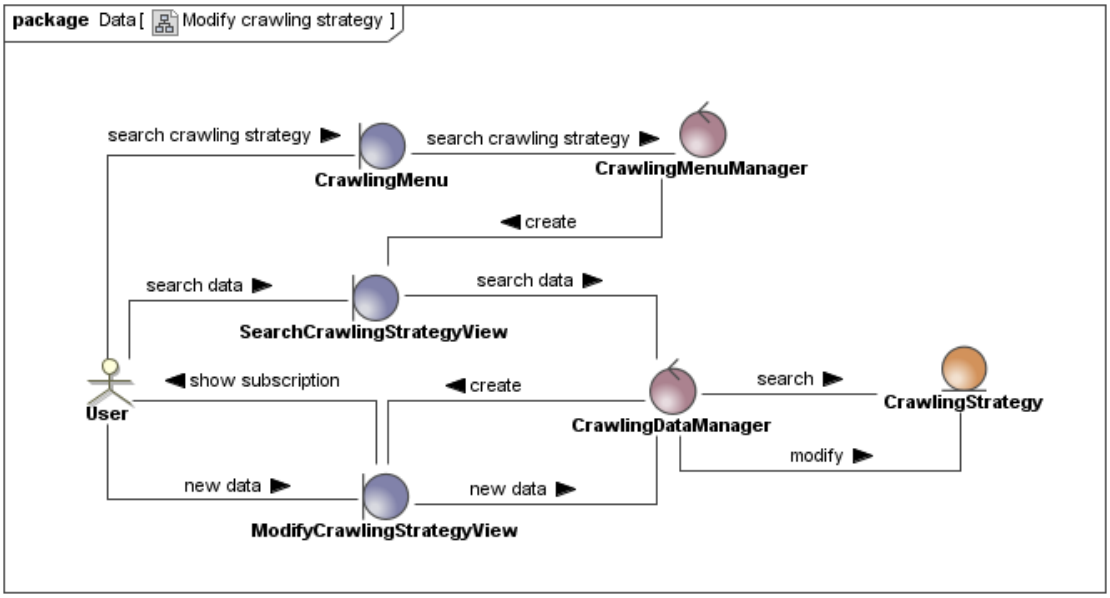


Figura 18. Diagrama de colaboración asociado a la modificación de una estrategia de rastreo

Borrar estrategia de rastreo:

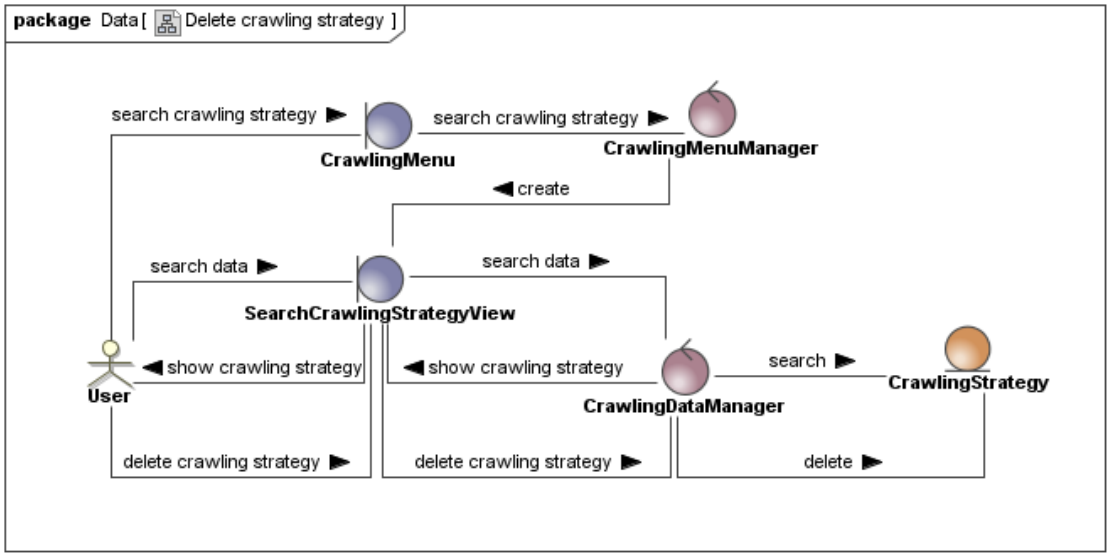


Figura 19. Diagrama de colaboración asociado al borrado de una estrategia de rastreo

Crear semilla de rastreo:

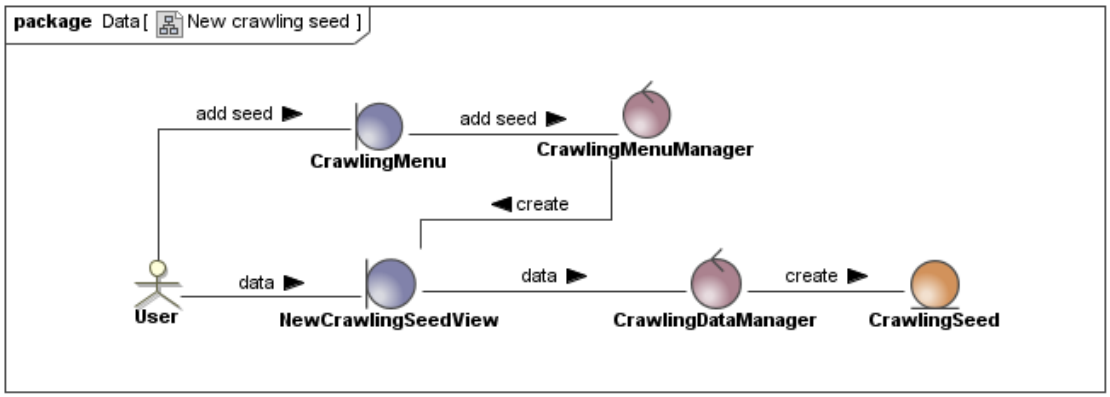


Figura 20. Diagrama de colaboración asociado a la creación de una semilla de rastreo

Modificar una semilla de rastreo:

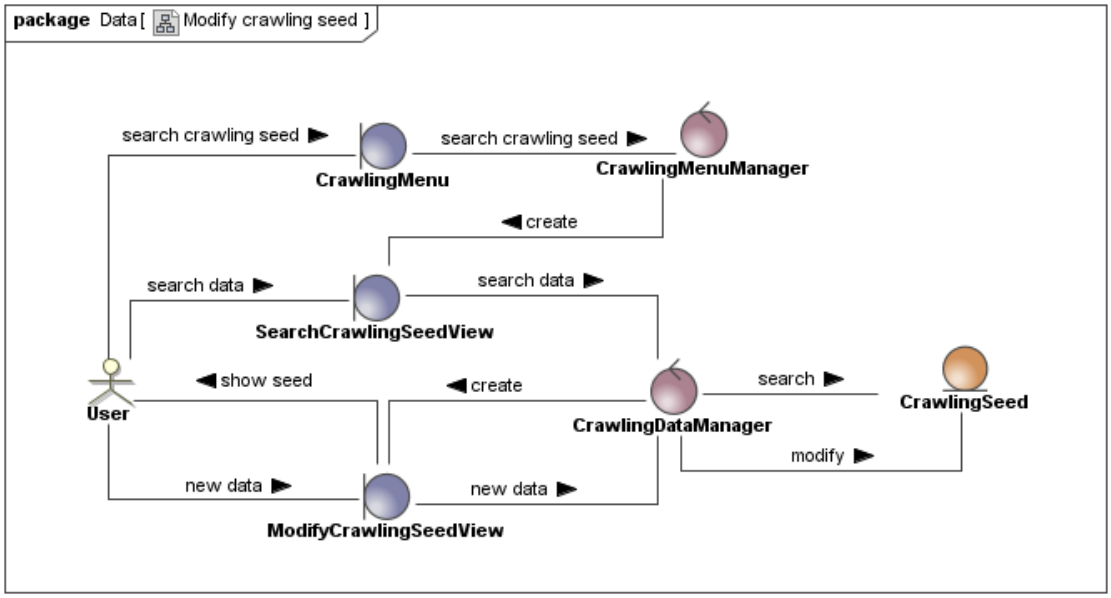


Figura 21. Diagrama de colaboración asociado a la modificación de una semilla de rastreo

Borrar semilla de rastreo de contenidos:

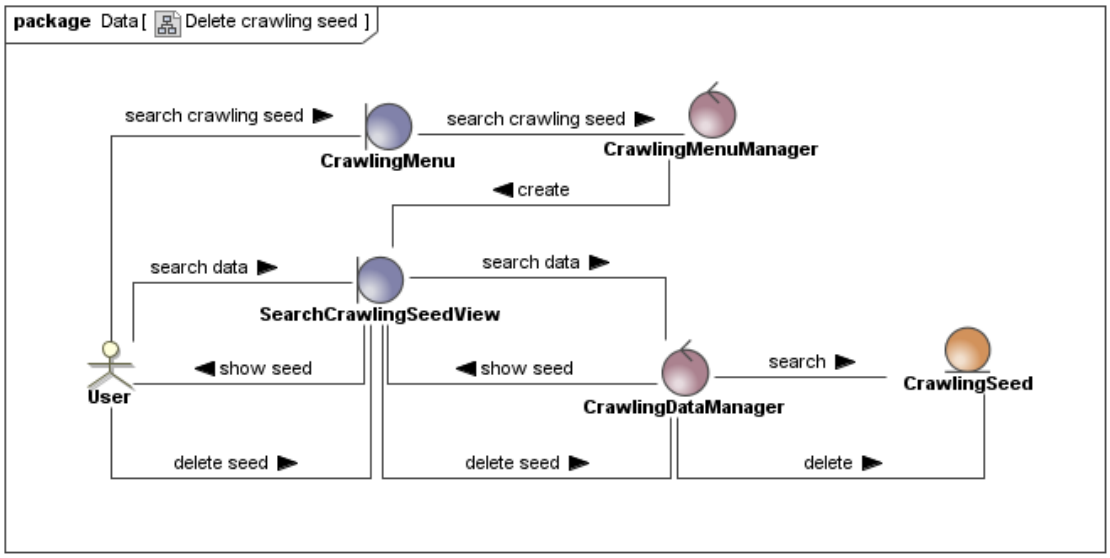


Figura 22. Diagrama de colaboración asociado a la eliminación de una semilla de rastreo de contenidos

Funcionalidades de rastreador de contenidos:

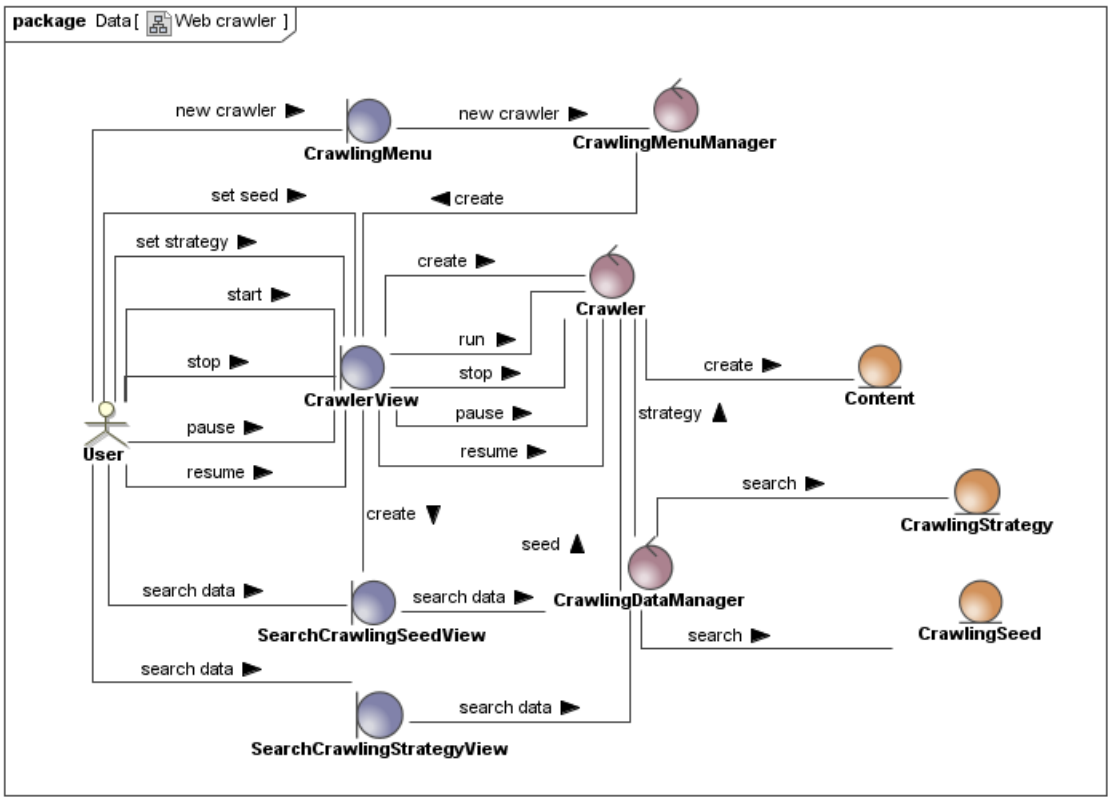


Figura 23. Diagrama de colaboración asociado a las funcionalidades del rastreador de contenidos

Gestión de contenidos:

Crear nuevo índice:

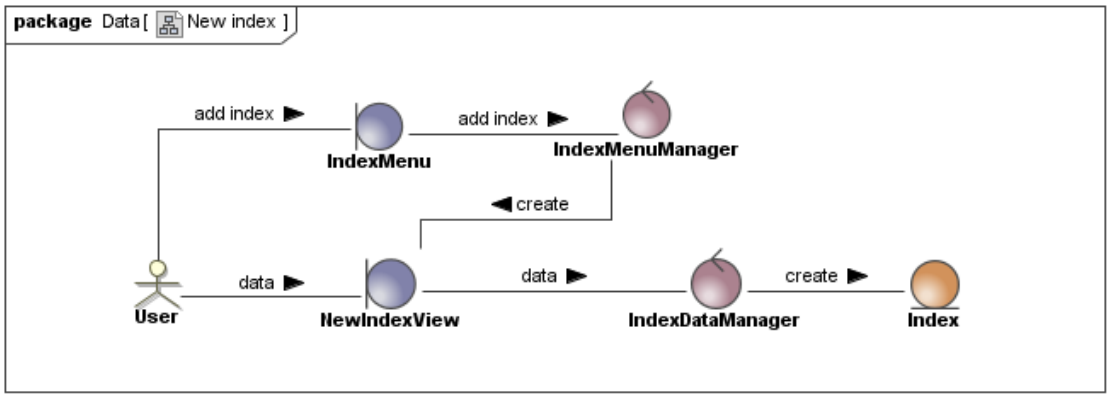


Figura 24. Diagrama de colaboración asociado a la creación de un nuevo índice

Eliminar un índice existente:

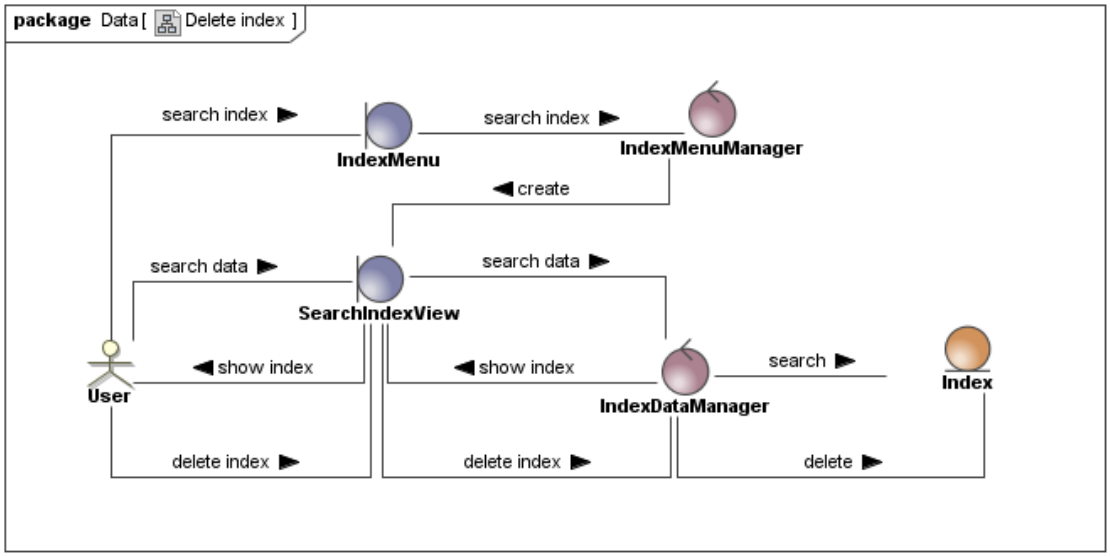


Figura 25. Diagrama de colaboración asociado a la eliminación de un índice existente

Añadir documento a un índice:

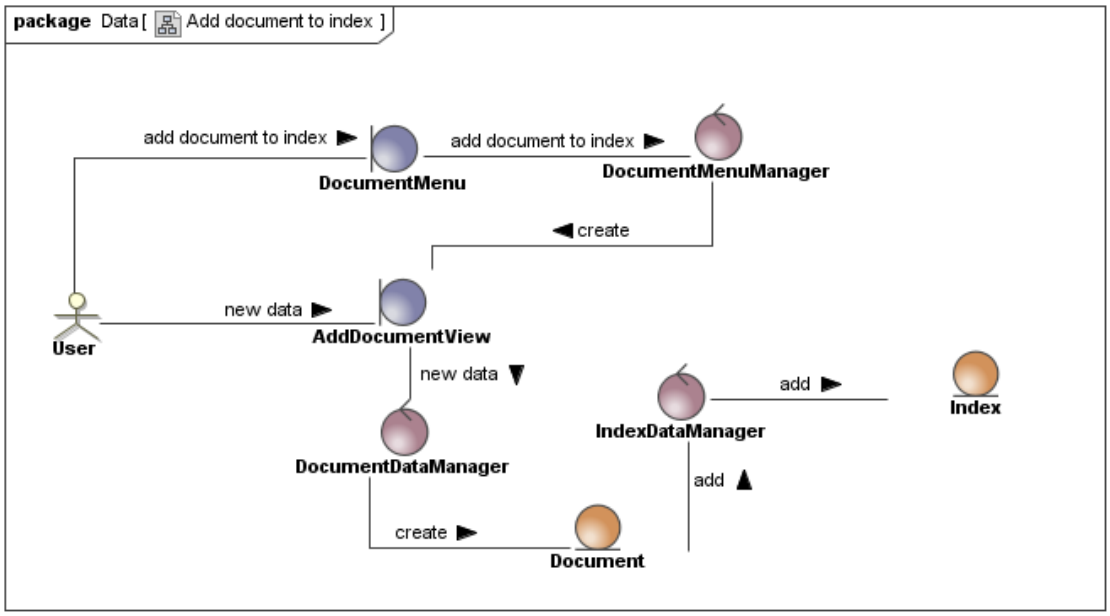


Figura 26. Diagrama de colaboración asociado a la adición de un documento a un índice

Borrar documento:

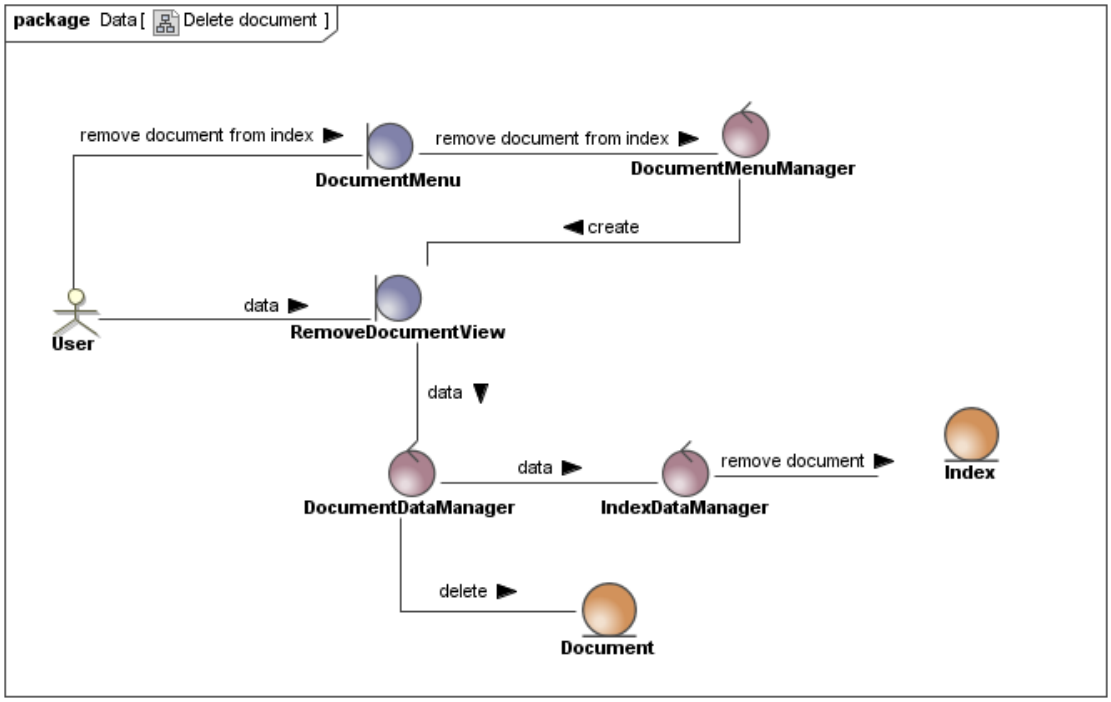


Figura 27. Diagrama de colaboración asociado al borrado de un documento

Buscar documento:

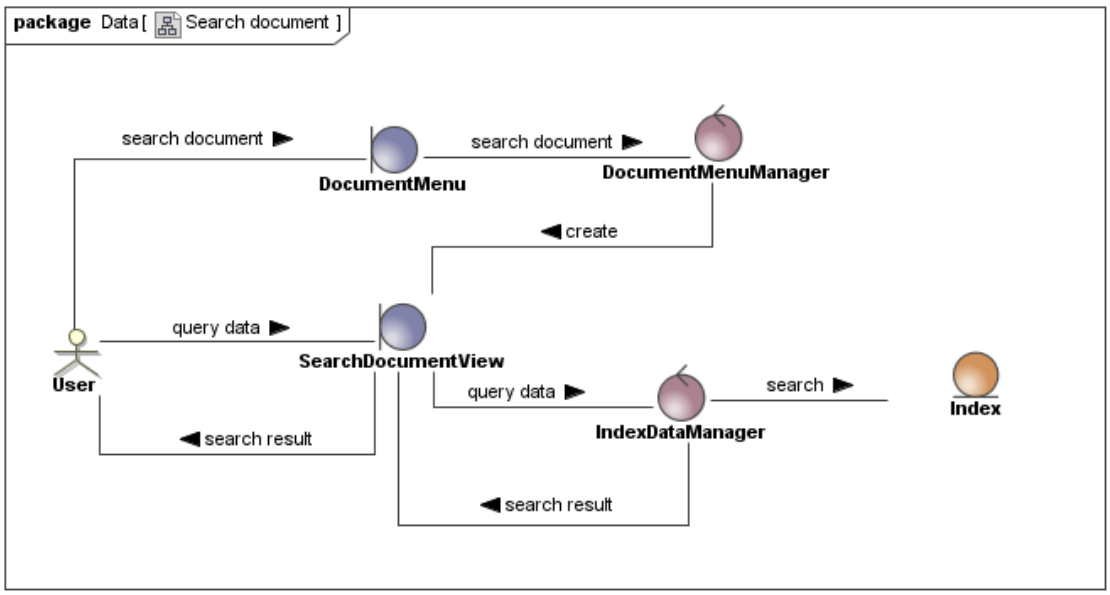


Figura 28. Diagrama de colaboración asociado a la búsqueda de documentos

Obtención de datos de mercado

Crear nueva conexion de datos de mercado streaming:

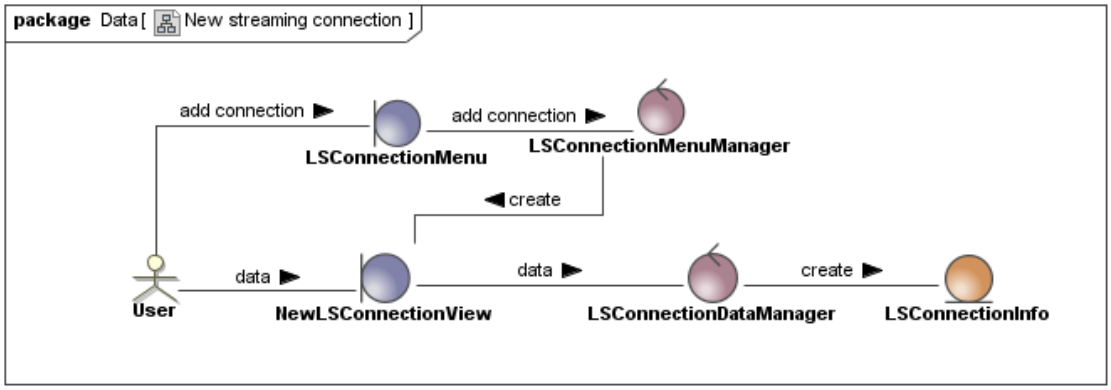


Figura 29. Diagrama de colaboración asociado a la creación de una nueva conexión de datos de mercado streaming

Modificar una connexion de datos de mercado streaming:

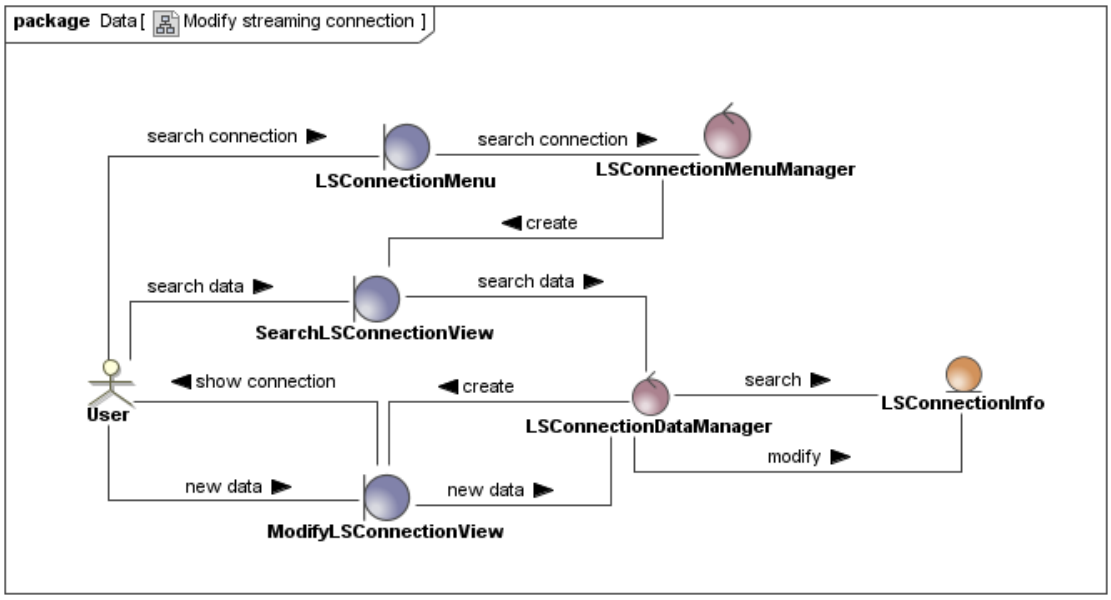


Figura 30. Diagrama de colaboración asociado a la modificación de una conexión de datos de mercado streaming

Eliminar una connexion de datos de mercado streaming:

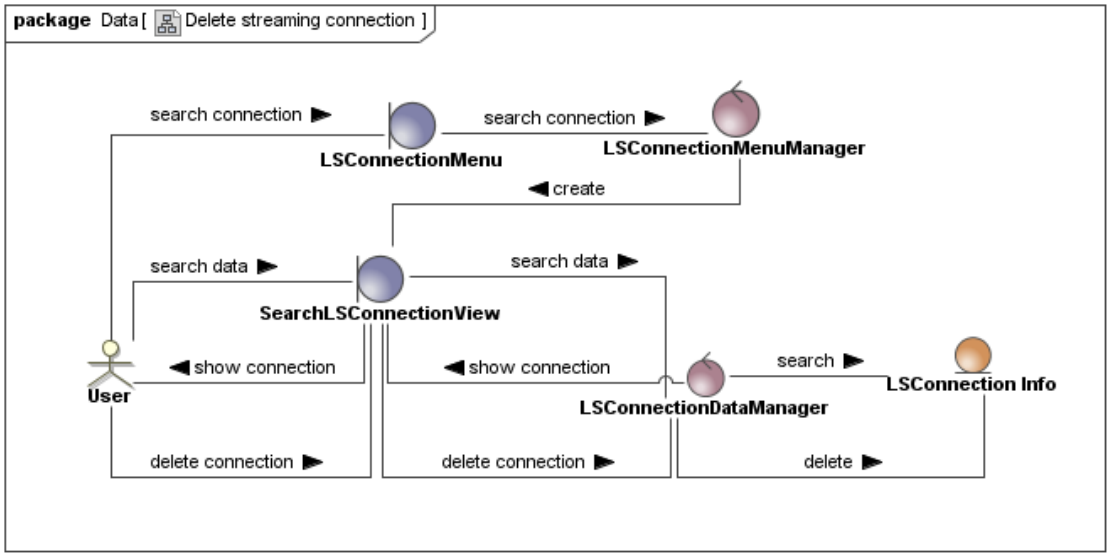


Figura 31. Diagrama de colaboración asociado a la eliminación una conexion de datos de mercado streaming

Funcionalidades del adaptador streaming de datos de mercado:

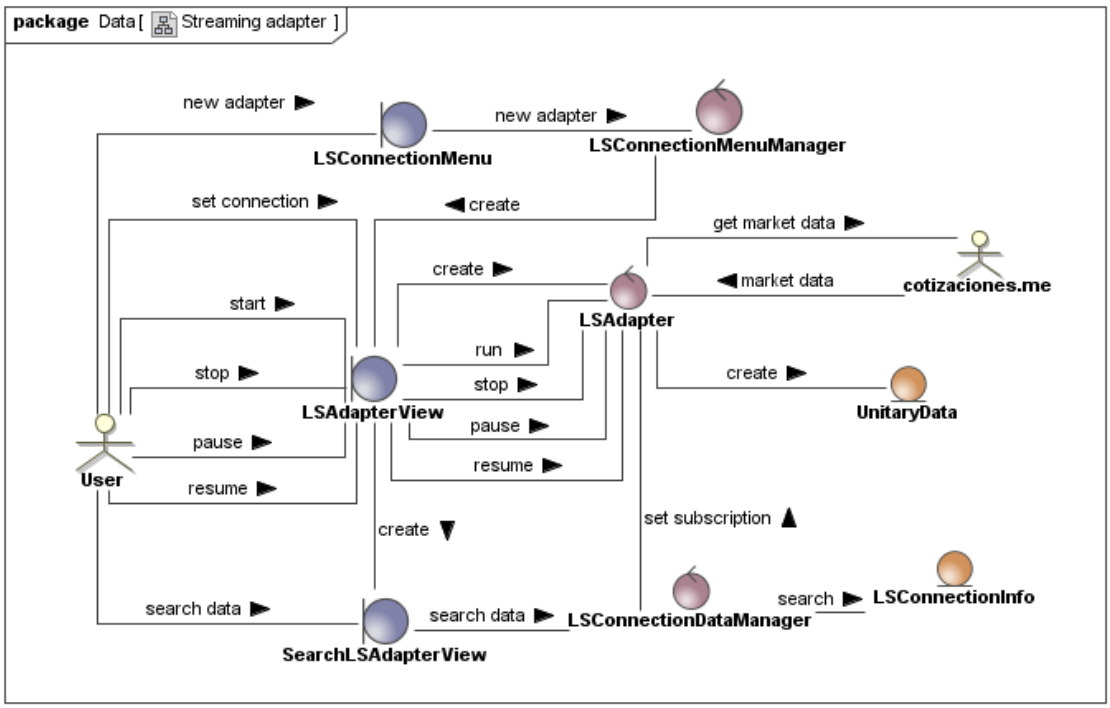


Figura 32. Diagrama de colaboración asociado a las funcionalidades del adaptador streaming de datos de mercado

Descargar datos de Mercado de Yahoo:

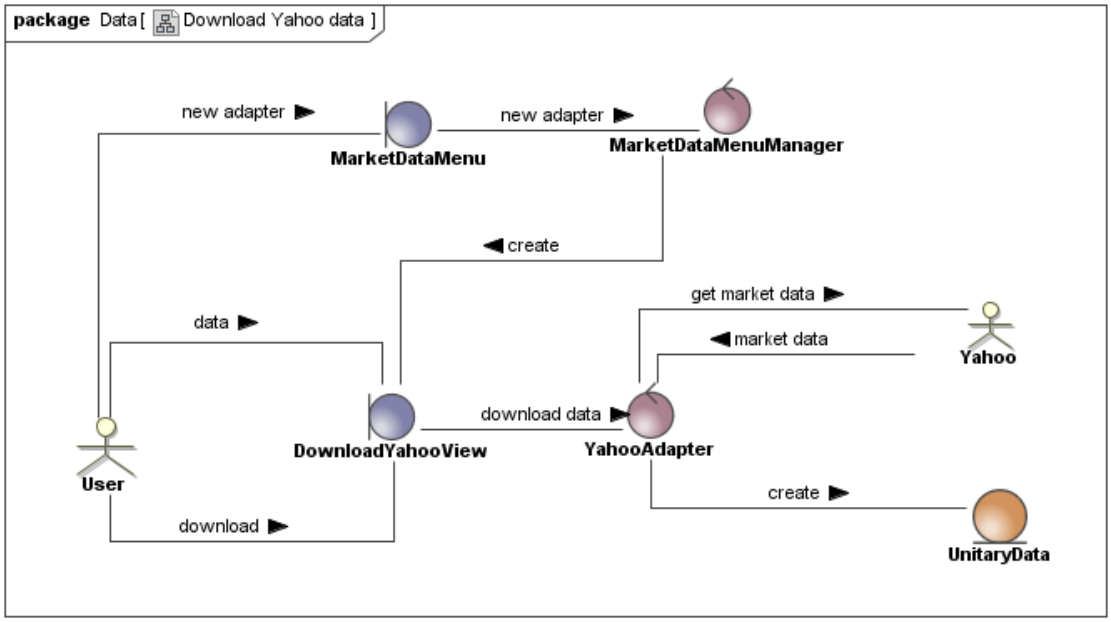


Figura 33. Diagrama de colaboración asociado a la descarga de datos de Mercado de Yahoo

Análisis de documentos:

Crear nueva colección de tokens:

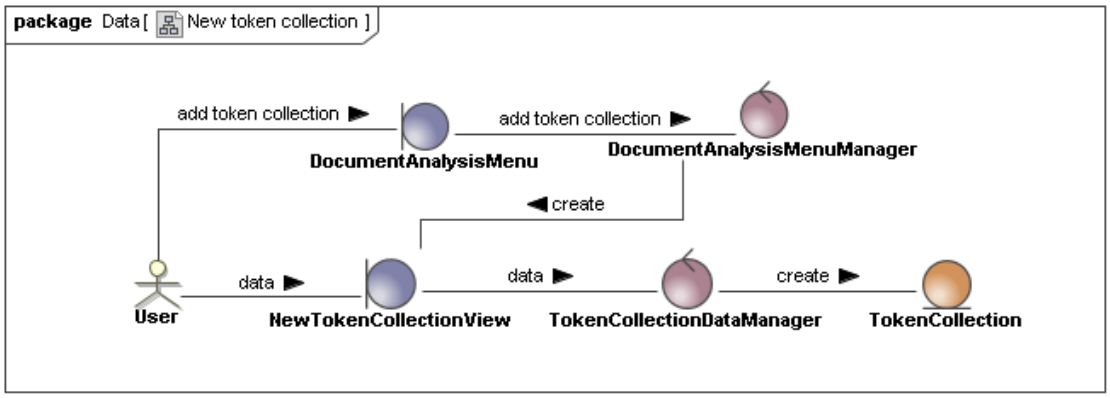


Figura 34. Diagrama de colaboración asociado a la creación de una colección de tokens

Modificar una colección de tokens existente:

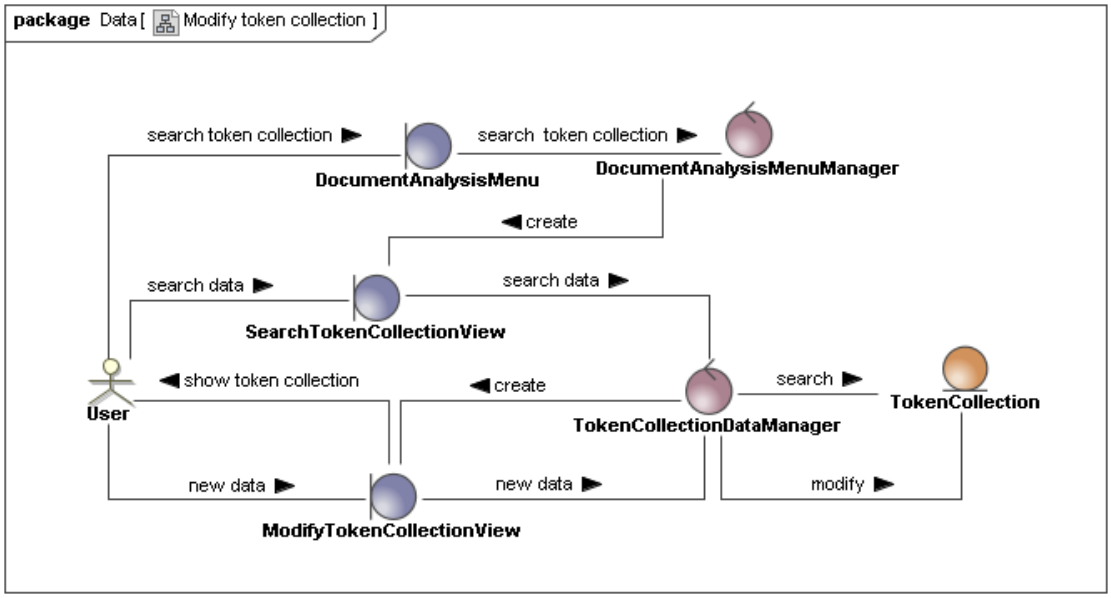


Figura 35. Diagrama de colaboración asociado a la modificación de una colección de tokens

Eliminar una colección de tokens existente:

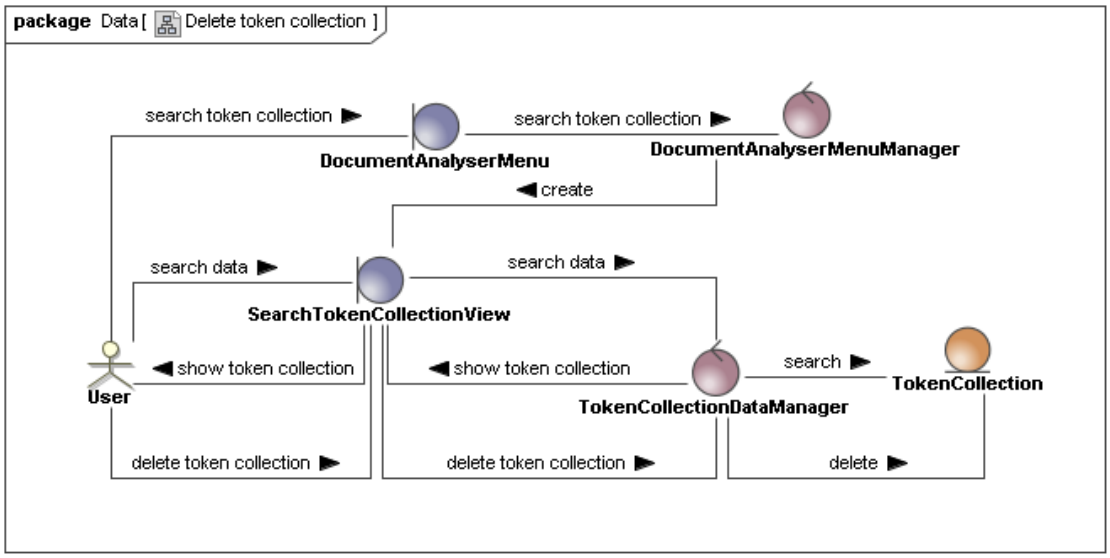


Figura 36. Diagrama de colaboración asociado al borrado de una colección de tokens

Crear un nuevo analizador de documentos:

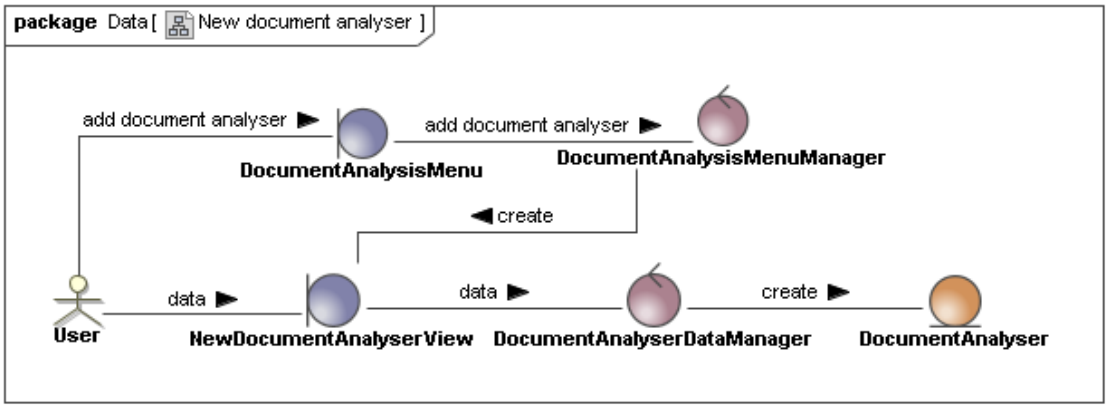


Figura 37. Diagrama de colaboración asociado a la creación de un analizador de documentos

Modificar un analizador de documentos existente:

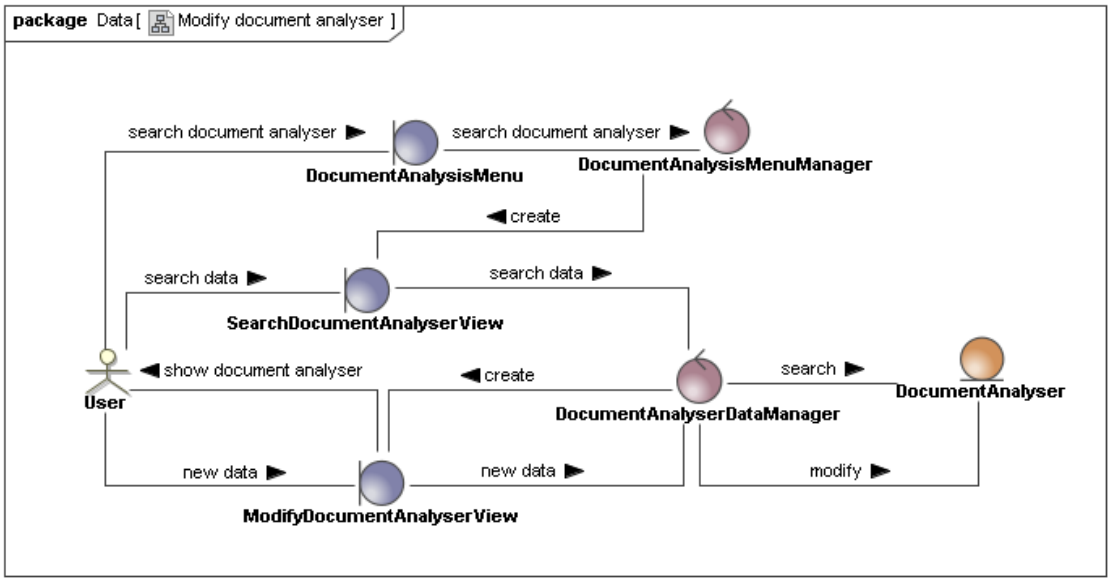


Figura 38. Diagrama de colaboración asociado a la modificación de un analizador de documentos

Eliminar un analizador de documentos existente:

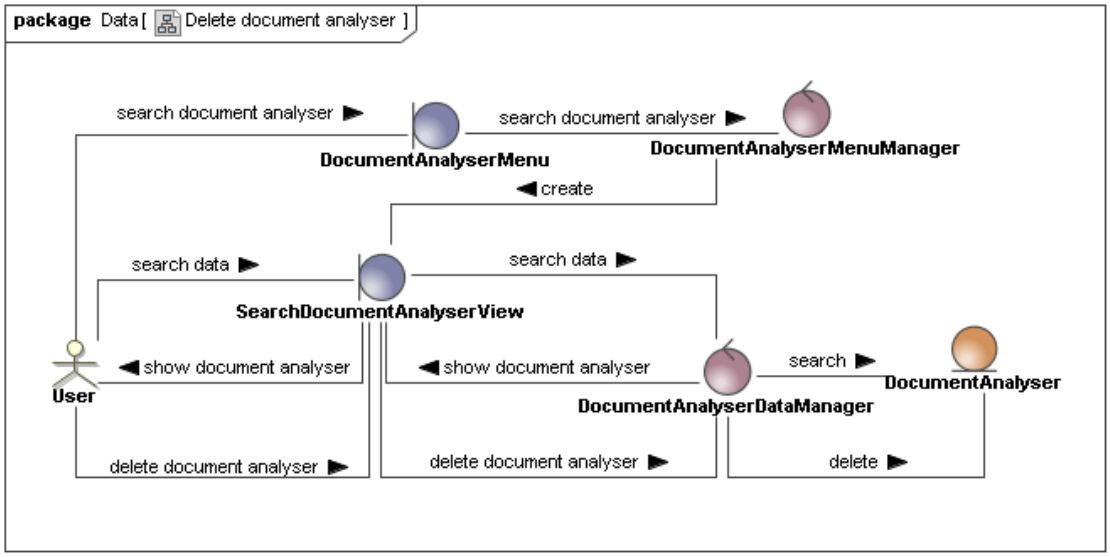


Figura 39. Diagrama de colaboración asociado al borrado de un analizador de documentos

3. Ejecución del plan de trabajo — Diseño

3.1. Formato de contenidos y documentos

En la aplicación los contenidos obtenidos de Internet pasan por diferentes formatos según el uso que hacen de ellos los diferentes componentes del sistema. La siguiente tabla muestra los posibles formatos y los componentes de la aplicación donde son utilizados.

<table>
<tr><td>Obtención de contenidos
 
</td><td>URL: http://www.eleconomista.es/mercados-cotizaciones/noticias/5352444/11/13/IEB-En-2014-el-Ibex-se-movera-entre-los-11000-y-los-11500-puntos-con-la-prima-de-riesgo-en-los-125150-puntos.html

Format: HTML

Date: 28/11/2013 - 14:10</td></tr>
<tr><td>Gestión de contenidos
 
</td><td>
<table>
<tr><td>TITLE</td><td>IEB: "En 2014 el Ibex se moverá entre los 11.000 y los 11.500 puntos"</td></tr>
<tr><td>AUTHOR</td><td>I. M. Gaspar</td></tr>
<tr><td>URL</td><td>http://www.eleconomista.es/mercados-cotizaciones/noticias/5352444/11/13/IEB-En-2014-el-Ibex-se-movera-entre-los-11000-y-los-11500-puntos-con-la-prima-de-riesgo-en-los-125150-puntos.html</td></tr>
<tr><td>DATE</td><td>28/11/2013 - 14:10</td></tr>
<tr><td>TEXT</td><td>Esta es una de las conclusiones que se desprende del informe 'Perspectiva económicas para 2014' realizado por el Departamento de Investigación de Instituto de Estudios Bursátiles (IEB) que su coordinador, Miguel Ánge Bernal, ha presentado esta mañana.

En este sentido, Bernal ha destacado que en 2014 no descarta ver al Ibex 3 moverse entre los 11.000 y los 11.500 puntos, aunque no por unos dato macroeconómicos brillantes, si no más bien por la caída de la prima d riesgo. Así, el estudio sitúa al diferencial español entre los 125-150 punto con la rentabilidad del bono español entre el 4% y por debajo de este nivel (e riesgo país ronda ahora los 246 puntos y el rendimiento del bono el 4,16%).

Esa caída de la prima de riesgo, según explica Bernal ayudará a que l capacidad de financiación de la economía española continúe siendo positiv y tienda al alza, sustentada también en una tendencia a mejorar la previsiones de las casas de rating sobre España, la entrada de capita inversor en todos los sectores y una relajación de la fuerte tendencia a l aversión al riesgo. Eso sí, la financiación a pymes y familias llegará de form muy lenta.

¿Será entonces el mercado español una oportunidad de inversión el próxim año? "Creo que las mejores oportunidades surgirán en los países que ha sido rescatados como España e Italia y dentro de España me quedaría co los bancos cuya reestructuración deja margen para ver subidas, aunque n creo que lleguen a los niveles máximos que alcanzaron", indica Berna Además, también se inclina por "el dólar como activo y rentas fijas privada de las compañías españolas".

Eso sí, el profesor hace hincapié en que el crecimiento que manejan par España en 2014 es del 0,9%, "un nivel muy muy bajo, ya que significa crece en el trimestre un 0,2%. España debería crecer al 2%-3%, entonces veríamo mejorar el empleo". Precisamente prevén que la tasa de desempleo se sitú en el 25,6% con una tendencia de creación de puestos de trabaj temporales.

En cuanto a la zona euro, creen que se mantendrá el ritmo de atonía y que e PIB crecerá entre el 0,5% y 1%, frente al 2% que espera el Banco Centra Europeo (BCE). Este crecimiento será desigual entre los países, co Alemania marcando la senda del crecimiento moderado pero con nuevo focos de preocupación como Francia: "Es un peso pesado y si no s recupera entraremos en una 'dictadura' del BCE", señala Bernal.

En este contexto, para el IEB la máxima preocupación del BCE debería ser e contexto de precios muy contenidos, lo que dificulta a medio y largo plazo l salida de la depresión. En cuanto a la política monetaria que el organism</td></tr>
</table>
</td></tr>
</table>

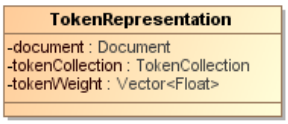
llevará a cabo el año próximo el estudio recoge que esperan que los tipos de interés bajen al 0,10% lo que apoyaría que el tipo de interés de la facilidad de depósito se sitúe en el -0,10%. Además, esperan medidas no ortodoxas, como la compra de carteras a bancos, o la instrumentación de fondos para el incremento del crédito a pymes y familias.

EEUU crecerá, pero no de forma espectacular

Según las previsiones lanzadas por el IEB, el PIB de la primera economía del mundo crecerá entre un 1,5% y un 2%, "algo positivo pero no espectacular, ya que la población continuará creciendo, cosa que no ocurre en Europa, por lo que un crecimiento del 2% nos indica que simplemente se mantendrá el nivel de vida", explica Miguel Ángel Bernal.

En cuanto a los mercados emergentes "continuarán siendo un foco de crecimiento pero no va a ser tan brillante". A este respecto, no prevén que constituyan un problema económico en 2014, pero no van a ser un motor de crecimiento.

Análisis de contenidos



Ibex	alta rentabilidad	baja rentabilidad	oportunidad	negativo	positivo	preocupación
W1	W2	W3	W4	W5	W6	W7

Figura 40. Formatos adoptados por los contenidos y documentos.

3.2. Diseño de la arquitectura del sistema

Para el diseño del sistema se adopta una arquitectura en tres capas, capa de presentación, capa de negocio y capa de integración. En principio se tratará que esta decisión de diseño sea lo más independiente de la implementación de manera que esta arquitectura de capas se pueda desarrollar como una aplicación local o bien adoptar una arquitectura cliente-servidor. En el primer caso la capa de presentación se limitaría a la interfaz de usuario. En el segundo caso la capa de presentación se desarrollaría como un componente cliente siendo la capa de negocio y la capa integración el componente servidor.

La motivación de esta decisión de diseño es otorgar flexibilidad de manera que los componentes “Invoke” en la capa de presentación y “Manager” en la capa de negocio se puedan comunicar tanto de forma local o bien de forma remota si por ejemplo se decide desarrollar la aplicación como objetos distribuidos.

En La figura siguiente se muestra la estructura de componentes de la arquitectura del sistema:

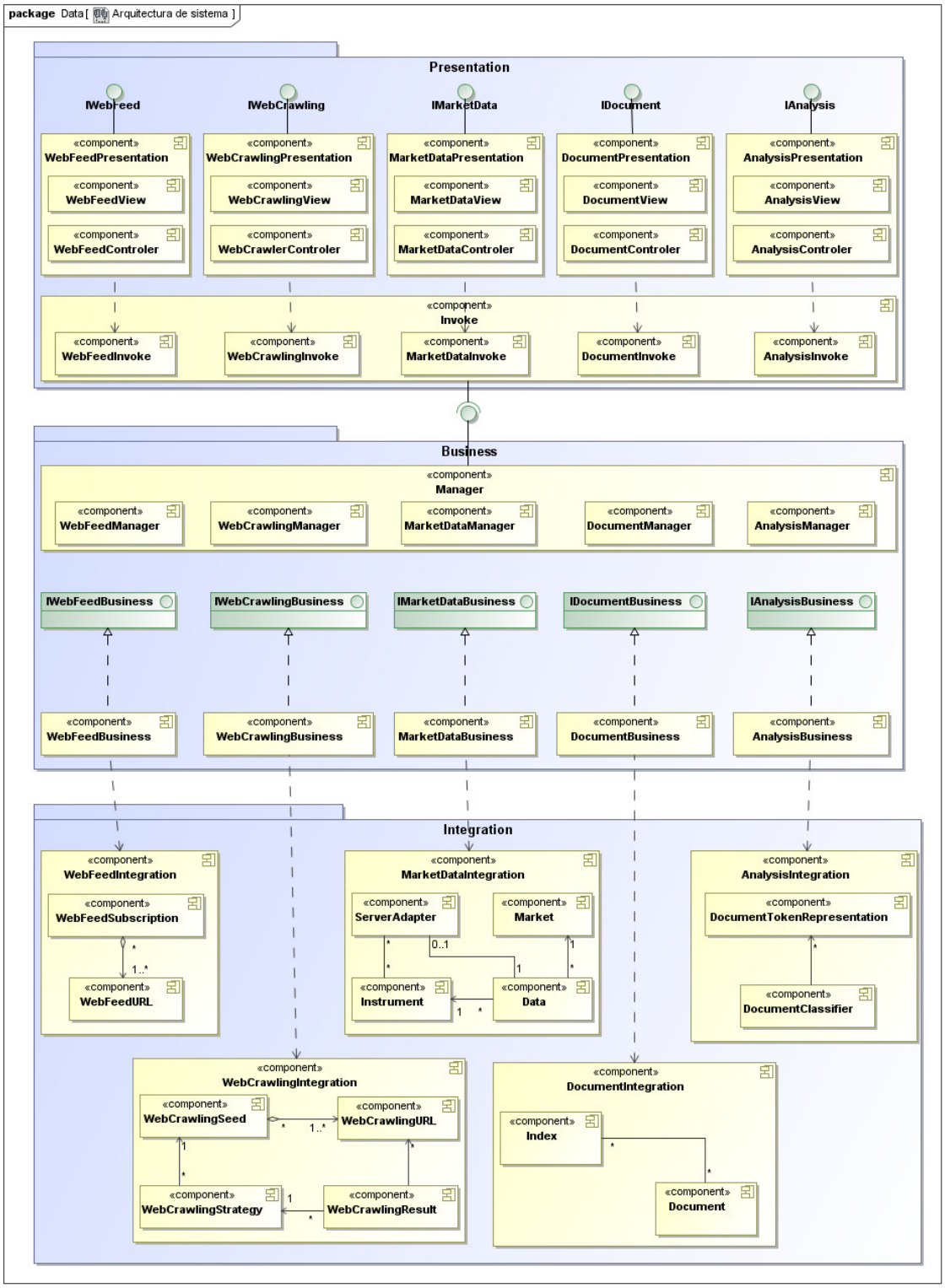


Figura 41. Diagrama del diseño de la arquitectura del sistema detallado a nivel de componentes.

3.3. Diseño de componentes

De cara a detallar el diseño de los componentes del sistema se han tomado en esta fase las siguientes decisiones de diseño que son comunes a la fase de implementación:

1. Las entidades del modelo de dominio de la aplicación estarán desarrolladas en Java como una API independiente de la aplicación con el fin de incrementar la reusabilidad y el desacoplamiento del resto de los componentes del diseño.
2. La interfaz grafica de usuario se desarrollará en Oracle FXML un lenguaje basado en XML para la definición de interfaces gráficas en JavaFX [7].

Obtención de contenidos:

- ***Subscripción de contenidos RSS y Atom***

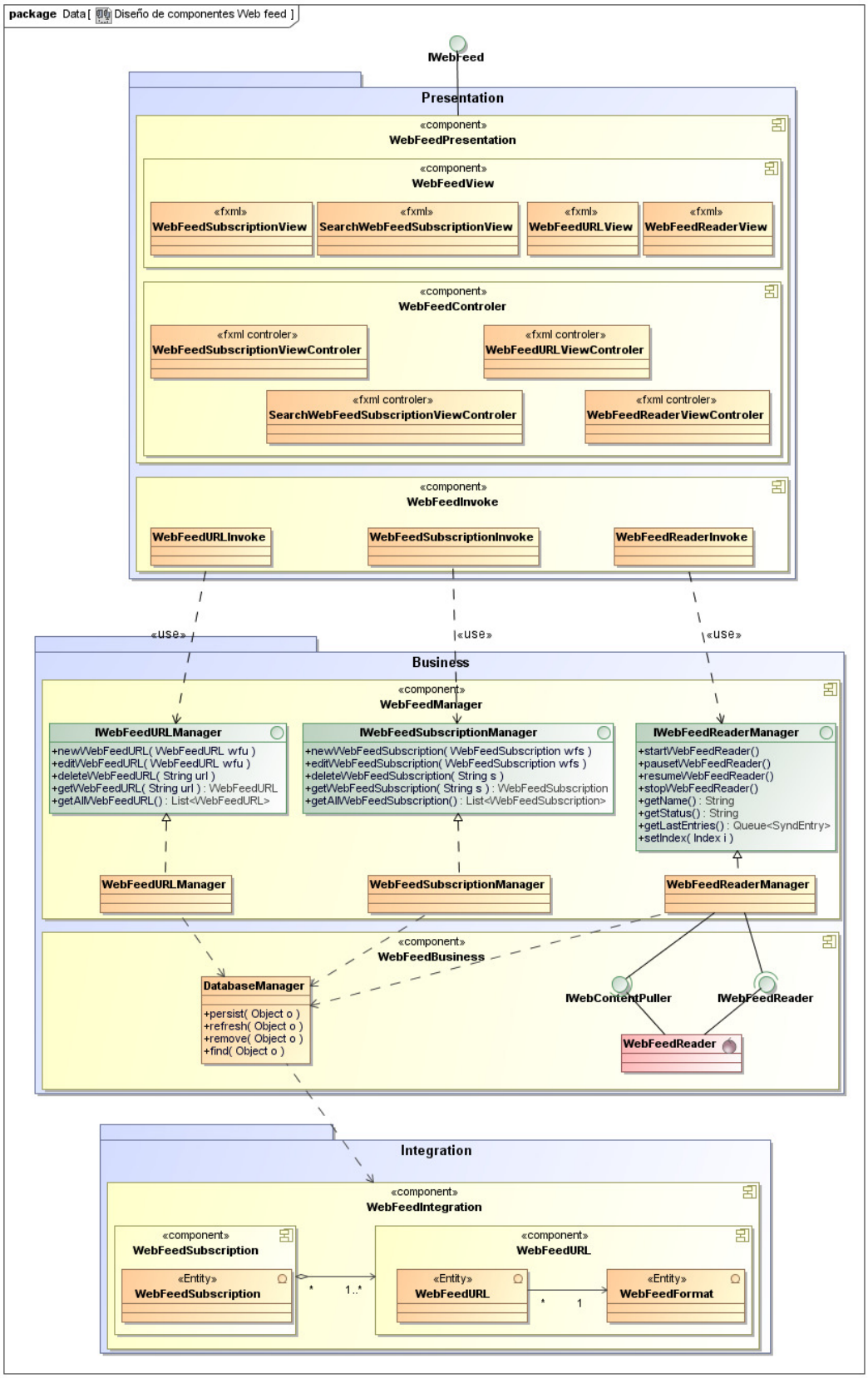


Figura 42. Diagrama de diseño de componentes asociado a la subscripción de contenidos RSS y Atom.

- ***Rastreado de contenidos:***

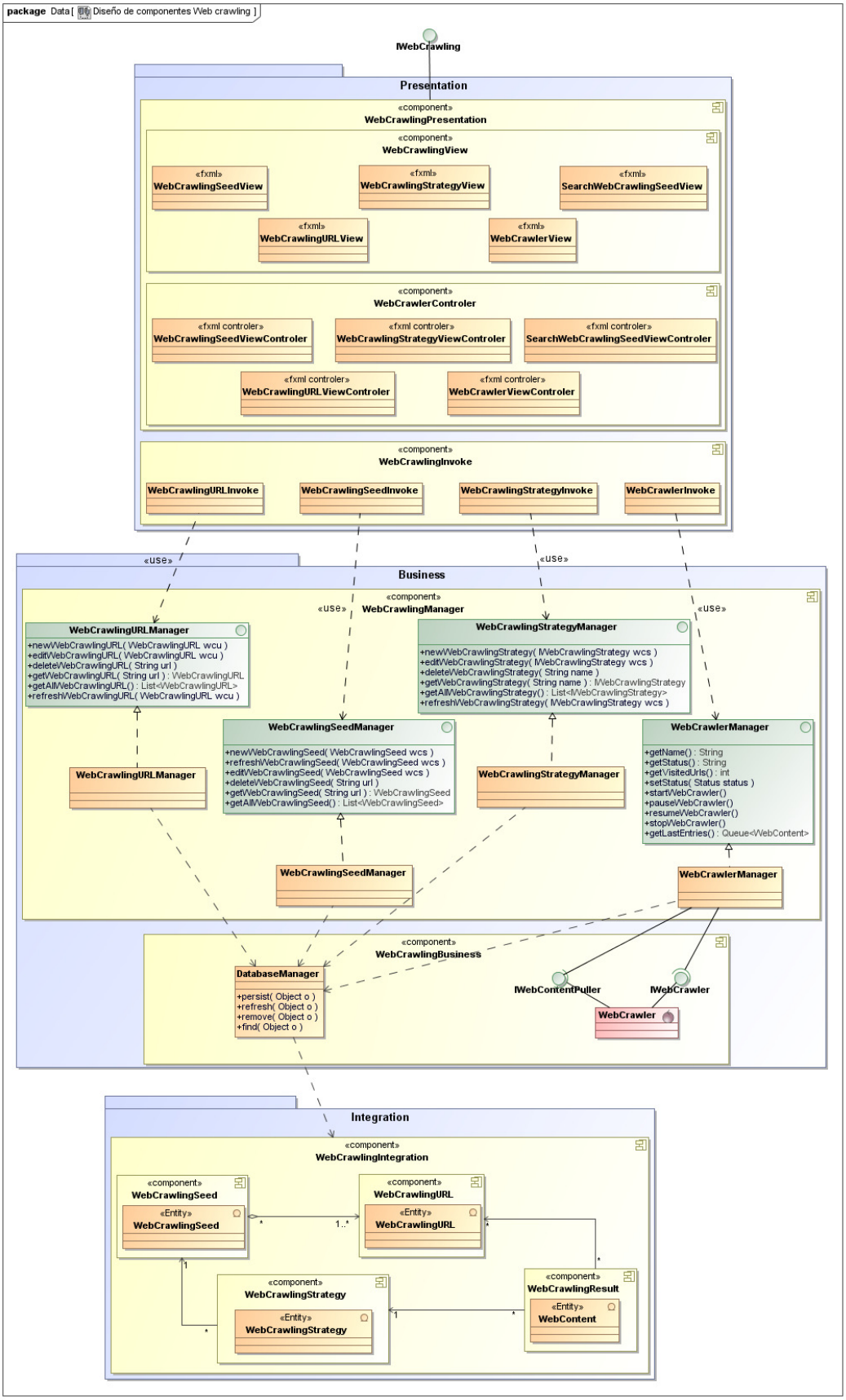


Figura 43. Diagrama de diseño de componentes asociado al rastreo de contenidos en Internet (crawling).

Gestión de documentos

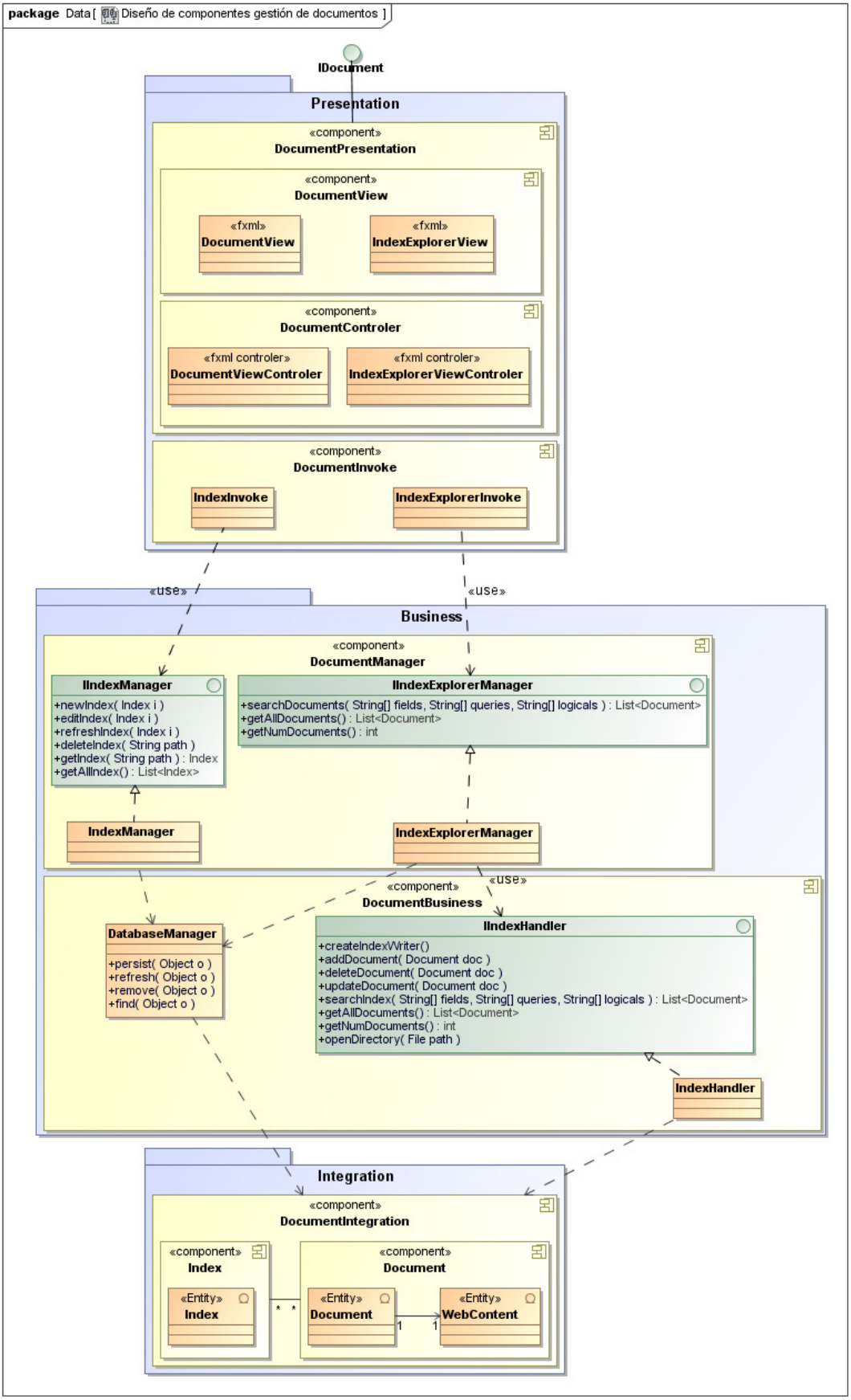


Figura 44. Diagrama de diseño de componentes asociado a la gestión de documentos.

Datos de mercado

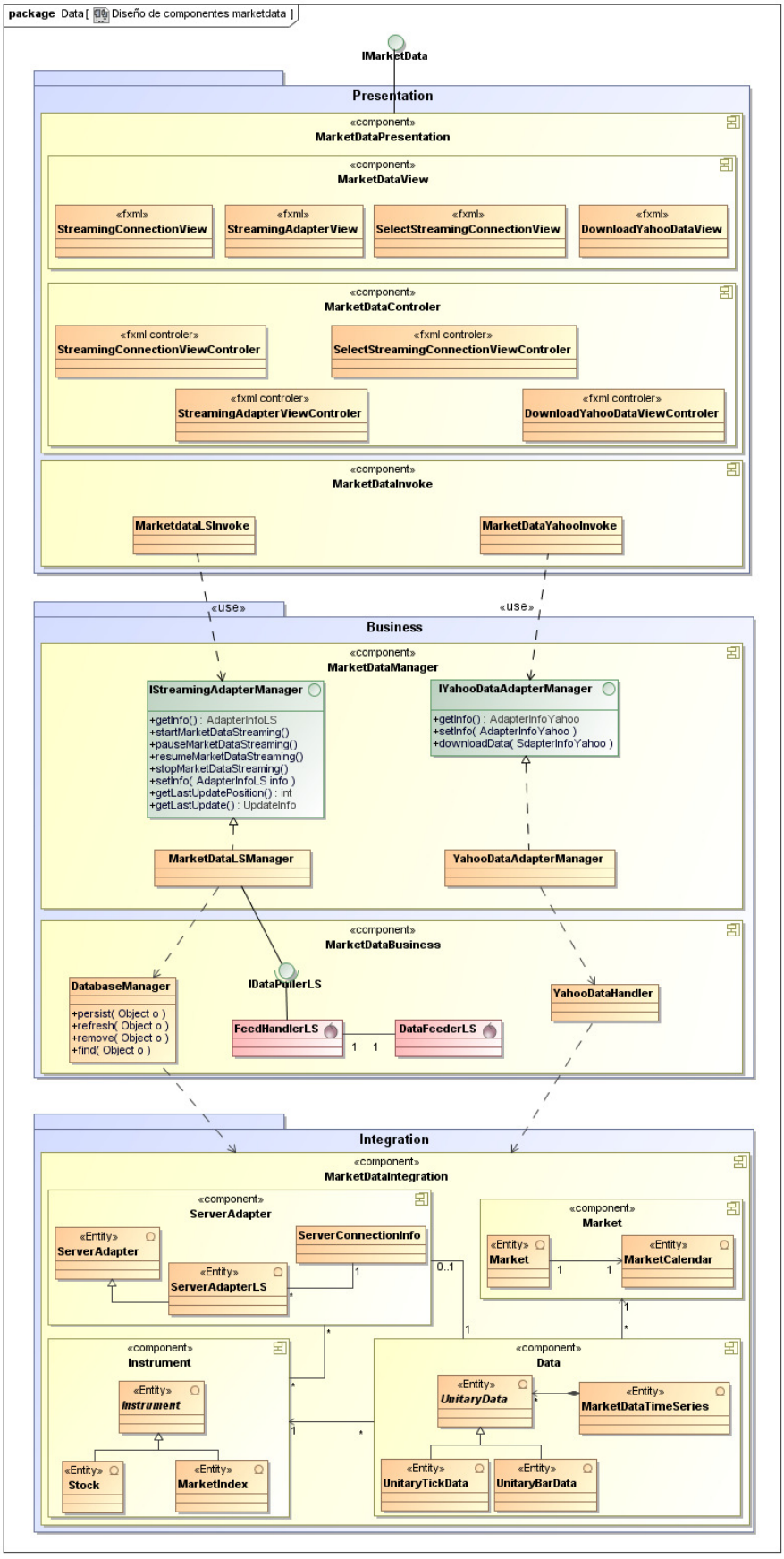


Figura 45. Diagrama de diseño de componentes asociado a la obtención de datos de mercado.

Análisis de documentos:

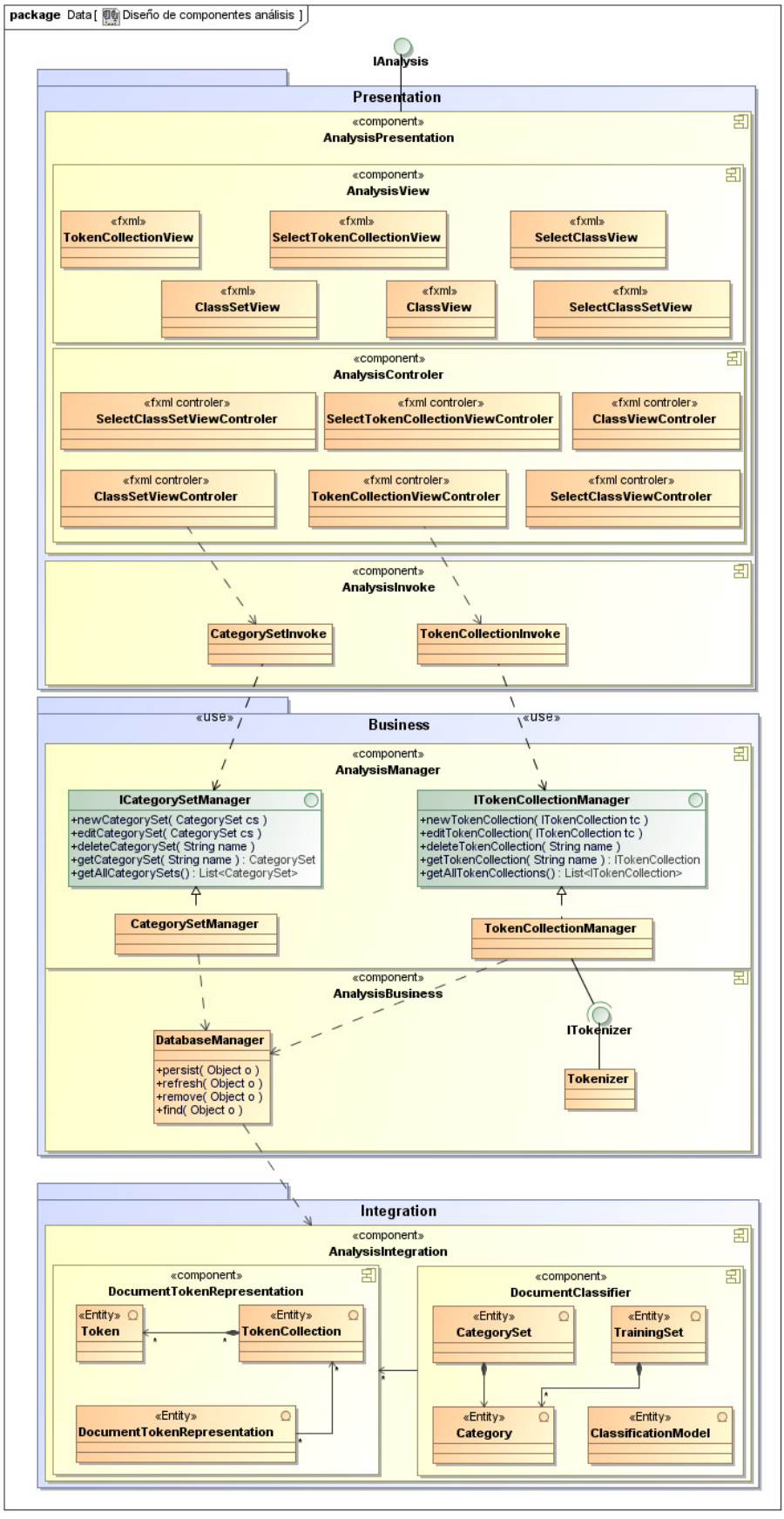


Figura 46. Diagrama de diseño de componentes asociado al análisis de documentos.

3.4. Diseño de la interfaz de usuario

El lanzamiento de la aplicación presenta a usuario un vista consistente en una barra de menús desde donde se tiene acceso a las funcionalidades del sistema.

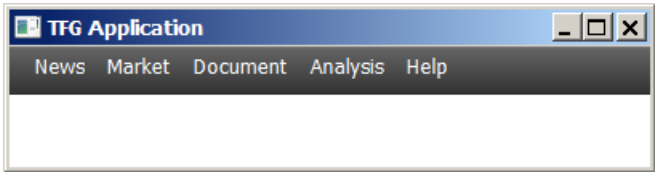


Figura 47. Vista de la pantalla principal al inicio de la aplicación.

Obtención de contenidos

- ***Subscripción de contenidos RSS y Atom***

Mediante la siguiente vista la aplicación permite la creación o modificación de una subscripción de contenidos. Los campos requeridos son un nombre, una descripción de la subscripción y una colección de URLs que pueden ser seleccionadas desde la base de datos o ser nuevamente creadas por el usuario. Además se selecciona un intervalo de tiempo, a partir de una colección preestablecida, que es el que utilizara la aplicación para refrescar la lectura de las subscripciones.

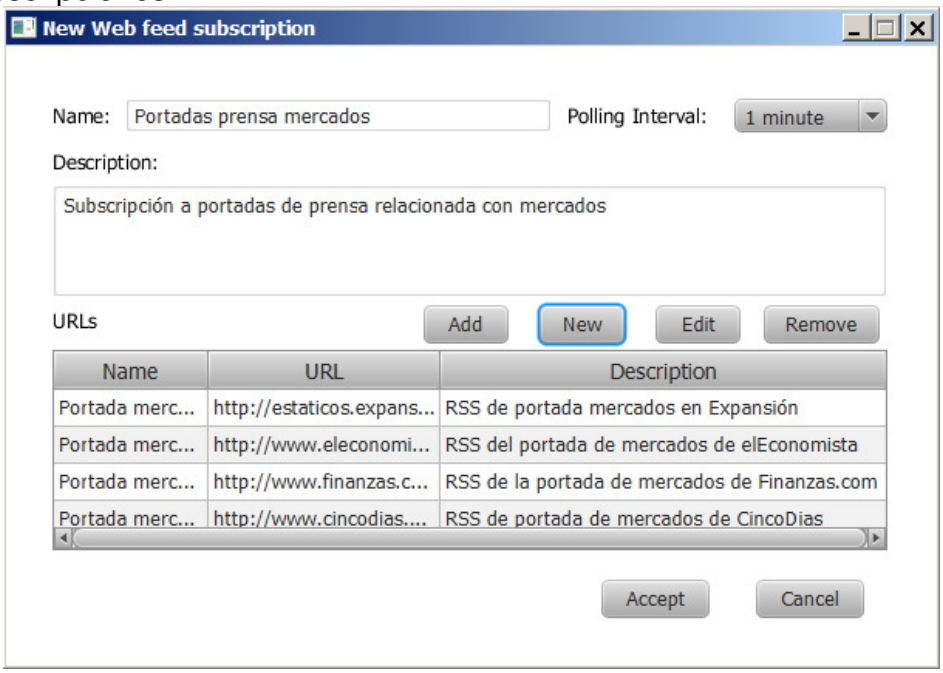


Figura 48. Vista asociada a la creación y modificación de una subscripción de contenidos RSS y Atom.

La creación o modificación de una nueva URL para una suscripción de contenidos, requiere la incorporación de un nombre, una descripción y un formato desde una colección de formatos preestablecida.

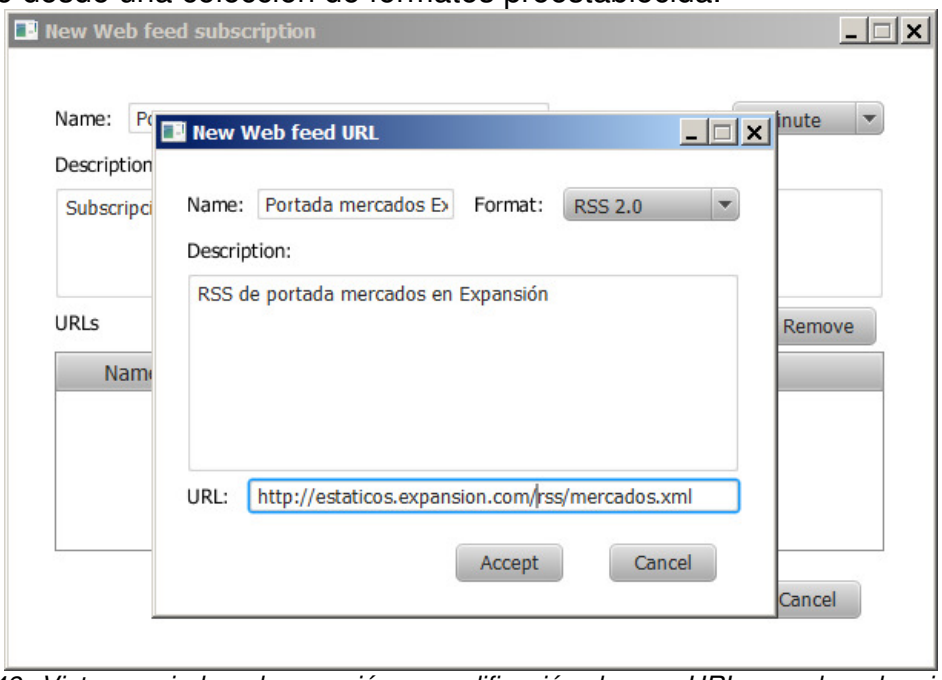


Figura 49. Vista asociada a la creación y modificación de una URL para la subscripción de contenidos RSS/Atom.

El panel siguiente permite la búsqueda de una subscripción en la base de datos. Desde el panel de búsqueda, también es posible la selección de una suscripción y la llamada para que se inicie un lector de contenidos.

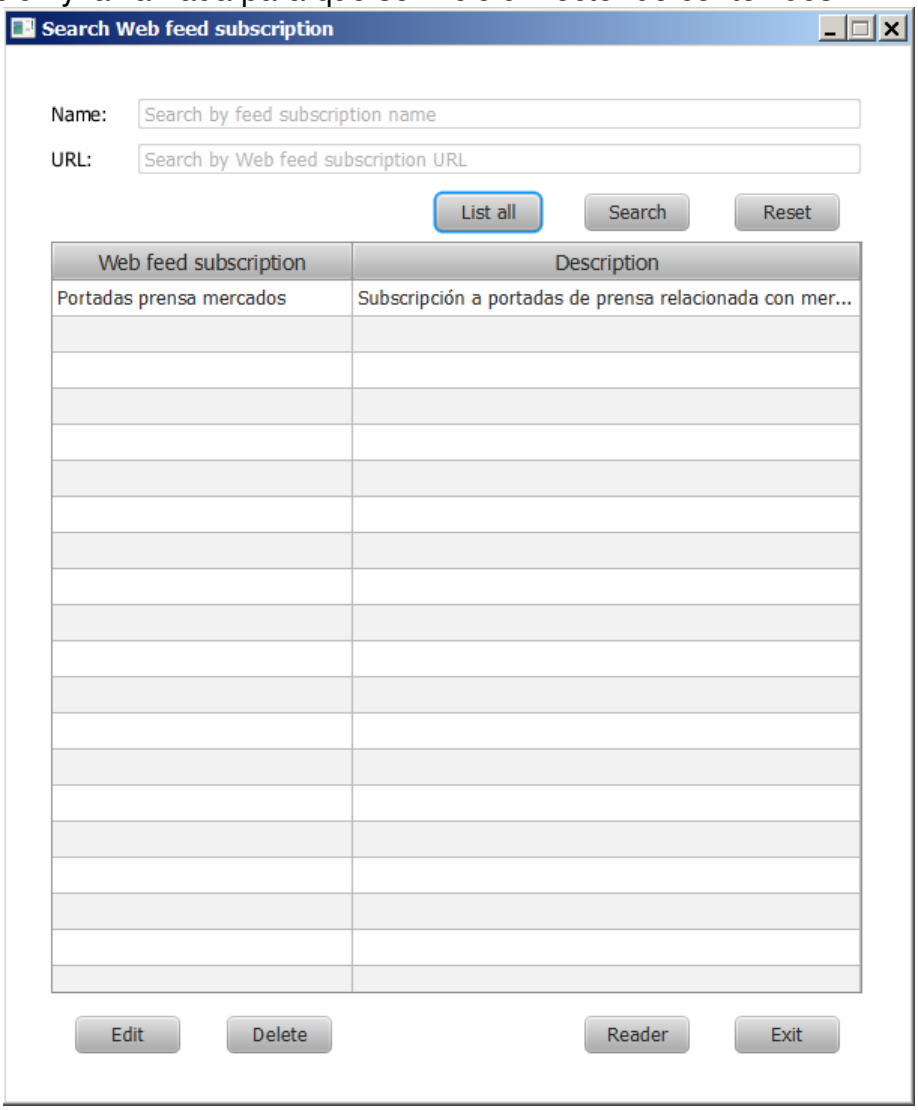


Figura 50. Vista asociada a la búsqueda de una subscripción de contenidos RSS y Atom.

El siguiente panel corresponde a la vista de un lector de contenidos. Tiene cuatro partes principales:

- El encabezamiento donde se puede seleccionar una subscripción de contenidos y un índice para la persistencia y el indexado de los documentos. Además incorpora los controles para el inicio, pausa, reinicio y parada de las lecturas.
- La tabla izquierda corresponde a las subscripciones abiertas donde se muestra el nombre y el estado de la subscripción.
- La tabla derecha muestra la actualización de las lecturas de la subscripción seleccionada.
- Finalmente el panel inferior corresponde a un interprete XML/HTML para la visualización de los enlaces.

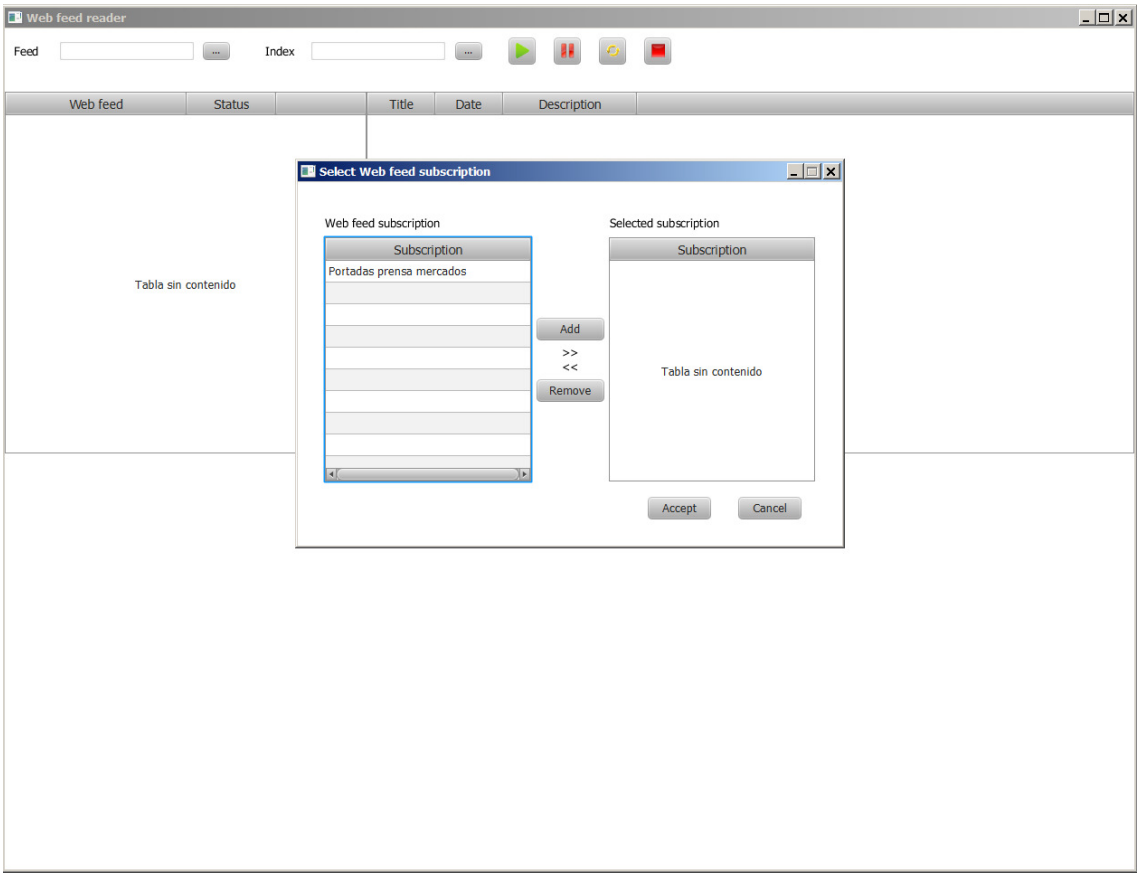


Figura 51. Vista asociada a la selección de una subscripción tras a la apertura de un lector de subscripciones de contenidos RSS y Atom.

En este panel se ve un subscripción abierta (tabla izquierda), con varios contenidos asociados (tabla derecha). En la parte inferior se visualiza el contenido seleccionado.

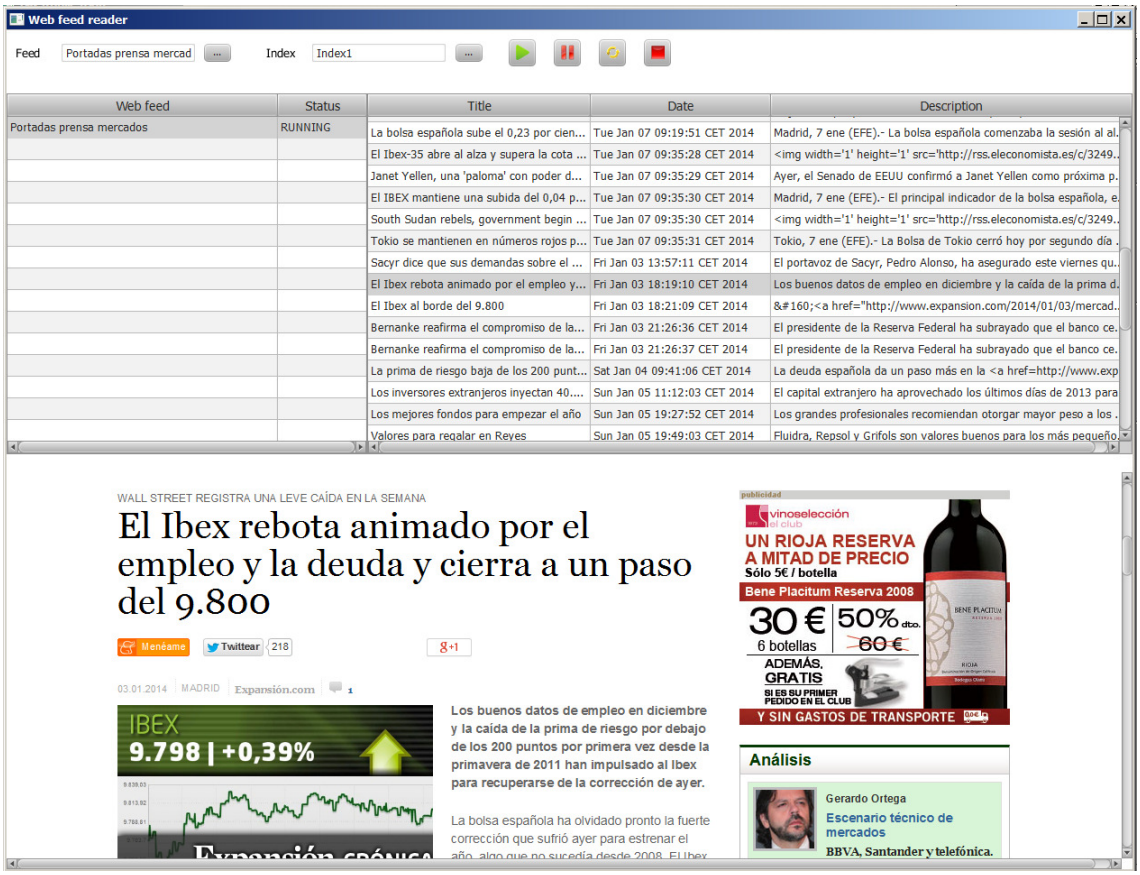


Figura 52. Vista asociada a un lector de subscripciones de contenidos RSS y Atom que está recibiendo la entrada de contenidos.

- ***Rastreado de contenidos***

Mediante la siguiente vista, la aplicación permite la creación o modificación de una semilla para el rastreo de contenidos. Los campos requeridos son un nombre, una descripción de la semilla y una colección de URLs que pueden ser seleccionadas desde la base de datos o ser nuevamente creadas por el usuario.

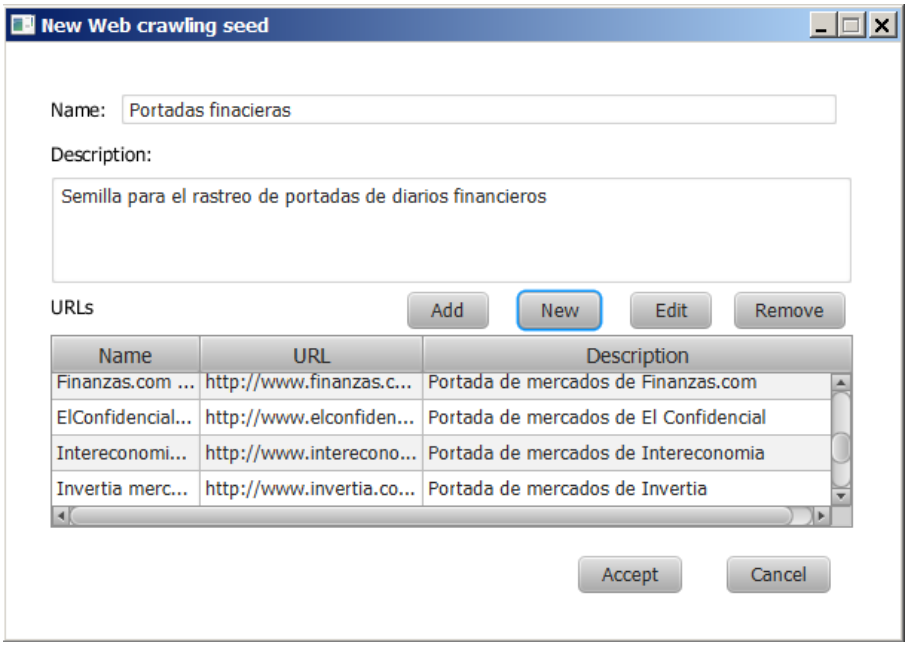


Figura 53. Vista asociada a la creación o modificación de una semilla para el rastreo de contenidos en Internet (crawling).

Por su parte la creación o modificación de una nueva URL para el rastreo de contenidos, requiere la incorporación de un nombre, una descripción y una URL utilizando la siguiente vista.

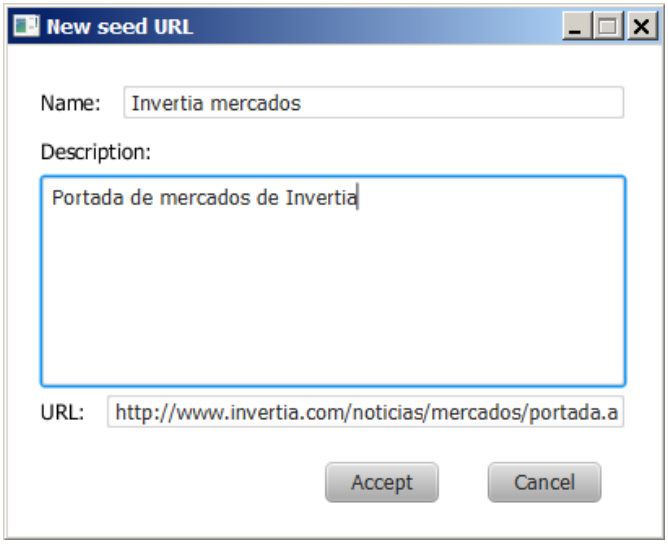


Figura 54. Vista asociada a la creación o modificación de una URL para el rastreo de contenidos en Internet (crawling).

El panel de búsqueda de una semilla permite la búsqueda por nombre o URLs. Desde el panel de búsqueda también es posible la selección de una semilla y la invocación para que se abra el panel de un nuevo rastreador de contenidos.

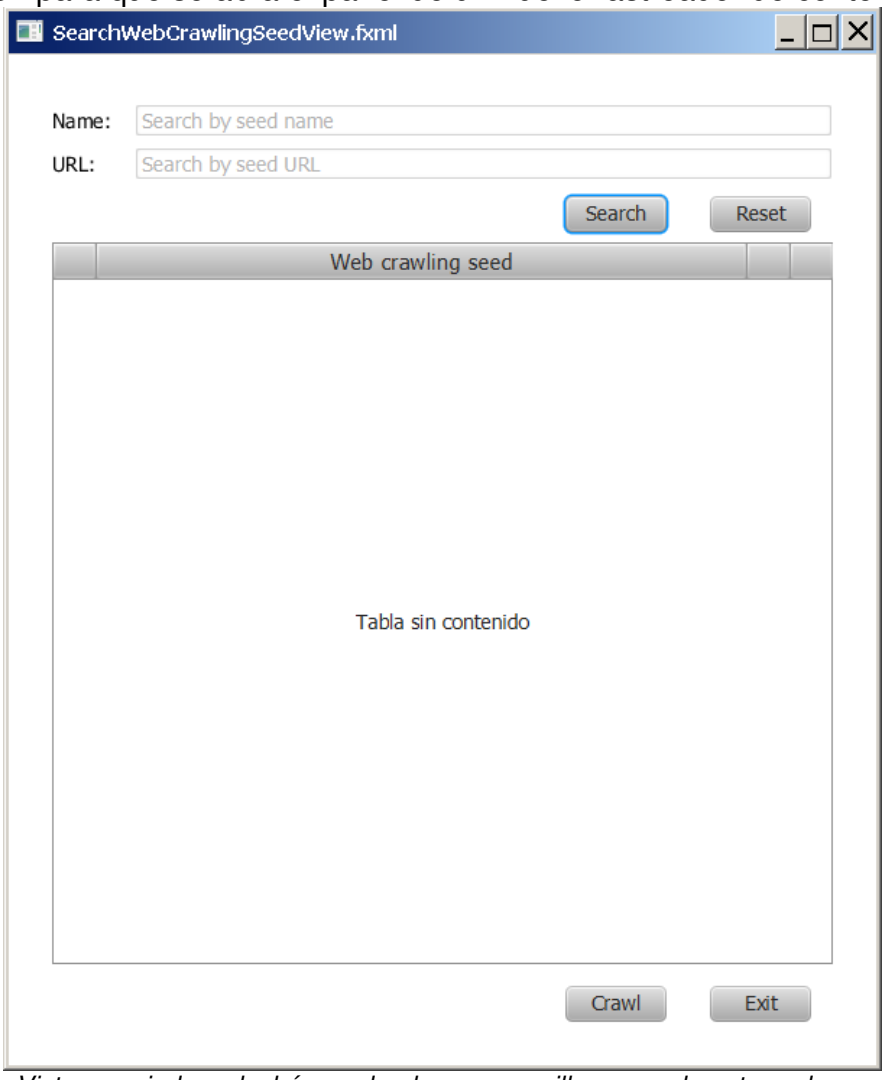


Figura 55. Vista asociada a la búsqueda de una semilla para el rastreo de contenidos en Internet (crawling).

Un rastreador de contenidos utiliza una estrategia de rastreo, que puede ser creada y modificada utilizando el siguiente panel. Una estrategia de rastreo se puede crear con los parámetros: nombre de estrategia, descripción de estrategia, número máximo de URLs visitadas, máxima profundidad de rastreo, retraso entre URLs visitadas, la selección de una semilla y el algoritmo de rastreo. Además la estrategia puede incorporar un criterio para medir la relevancia de los lugares rastreados.

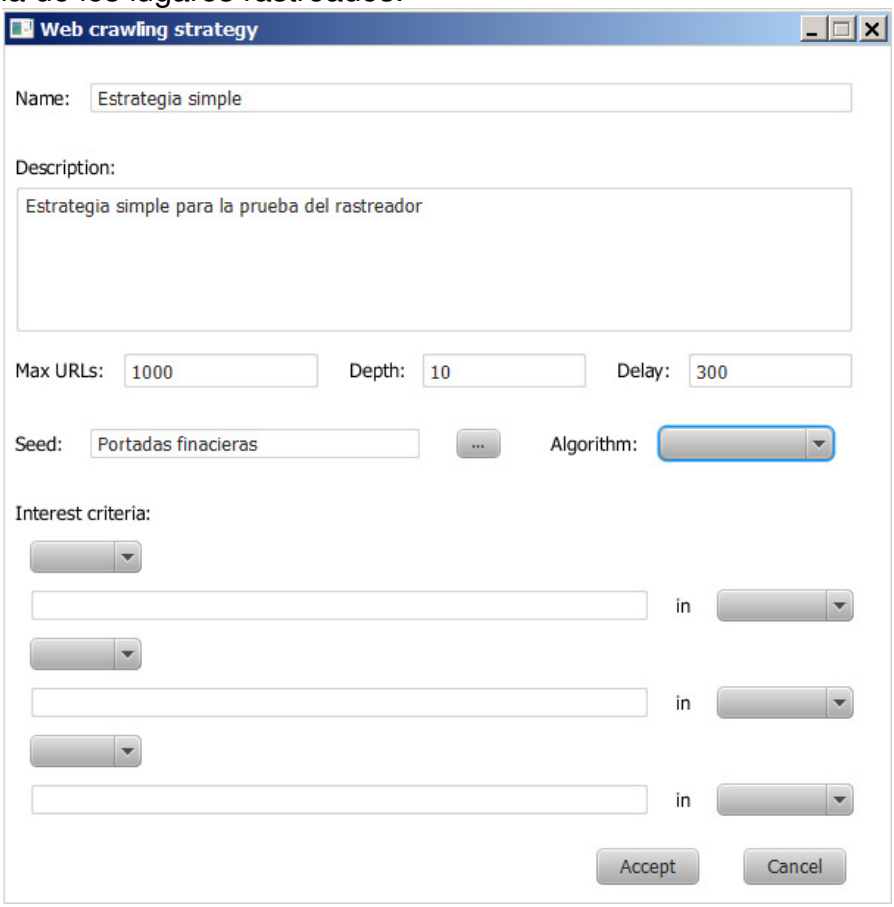


Figura 56. Vista asociada a la creación y modificación de una estrategia de rastreo de contenidos en Internet (crawling).

El siguiente panel muestra la vista de un rastreador de contenidos con las siguientes partes principales:

- El encabezamiento donde se puede seleccionar una estrategia de rastreo y el índice para el indexado y la persistencia de los documentos obtenidos. Además los controles para iniciar, parar, reiniciar o finalizar el rastreo.
- En la parte inferior un primer panel donde se muestran los parámetros de la estrategia seleccionada y desde donde se pueden modificar.
- Un segundo panel con la consola del rastreador. Que se describe a continuación.

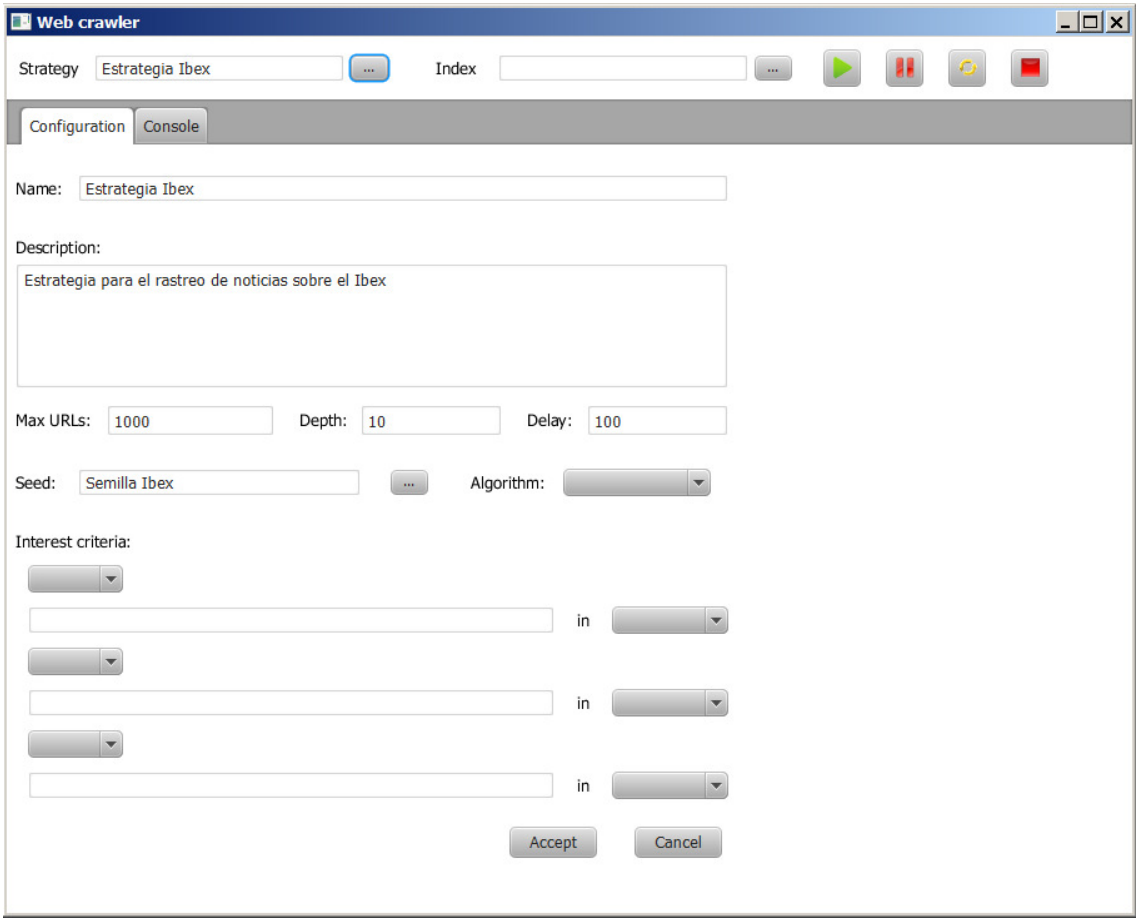


Figura 57. Vista asociada a la configuración de un rastreador de contenidos en Internet (crawling).

La consola del rastreador muestra una tabla superior con los datos de los contenidos encontrados y un panel inferior se puede visualizar el contenido seleccionado.

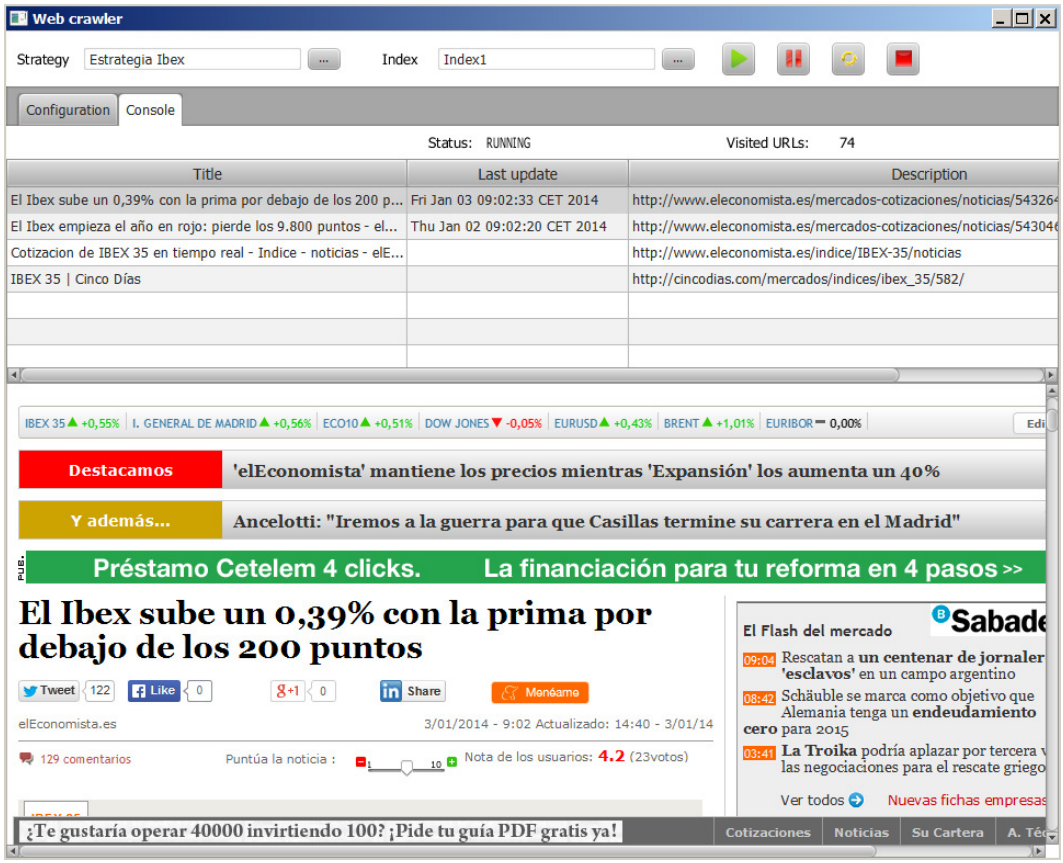


Figura 58. Vista asociada a la consola de un rastreador de contenidos en Internet (crawling).

Gestión de documentos

La aplicación lleva a cabo la persistencia de documentos utilizando el sistema de Indexado de Apache Lucene [8]. La localización de un índice se representa únicamente con un directorio en el sistema de archivos. La aplicación permite a usuario la creación de índices con un nombre, una descripción y una ruta de acceso. La estructura de archivos que presenta un índice se muestra más adelante en la sección 4.2 de la memoria.

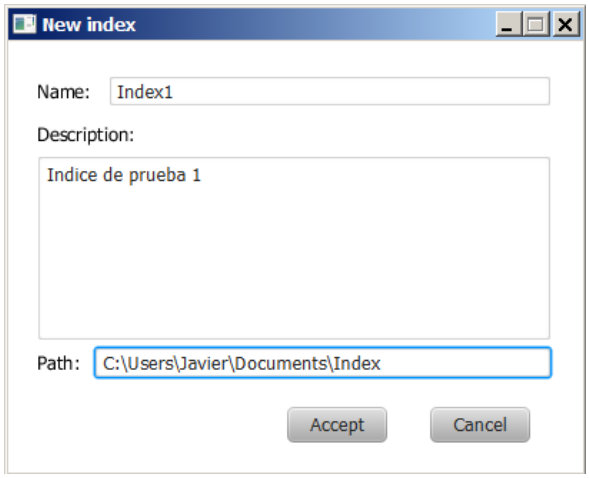


Figura 59. Vista asociada al panel para la creación de un índice de documentos.

La aplicación utiliza un explorador de índices para la gestión y visualización de los documentos indexados dentro de un índice. Mediante la selección de un índice se pueden cargar los documentos asociados con este.

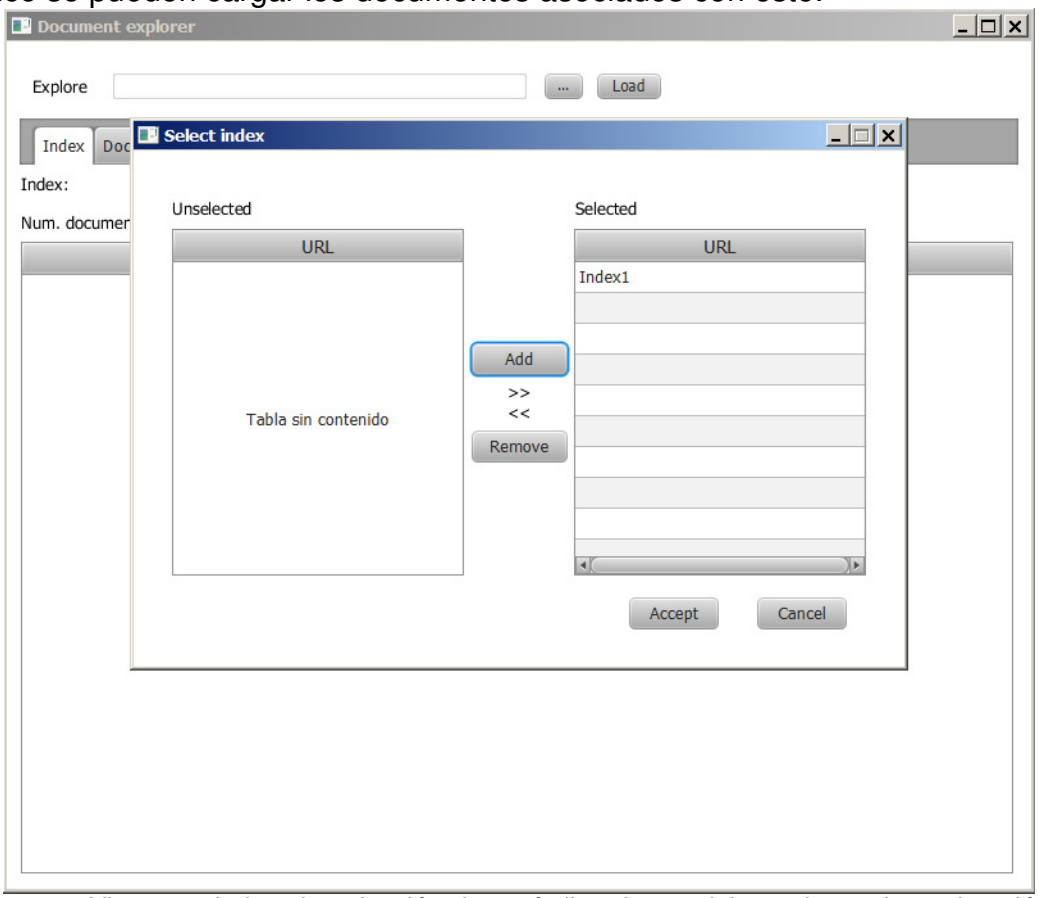


Figura 60. Vista asociada a la selección de un índice dentro del panel para la exploración de índices de documentos.

El explorador se estructura en cuatro paneles.

- Un panel dedicado al índice y el litado de sus documentos asociados.
- Un segundo panel en el que se puede visualizar un documento seleccionado.
- Un herramienta de búsqueda de documentos.
- Un panel para la creación de colecciones de entrenamiento para la clasificación de documentos.

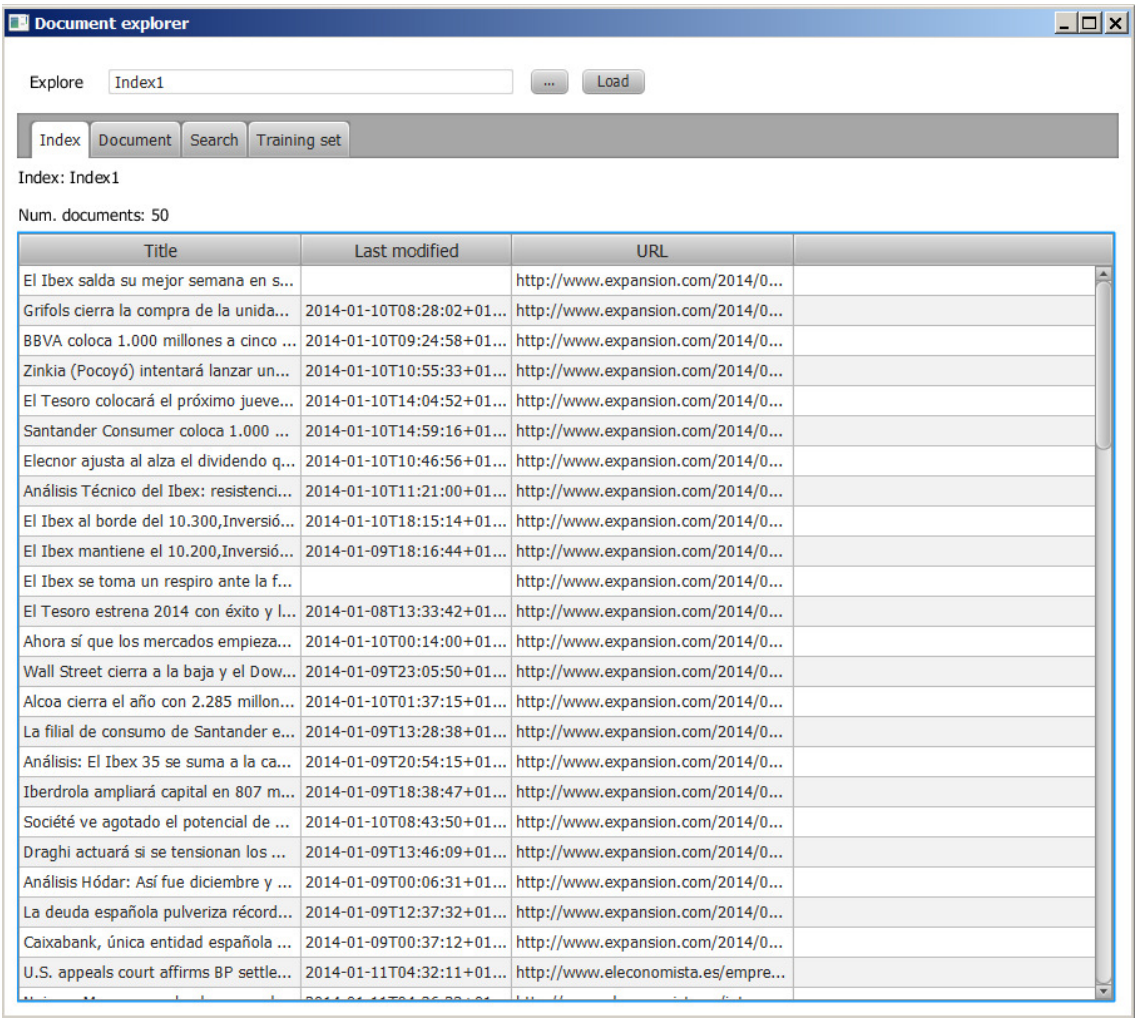


Figura 61. Vista asociada a la visualización de los documentos indexados en un índice.

A partir de la selección de un documento dentro de un índice el explorador permite la visualización del documento en formato tabular, mostrando los campos y los valores para cada campo; además de permitir la visualización del documento original en el panel inferior.

Figura 62. Vista asociada a la visualización de un documento seleccionado, en formato tabular así como a la visualización del contenido Web original.

El explorador además permite la posibilidad de desarrollar estrategias de búsqueda de documentos dentro de un índice. En este caso la selección de un documento también permite su visualización en formato tabular en el panel de documento.

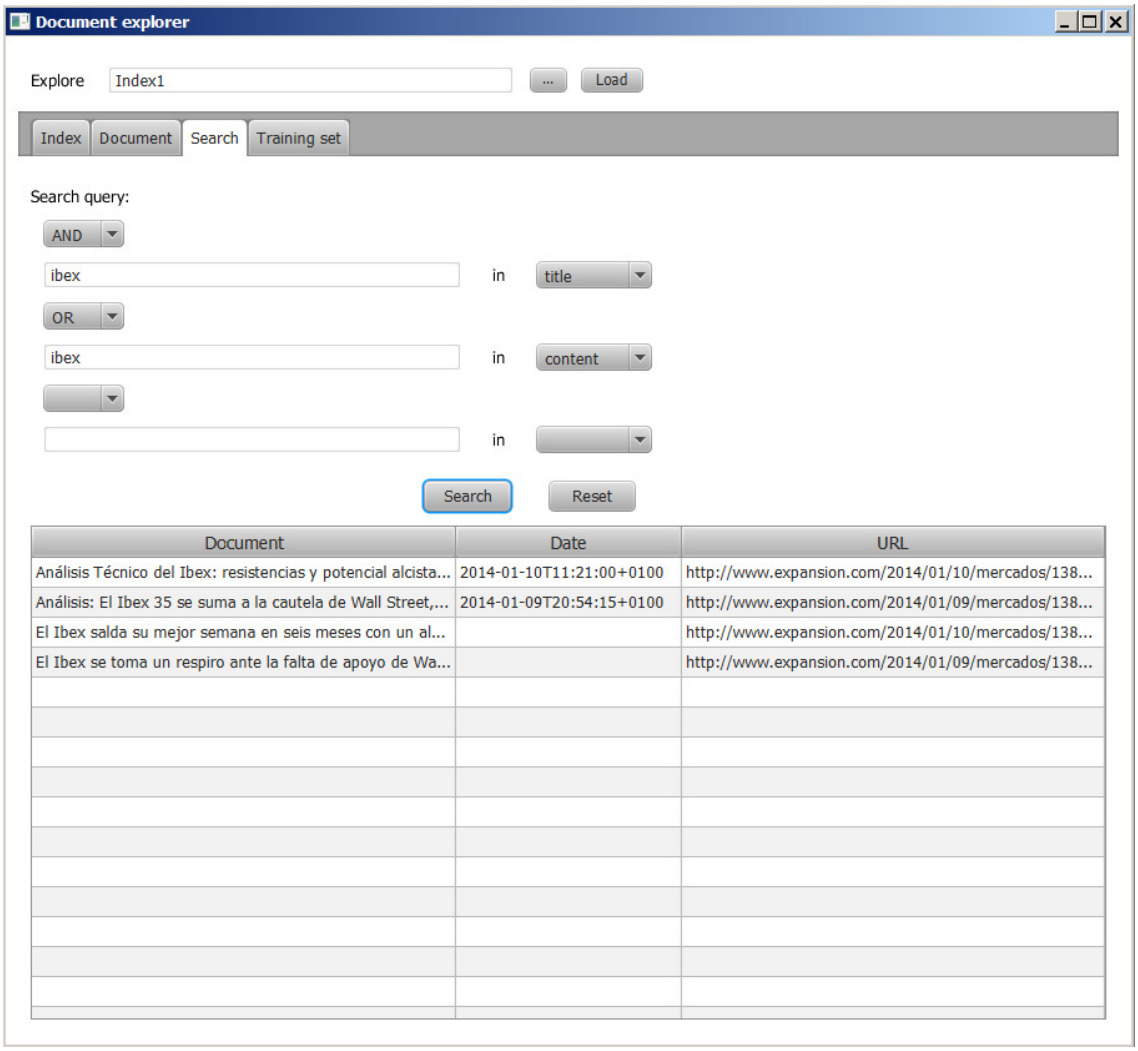


Figura 63. Vista asociada a la búsqueda de documentos dentro de un índice.

Datos de mercado

La aplicación permite al acceso de datos de mercado en modo *streaming*. Para ello es necesario la utilización de una conexión mediante la siguiente vista, en la que hay que proporcionar un nombre de conexión, un servidor de conexión, una descripción de conexión así como la selección de los símbolos de valores financieros disponibles en el servidor.

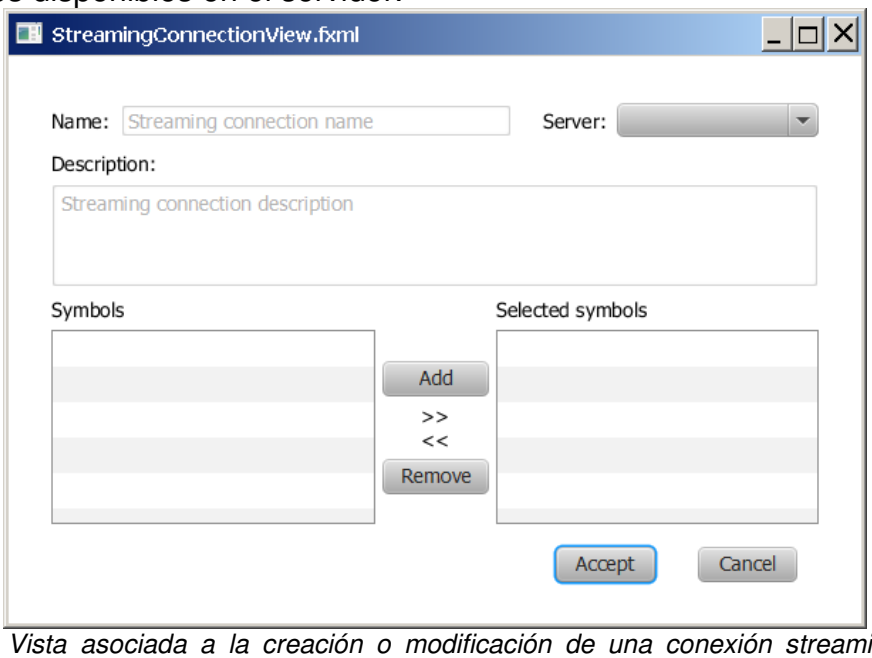


Figura 64. Vista asociada a la creación o modificación de una conexión streaming para la obtención de datos de mercado.

Un adaptador de conexión permite la selección de una conexión en la base de datos e iniciar, pausar, reiniciar o parar la recepción de datos de mercado. Los datos recibidos son monitorizados en formato tabular.

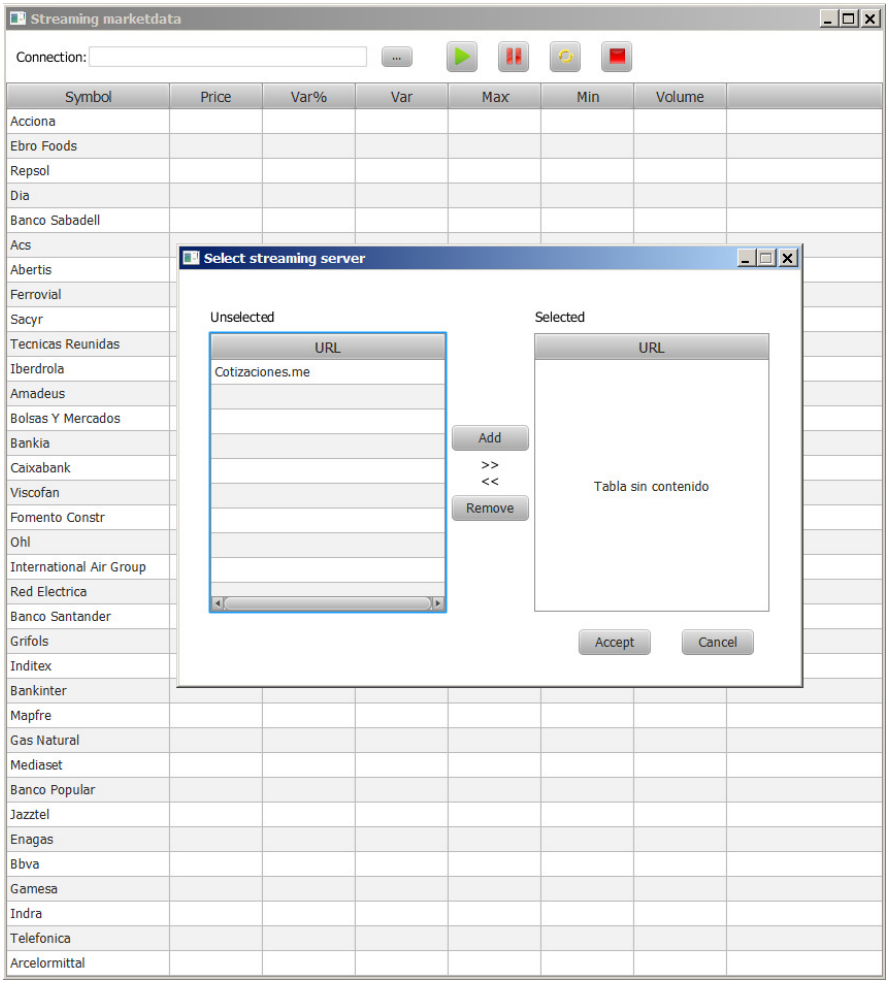


Figura 65. Vista asociada a la selección de una conexión para ser utilizada por una adaptador de obtención de datos de mercado.

Figura 66. Vista asociada a un adaptador para la recepción de datos de mercado.

Además de la obtención de datos en modo *streaming* la aplicación también permite la obtención de datos históricos desde el servidor de Yahoo. Para ello el panel solicita la fecha de inicio, la fecha final, el tipo de datos requeridos y el símbolo del valor financiero a descargar.

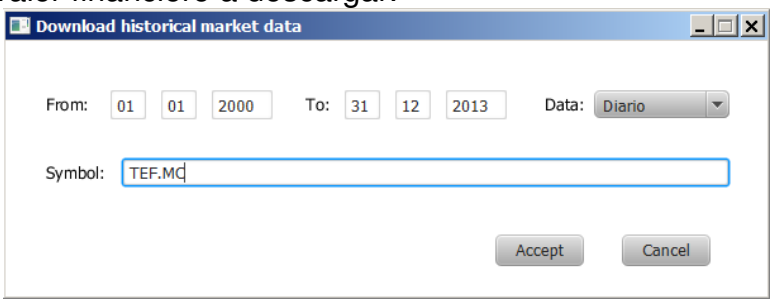


Figura 67. Vista asociada a la descarga de datos de mercado históricos de Yahoo.

Análisis de documentos

La clasificación de documentos implica la utilización de clases numéricas discretas preestablecidas del tipo {0,1} o {0,1,2,3}. La aplicación da la posibilidad de crear conjuntos de clases en un rango y asociar atributos descriptivos a cada clase.

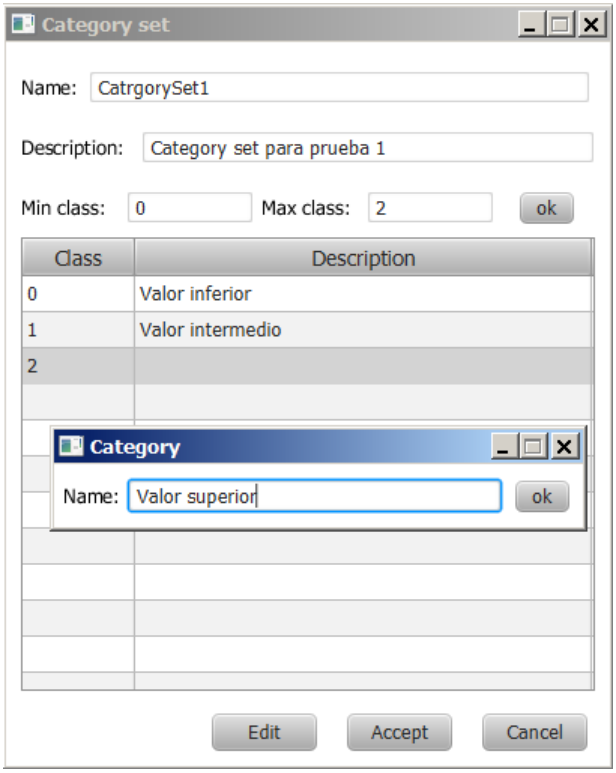


Figura 68. Vista asociada a la creación de un conjunto de clases para la clasificación de documentos.

Con el fin de llevar a cabo representaciones vectorizadas de documentos en base a una colección de términos preestablecida por el usuario, la aplicación permite al usuario crear colecciones de términos utilizando la siguiente pantalla, donde se introduce un nombre una descripción y la colección de términos que el usuario quiere utilizar.

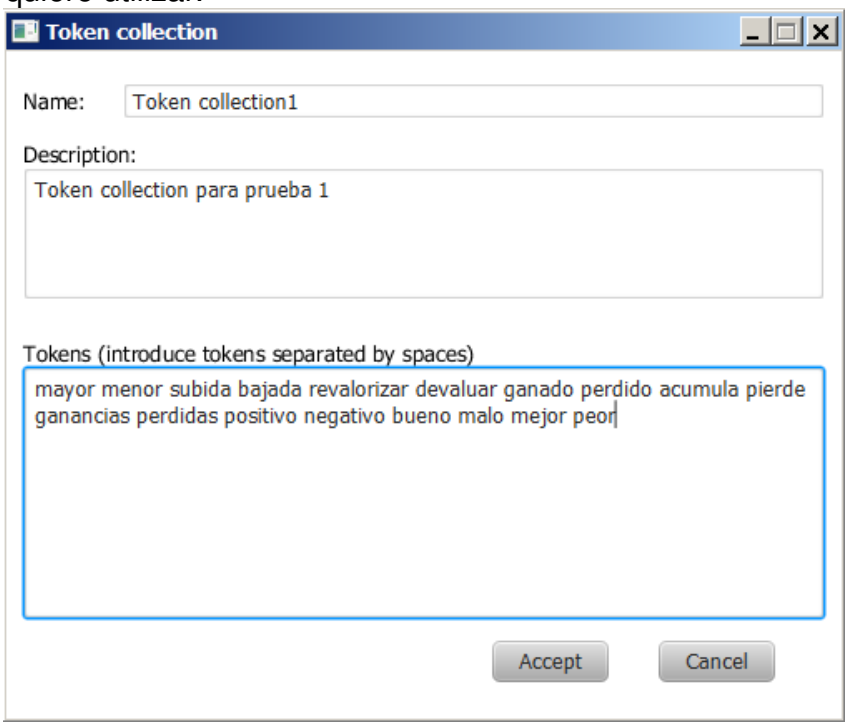


Figura 69. Vista asociada a la creación de una colección de términos (tokens) para la representación vectorizada de documentos.

La creación de la colección implica la tokenización de los términos iniciales introducidos por el usuario tal y como se muestra en la tabla siguiente procedente de la base de datos. Por ejemplo el término *ganancia* es tokenizado como *gananci* de modo que términos como *ganancia* o *ganancias* son equivalentes a efectos de la representación de un documento.

	token [PK] character varyi	collection [PK] character varying(255
1	acumul	Token collection1
2	bajad	Token collection1
3	buen	Token collection1
4	devaluar	Token collection1
5	ganad	Token collection1
6	gananci	Token collection1
7	malo	Token collection1
8	mayor	Token collection1
9	mejor	Token collection1
10	menor	Token collection1
11	negativ	Token collection1
12	peor	Token collection1
13	perdid	Token collection1
14	pierd	Token collection1
15	positiv	Token collection1
16	revalorizar	Token collection1
17	subid	Token collection1

Figura 70. Colección de tokens tal y como resultan del proceso de tokenización.

Mediante el explorador de índices y documentos es posible la utilización de una colección de documentos para construir un *training set* de clasificación. La creación de un *training set* implica elegir el formato vectorizado para la representación de los documentos, el conjunto de clases y por último asociar una clase a cada documento.

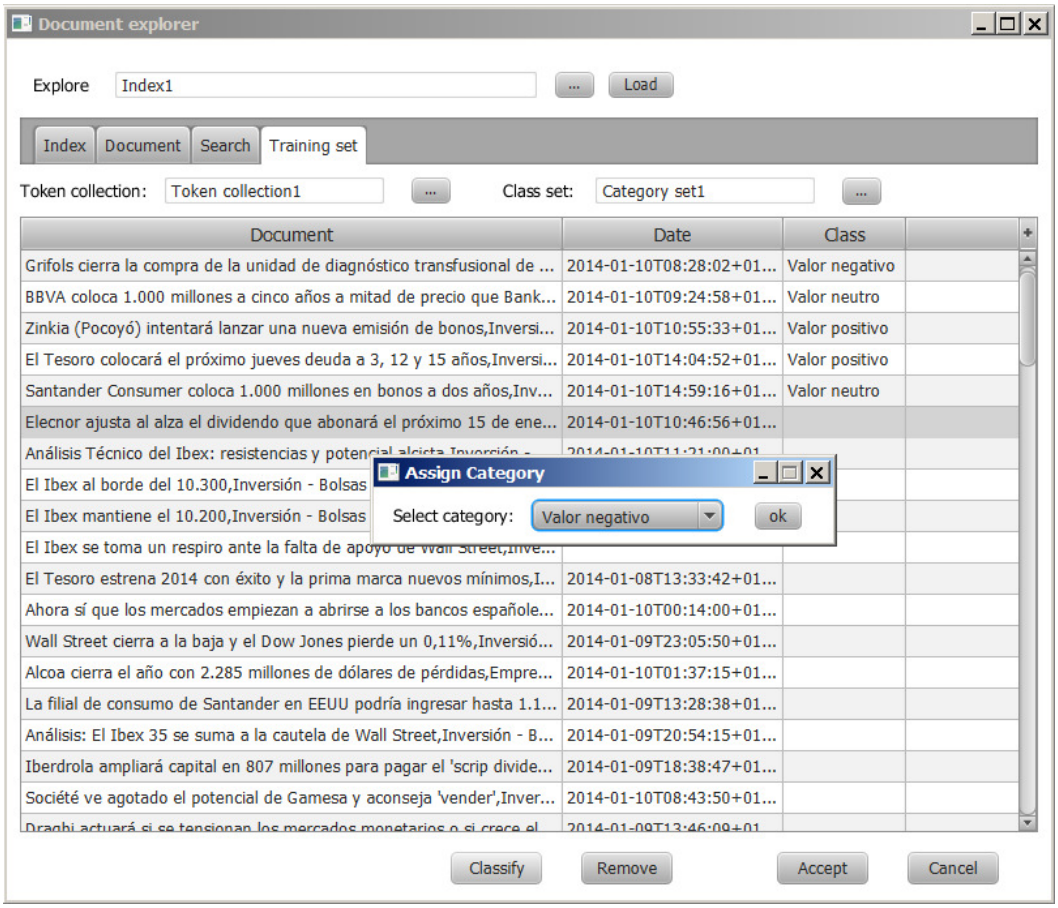


Figura 71. Vista asociada a la creación o modificación de un training set para la clasificación de documentos.

4. Ejecución del plan de trabajo — Producto

El prototipo desarrollado para el proyecto se basa en una aplicación local, aunque como se ha descrito en la etapa de diseño las decisiones tomadas han ido destinadas a dar la suficiente flexibilidad para que la aplicación se pueda extender a una arquitectura distribuida.

4.1. Decisiones de implementación

Las principales decisiones de implementación tomadas han sido:

Entorno de desarrollo: La implementación se ha llevado a cabo en Java utilizando las herramientas de desarrollo “Java Platform (JDK) 7u45” [9].

Desarrollo de la interfaz de usuario: Para el desarrollo de interfaz gráfica se ha utilizado tecnología JavaFX 2 [10]. JavaFX 2 es una evolución de Java para la construcción de entornos gráficos enriquecidos, la cual permite definir la estructura de vistas utilizando un lenguaje propio (FXML) basado en XLM a las cuales se asocian clases controladoras desarrolladas en Java bajo el entorno de desarrollo “JavaFX SDK” [11]. Además el diseño de vistas se ve facilitado por la posibilidad de desarrollarlas utilizando la herramienta “JavaFX Scene Builder” [12].

Persistencias de datos: la aplicación implementa la persistencia de datos utilizando dos aproximaciones:

- Base de datos: por un lado una base de datos relacional, en este caso he optado por la selección de *PostgreSQL* 9.1 [13]. Además la persistencia en la base de datos se ha gestionado mediante EclipseLink-2.5.1 [14] utilizando el estándar de persistencia de la “Java Persistence API (JPA).
- Persistencia de documentos: por otro lado una aproximación basada en sistemas de archivos para la persistencia de documentos indexados de acuerdo a la aproximación estándar de indexado que facilita el entorno “Apache Lucene-4.6.0” [8].

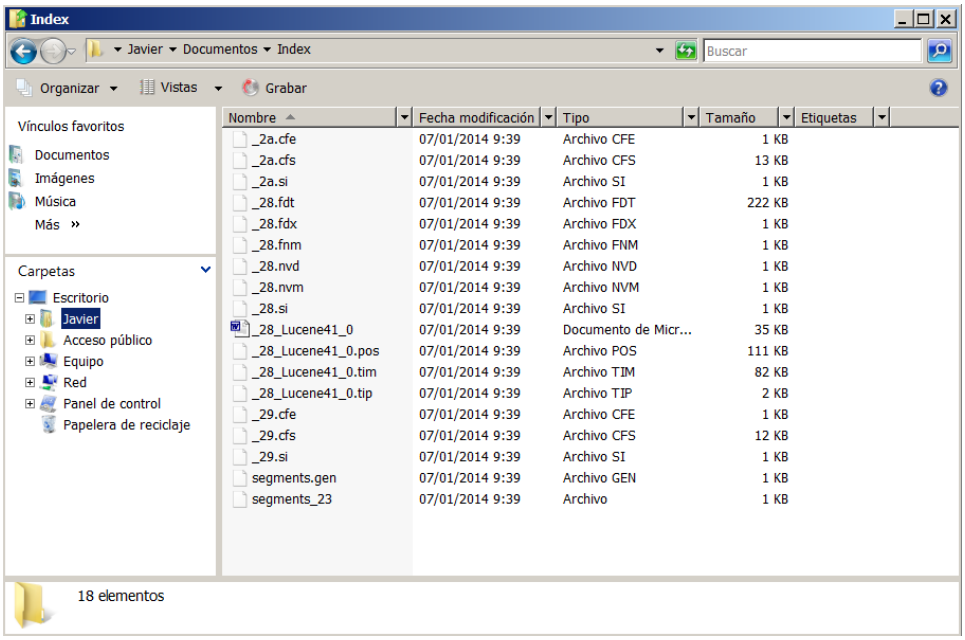


Figura 72. Estructura de archivos asociada a un índice de documentos de Apache Lucene.

Dependencia de librerías externas:

En la siguiente pantalla se muestran las dependencias a librerías externas de la aplicación:

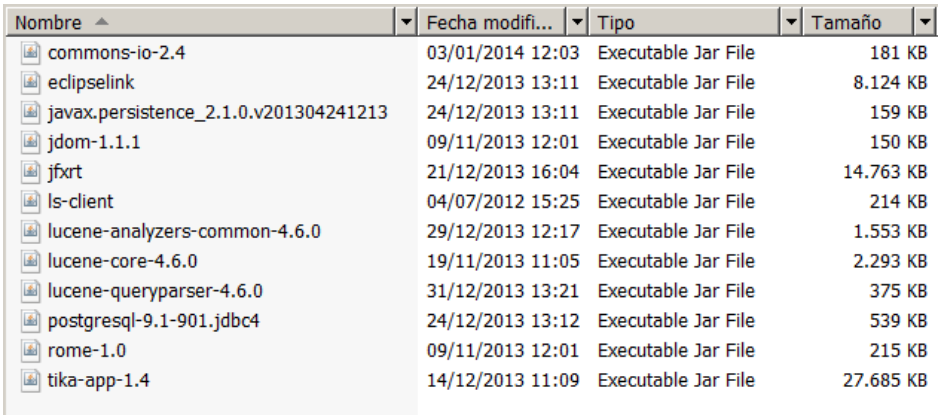


Figura 73. Dependencias a librerías externas utilizadas por la aplicación.

Las no mencionadas hasta ahora se describen brevemente:

- Apache commons-io-2.4: es una librería que provee utilidades para asistir en el desarrollo de funcionalidades de entrada y salida (IO) [16].
- JDOM jdom-1.1.1: es una librería para facilitar el acceso manipulación y salida de datos en formato estándar XML [17].
- ROME rome-1.0: es una librería para proveer las utilidades recesarías para la publicación y lectura de canales de sindicación basados en los protocolos RSS y Atom [18].
- LightStreamer ls-client-2.5.2: es un entorno de aplicaciones que permite el desarrollo de varias formas de mensajería en tiempo real. Es utilizado por algunos servidores que facilitan datos de mercado en tiempo real. La

compañía facilita entre otras la librería “ls-client”, que se utiliza para la construcción de clientes que tengan que comunicarse con servicios LightStreamer [19]. En el caso del proyecto esta se ha utilizado para la implementación de la funcionalidad que da acceso de datos de mercado en streaming.

- Apache Tika tika-app-1.4: La aplicación utiliza la librería Apache Tika para el procesamiento de documentos y la extracción de metadatos y texto extructurado.

4.2. Proyecto Eclipse desarrollado

Para la implementación del prototipo se ha desarrollado un proyecto utilizando el IDE Eclipse. La siguiente figura muestra la estructura de archivos del proyecto:

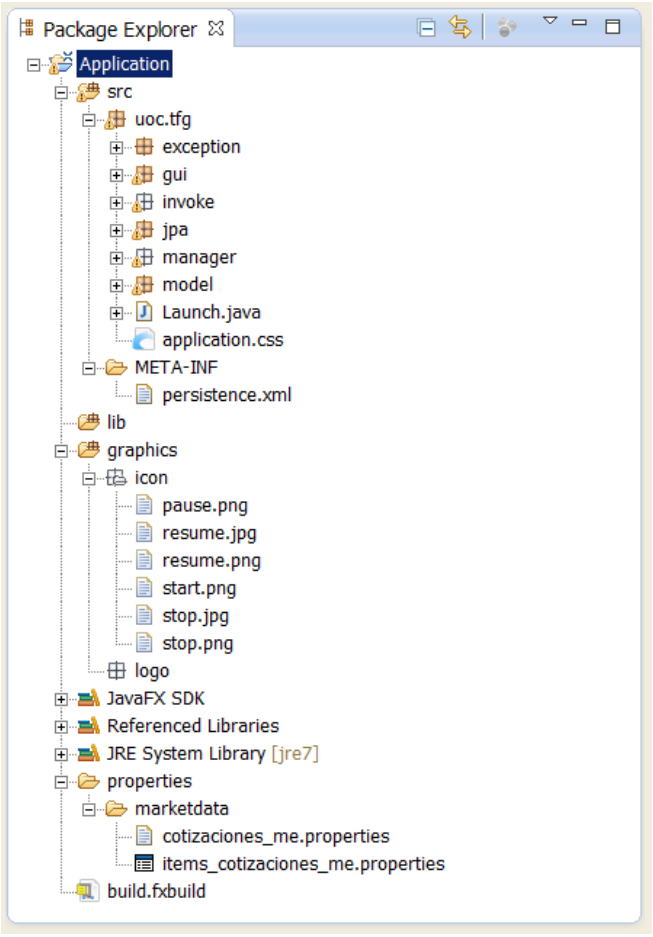


Figura 74. Estructura de archivos del proyecto Eclipse desarrollado para la implementación.

La carpeta de código fuente tiene la estructura de paquetes que se muestra en la siguiente figura. Los paquetes principales se corresponden con las decisiones de diseño de la aplicación:

- **uoc.tfg.gui**: paquete que recoge la interfaz gráfica de usuario. Se corresponde al componente *Presentation*.
- **uoc.tfg.invoke**: su correspondiente en el diseño es el componente *Invoke*.
- **uoc.tfg.manager**: es la capa de software que implementa el componente *Manager* del diseño.
- **uoc.tfg.model**: implementa el modelo de negocio (*Bussines*) de la aplicación.
- **uoc.tfg.jpa**: implementa los componentes de persistencia en la base de datos.

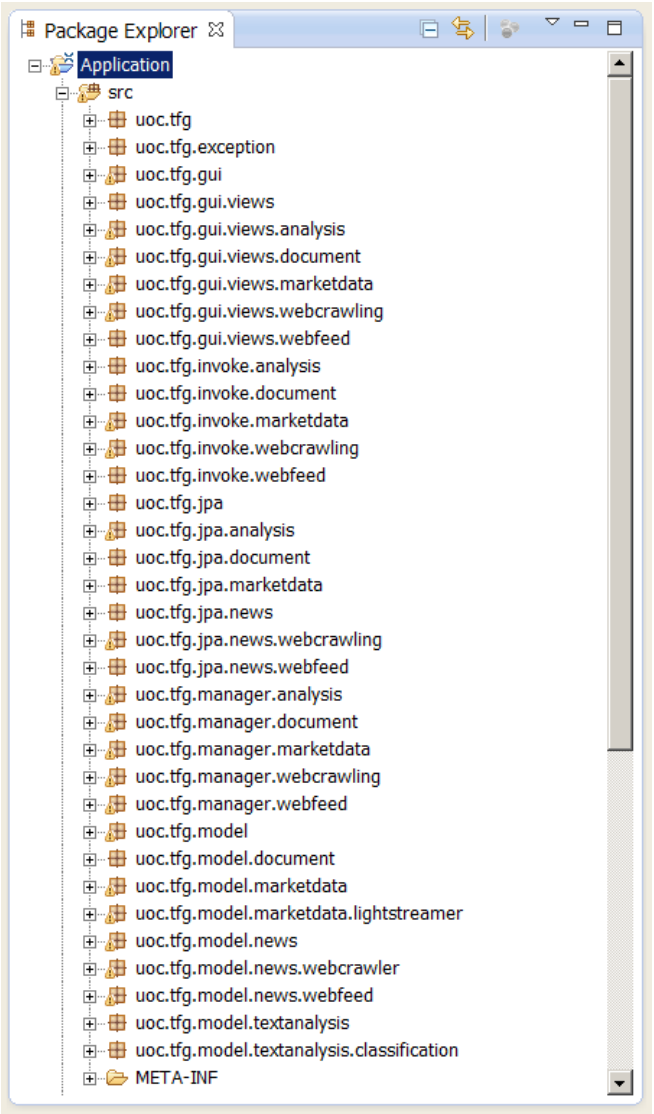


Figura 75. Estructura de paquetes de la carpeta de código fuente del proyecto Eclipse desarrollado para la implementación.

5. Conclusiones

En el proyecto se han completado una gran parte de los objetivos planteados inicialmente. Algunos de los objetivos no se han podido acometer con éxito, aunque la causa ha sido principalmente la limitación de tiempo, quizás por el sobredimensionado inicial de algunas de las tareas planificadas. No obstante es esperable que se finalicen en trabajos posteriores. Como principales objetivos alcanzados se destacan los siguientes:

- Se ha acometido con éxito el análisis y diseño del sistema, lo cual ha permitido identificar y diseñar los componentes software necesarios para poder implementar la solución al problema planteado inicialmente.
- La aplicación desarrolla con éxito la automatización en la obtención y persistencia de contenidos de Internet. En este sentido cabe destacar que se ha podido integrar con éxito las utilidades provistas por ROME para suscripción a contenidos utilizando los protocolos *RSS* y *Atom*, y por Apache Tika para el análisis y extracción estructurada de contenidos Web.
- La aplicación desarrolla con éxito la adquisición de datos de mercado, y ha incorporado con éxito las funcionalidades provistas por *LightStreamer* para el diseño de clientes que se alimentan en *streaming* de sus servidores.
- La aplicación desarrolla las funciones referentes a la gestión de documentos, creación y gestión de índices, indexado y búsqueda de documentos. Se han incorporado con éxito las funcionalidades provistas por entorno Apache Lucene [8].
- La aplicación desarrolla parcialmente las funcionalidades previstas para el subsistema de análisis. Así se han desarrollado las funcionalidades de creación y gestión de grupos de categorías, y de colecciones de términos, así como las funcionalidades de tokenización de documentos y de construcción de colecciones de entrenamiento para clasificación.

El seguimiento de la planificación ha planteado algún problema en los plazos planteados inicialmente. Visto en perspectiva quizá cabe concluir que la magnitud del trabajo planteado ha sido algo superior al que se podía acometer en el periodo de tiempo de un semestre. Esto ha ocasionado que algunos de los objetivos planteados inicialmente no se hayan alcanzado particularmente en lo referente al desarrollo del producto. Entre los objetivos que no se han conseguido destaco los siguientes:

- Inicialmente la implementación se planteó para el desarrollo de la aplicación en forma distribuida, sin embargo la limitación de tiempo me hizo reconsiderar esta posibilidad y optar por el desarrollo del prototipo con una arquitectura local.
- En el tiempo del proyecto no se han podido finalizar la implementación de todos los casos de uso planteados en el análisis. Por ejemplo faltaría

por desarrollar toda la parte referente al desarrollo de la clasificación y agrupamiento de documentos.

- Finalmente no se ha podido llevar a cabo la fase de pruebas de la implementación.

En base a lo comentado anteriormente las líneas de trabajo futuro que se plantean son las siguientes:

- Llevar a cabo la finalización de la implementación de todos los componentes que no han podido ser finalizados.
- Llevar a cabo la fase de pruebas.
- Extender el desarrollo de la aplicación a una arquitectura distribuida que permita ejecutar la parte cliente de forma remota utilizando la tecnología Java Web Start.

6. Glosario

Algoritmos de agrupación: procedimientos no supervisados para la averiguación de la existencia de relaciones dentro de los datos.

Algoritmos de clasificación: procedimientos supervisados para la averiguación de la existencia de relaciones dentro de los datos, en base a la existencia de categorías preestablecidas.

Aprendizaje computacional: rama de la ciencia de la computación que estudia y desarrolla las técnicas que permiten a un computador aprender a partir del análisis de datos.

Atom: concepto que engloba dos estándares para la redifusión de contenidos en la Web y un protocolo para crear y actualizar contenidos.

Data streaming: transferencia continúa de datos sobre una conexión entre un proveedor (normalmente un servidor) y un consumidor (normalmente un cliente).

Eclipse: entorno de desarrollo integrado multiplataforma de código abierto.

HTML (HyperText Markup Language): lenguaje de marcado estándar en la elaboración de páginas Web.

Mercado financiero: espacio físico o virtual en el que se llevan a cabo intercambios de instrumentos financieros y la fijación de sus precios.

Minería de datos: rama de la ciencia de la computación cuyo objetivo es la búsqueda de patrones y la extracción de conocimiento mediante el análisis de grandes volúmenes de datos.

Noticia: relato de un texto informativo.

Opinión: exposición de un pensamiento o una creencia.

PostgreSQL: sistema de gestión de base de datos relacional y orientada a objetos.

RSS (Really Simple Syndication): familia de estándares para la sindicación y difusión de contenidos en la Web.

Trading algorítmico: forma de *trading* supervisado por sistema automático computerizado en la toma de decisiones.

Trading: en este proyecto hace referencia a la acción de operar en un mercado financiero.

URL (Uniform Resource Locator): secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet.

Vocabulario: conjunto de palabras que forman parte de un idioma específico.

Web crawling: búsqueda sistemática de enlaces de interés en Internet, comenzando con una semilla de URLs y utilizando recursivamente los enlaces encontrados dentro de los contenidos.

XML (eXtensible Markup Language): lenguaje de marcado estándar utilizado para el almacenamiento de datos de forma estructurada.

7. Bibliografía

[1] Mitra G., Mitra L., The Handbook of News Analytics in Finance. Ed. Wiley Finance Series, West Sussex, UK, 2010.

[2] Drury B., Torgo L. and Almeida J.J.: Classifying News Stories with a Constrained Learning Strategy to Estimate the Direction of a Market Index. International Journal of Computer Science & Applications 9: 1-22, 2011.

[3] Thomson Reuters. http://thomsonreuters.com/, 2-10-2013

[4] RavenPack. http://www.ravenpack.com/, 2-10-2013

[5] Dow Jones Newswires – Breaking News, Exclusive Analysis & Expert Commentary. http://www.dowjones.com/djnewswires.asp, 2-10-2013

[6] Bloomberg Professional service | Software for Data, Analytics, News. http://www.bloomberg.com/professional/, 2-10-2013

[7] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. Lucene in Action, Second Edition: Covers Apache Lucene 3.0. Manning Publications Co., Greenwich, CT, USA. 2010.

[8] Apache Lucene - Welcome to Apache Lucene. http://lucene.apache.org/

[9] Java SE - Downloads | Oracle Technology Network | Oracle http://www.oracle.com/technetwork/java/javase/downloads/index.html

[10] JavaFX Developer Home http://www.oracle.com/technetwork/java/javafx/overview/index.html

[11] JavaFX SDK | Install JavaFX SDK | Java FX http://www.oracle.com/technetwork/java/javafx/install-javafx-sdk-1-2-139156.html

[12] JavaFX Tools http://www.oracle.com/technetwork/java/javafx/tools/index.html

[13] PostgreSQL: PostgreSQL 9.1 released http://www.postgresql.org/about/news/1349/

[14] EclipseLink Home http://www.eclipse.org/eclipselink/

[15] Eclipse - The Eclipse Foundation open source community website. http://www.eclipse.org/

[16] Commons IO - Commons IO Overview http://commons.apache.org/proper/commons-io/

[17] JDOM http://www.jdom.org/

[18] ROME 1.0 Release. https://rometools.jira.com/wiki/display/ROME/ROME+1.0+Release

[19] Lightstreamer. http://www.lightstreamer.com/

[20] Satnam Alag. Collective Intelligence in Action. Manning Publications Co., Greenwich, CT, USA. 2008.

8. Anexos

Anexo a la memoria del proyecto se presenta:

1. Una presentación en video con una breve descripción del sistema y demostraciones de casos de uso de la aplicación (PresentacionTFG_SanzolSanzFJavier.avi).
2. Un archivo conteniendo el proyecto Eclipse de la implementación desarrollada (AplicacionTFG_SanzolSanzFJavier.zip).