Escuela Universitaria de Ingeniería Técnica Industrial de Bilbao

Memoria del Proyecto

SISTEMA DE PROCESAMIENTO Y DESCARGA AUTOMATIZADA DE TEXTOS MÉDICOS VÍA PÁGINAS WEBS Y TWITTER

Autor:
Jon Ander Hierro Navas

Directoras:
Aitziber Atutxa Salazar
Arantza Casillas Rubio

Grado en Ingeniería Informática de Gestión y Sistemas de Información

Trabajo Fin de Grado

2018 / 2019

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Introducción

Este proyecto tiene como base el trabajo previo elaborado en 2016 por Alma Sainz-Maza Cañive, estudiante de Ingeniería Informática de Gestión y Sistemas de Información entonces, *"Sistema de búsqueda, descarga y procesamiento masivo de textos relacionados con la farmacovigilancia a partir de páginas web, foros y redes sociales"*[3]. El objetivo principal de dicho trabajo era la descarga masiva de textos relacionados con la farmacovigilancia con el propósito de identificar posibles efectos adversos de medicamentos.

La primera fuente de información de este trabajo previo eran páginas webs y foros o blogs de ámbito médico preestablecidos en el software. Por otra parte, se hacía uso de la conocida red social Twitter, descargando los mensajes que la gente publica diariamente acerca de alguna enfermedad o medicamento.

Este software presenta ciertas limitaciones y, pasados los años, ciertas funcionalidades han quedado obsoletas. Por esa razón este trabajo tiene como tarea actualizar e implementar ciertas mejoras en el sistema.

En los apartados de webs y foros se ha automatizado la búsqueda de información médica, es decir, se ha pasado de una búsqueda predeterminada y planificada a una búsqueda genérica. En esta búsqueda se pueden diferenciar dos alternativas:

1. Una lista de webs o foros que el usuario podrá modificar a su gusto en un documento de texto sobre las cuales se hará una búsqueda de forma secuencial.
2. Al usuario se le da la opción de hacer una extracción de información mediante una búsqueda de un término en dos de los motores de búsqueda más conocidos actualmente.
   a) Buscador de Google
   b) Buscador de Bing

En cuanto a la sección de Twitter, se ha actualizado la descarga de mensajes considerando los cambios que ha sufrido la red social estos últimos años. Los cambios más relevantes en Twitter para este proyecto han sido considerar los tuits de más de 140 caracteres y tener en cuenta los *emojis* de los mensajes, ya que el software anterior los eliminaba directamente.

1.2. Antecedente

En este apartado se va a aportar una breve descripción de las características del software proporcionado al comienzo de este proyecto. Como ya se ha comentado previamente, es una herramienta que procesa y descarga textos médicos de páginas webs, foros y Twitter con el propósito de detectar efectos adversos, también conocidos como ADRs, de medicamentos.

La primera fuente de la que se extrae información es de publicaciones profesionales y foros relacionados con el ámbito de la salud. Los datos provienen de dos webs y dos foros preestablecidos y a los cuales se les hizo un estudio individual a cada uno de ellos con el fin de descargar únicamente el contenido que se considerase relevante. Las paginas webs y foros preestablecidos son los siguientes:

- AEMPS | "Asociación Española de Medicamentos y Productos Farmacéuticos"
- SEFH | "Sociedad Española de Farmacia Hospitalaria"
- Forumclinic | Foro en la página oficial del Hospital Clinic de Barcelona.
- DMedicina | Foro donde participan médicos colaboradores.

La segunda forma de extracción es vía Twitter. El proceso consiste en buscar un término, en concreto una enfermedad o un medicamento, y se recuperan los mensajes que incluyen la palabra usada en la búsqueda. Por otra parte, existe la opción de descargar los tuits de las cuentas que sigue el usuario @TFGFVIGILANCIA, propiedad de la alumna que realizó el trabajo, todas ellas relacionadas con la medicina o la farmacovigilancia.

Toda la información descargada se almacena en una base de datos para conservarla pese a que el lugar de donde se extrajo sea eliminado o modificado. Todo el contenido de la base de datos puede recuperarse pasando los datos almacenados a documentos de texto.

1.3. Motivación

En España, de forma periódica, se pregunta a un grupo de habitantes al azar sobre cuáles son sus mayores preocupaciones personales, lo que hace tener una idea general acerca de los temas que más interesan en el país en la actualidad. Según el Centro de Investigaciones Sociológicas, más conocido como CIS, en diciembre de 2018 un 12,9 % de las personas a las que se hizo este cuestionario escogió entre sus tres principales preocupaciones 'La sanidad', la quinta más escogida en ese mes tan solo por detrás del paro, la economía, la corrupción y los políticos.[4]

Este dato nos refleja que la sanidad es un tema importante para la sociedad actual. La sanidad y la salud en general son temas que interesan a gran parte de la población ya que nos afecta de forma directa. Por ese motivo, desde departamentos de investigación y desarrollo de la salud, se está constantemente tratando de averiguar nuevas técnicas y soluciones que hagan que la salud de las personas sea lo mejor posible.

Ciertos estudios médicos precisan de cantidades de información médica grandes para utilizar el contenido de éstas, pero los profesionales de la medicina no tienen el tiempo suficiente para conseguirlas. Es por este motivo que una herramienta que consigue reunir un conjunto de textos del ámbito médico podría ser de mucha utilidad para estudios que pretendan determinar si un texto es médico o no. No todo texto que contenga un término relacionado con la medicina tiene porque considerarse médico, por lo que tener una herramienta que descargue textos médicos de forma automática puede llegar a ser de gran utilidad.

Capítulo 2

Planteamiento inicial

2.1. Descripción del proyecto

Este proyecto va a tener como objetivo procesar y extraer información del ámbito médico de forma automatizada de páginas webs, foros y la red social Twitter para que todos esos datos sean útiles para algún tipo de estudio futuro. Para llevar a cabo esta tarea se van a emplear diferentes fuentes de información, ya sea vía páginas webs, vía API de Twitter o vía motores de búsqueda que nos proporcionen direcciones web de páginas que después se analizarán.

Una vez extraído el texto se harán dos verificaciones. La primera de ellas comprobará si el texto contiene alguna enfermedad o medicamento que se consideren relevantes para el proyecto y la segunda si el texto es considerado médico por un modelo clasificador.

Una vez se tenga el conjunto de textos, las aplicaciones de los mismos son infinitas. El estudio y análisis de los datos podría servir para detectar efectos adversos de medicamentos, identificar patrones de síntomas de enfermedades, determinar qué medicamento es el más elegido por las personas con cierta enfermedad, etc.

2.2. Objetivos

2.2.1. Objetivos principales

Dada una lista compuesta por páginas webs o foros genéricos, la herramienta ha de ser capaz de procesar y extraer información relacionada con el ámbito médico de cualquier página web y/o foro. Otro de los objetivos es hacer uso de APIs y librerías de motores de búsquedas usando términos médicos para encontrar direcciones webs. Una vez se tengan estas direcciones webs, se procederá al procesamiento y extracción de la información del contenido de las mismas.

Por otro lado estaría extraer información relacionada con el ámbito médico de la red social Twitter realizando búsquedas por nombre de medicamento/enfermedad, por una lista de medicamentos/enfermedad o por los tuits escritos por las cuentas que siga @TFGFVIGILANCIA.

Uno de los objetivos con mayor importancia será el de analizar, estudiar y procesar la información para, posteriormente, clasificarla y considerarla médica o no. Este proceso constará de verificar si un texto contiene un medicamento o enfermedad y de un modelo clasificador que distinga entre textos médicos y no médicos.

Por último, en caso de que alguna de las webs predefinidas en el software ahora sean páginas dinámicas, extraer información relacionada con el ámbito médico de páginas webs dinámicas.

2.3. Arquitectura

El estilo de arquitectura que se va a emplear es el conocido como Modelo-Vista-Controlador (MVC). Este nos permite tratar los datos de una aplicación en tres componentes distintos.

- **Modelo**: Representa los datos y las reglas que rigen el acceso y la actualización de estos datos.
- **Vista**: Determina cómo deben presentarse los datos del modelo. Si los datos del modelo cambian, la vista debe actualizar su presentación según sea necesario.
- **Controlador**: Traduce las interacciones del usuario con la vista en acciones que realizará el modelo.

Se ha optado por esta arquitectura en base a las ventajas que ofrece a la hora de elaborar un proyecto. Entre estas ventajas destacan la mayor facilidad para crear casos de pruebas unitarias y una organización del proyecto más clara y comprensible.

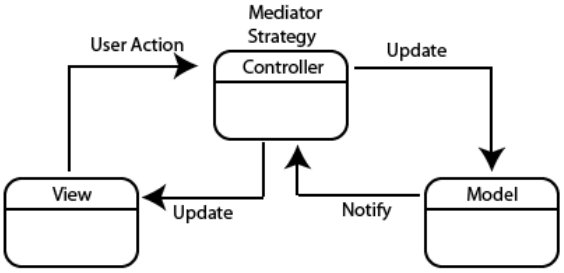


FIGURA 2.1: Arquitectura Modelo Vista Controlador
Fuente: https://www.ablancodev.com/sin-categoria/modelo-vista-controlador/

Por otra parte, en este proyecto se ha empleado una arquitectura mixta. Las arquitecturas a emplear serán tanto la local como la cliente-servidor. La primera de ellas se ha escogido ya que los datos recogidos durante la ejecución del programa serán almacenados de forma local, mientras que la segunda de ellas permitirá el acceso de los datos a través de Internet. Esta elección va a permitir recoger información nueva vía Internet desde páginas webs y Twitter y que toda ésta se almacene de forma permanente, de tal forma que si los datos en Internet son modificados o eliminados, la herramienta conserve esa información.

FIGURA 2.2: Arquitectura empleada

2.4. Alcance

2.4.1. Ciclo de vida

Para este proyecto se ha optado por un ciclo de vida prototipado incremental. Este ciclo de vida consiste en iteraciones en las que se van a implementar ciertas funcionalidades del proyecto de forma que una iteración contenga sus funcionalidades y la de las iteraciones anteriores.

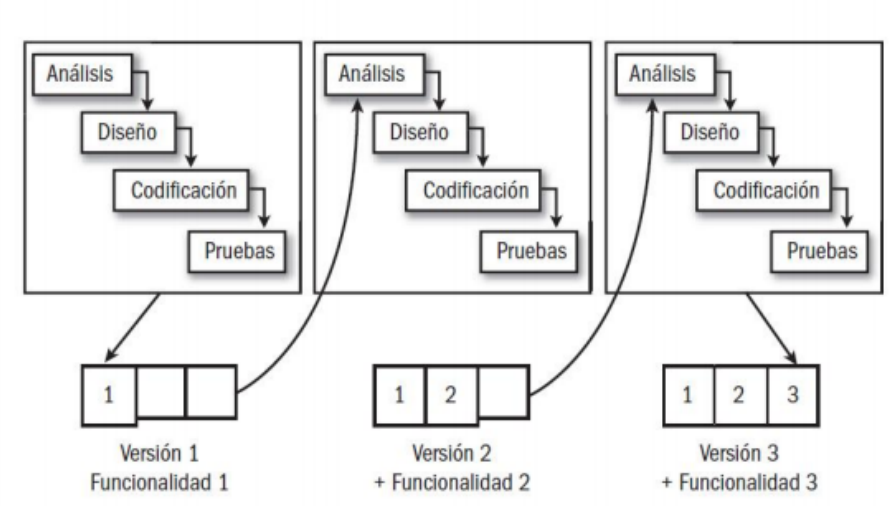


FIGURA 2.3: Ciclo de vida incremental[1]

Este modelo nos va a permitir ir aumentando progresivamente las capacidades del software. En caso de que surja un imprevisto, como un error en alguna función del software, es fácil de gestionar ya que solo habría que modificar o suprimirla de la iteración en la que se desarrolló en vez de hacerlo sobre un proyecto completo.

2.4.2. Prototipos

Para este proyecto en concreto se han identificado, analizado y desarrollado cuatro prototipos distintos.

2.4.2.1. Prototipo 1: Automatizar el procesamiento y descarga de información de Twitter

El primer prototipo tiene como objetivo principal obtener datos relativos al ámbito médico mediante búsquedas por nombres de medicamentos y enfermedades y por tuits de cuentas relacionadas con el tema en cuestión de la red social Twitter.

En el procesamiento de los textos se tendrán en cuenta aquellos tuits de más de 140 caracteres y los que incluyan *emojis*. Todos ellos se adaptan para proporcionarles el formato deseado para almacenarse tanto de forma local en la base de datos como en un documento de texto en el directorio oportuno.

2.4.2.2. Prototipo 2: Automatizar el procesamiento y descarga de información de páginas webs y/o foros

El segundo prototipo se encargará de descargar datos de páginas webs y foros relacionados con el ámbito médico de forma secuencial, esto es, dado un conjunto de páginas webs irá procesando y descargando la información de forma sucesiva de una en una. Todo aquello que se consiga extraer será posteriormente analizado en el prototipo de verificación para averiguar si es de interés para el proyecto.

2.4.2.3. Prototipo 3: Extraer páginas webs de APIs de motores de búsqueda

Este prototipo tiene como fin dotar al software de páginas webs en las que es posible obtener información válida con una mínima interacción del usuario con el software. Se va a hacer uso de las APIs de dos de los motores de búsqueda más usados mundialmente (Google y Bing) para que, dado un término médico, dichos motores de búsqueda nos devuelvan una lista de webs que tienen relación con ese término.

Mediante el prototipo 2, se extraerá la información de los resultados de la lista que se nos haya devuelto y pasará a la verificación.

2.4.2.4. Prototipo 4: Verificar si la información es susceptible de ser médica

Este prototipo se encargará de comprobar que la información descargada es relevante para su posterior aplicación en estudios médicos. Nada nos asegura que, pese a que las páginas que se procesan tienen relación con el ámbito médico, los procesos de descarga nos vayan a proporcionar textos referentes a ese tema en un 100 % de las ocasiones, por ello se ha de revisar la información y clasificarla.

La verificación consiste en una comprobación de si el texto contiene un medicamento o enfermedad y en un modelo clasificador que nos determinará si un texto es considerado médico o, en cambio, no lo es y por tanto no lo tendremos en cuenta para posteriores usos.

Ejemplo de clasificación por parte del modelo:

- "Ayer me sentía raro y mareado así que me tomé una aspirina y me fui a cenar con mis amigos." *nomedico*
- "Hace días el Bilaxten que me recetaron me causó insomnio y náuseas" *medico*

2.5. Herramientas

2.5.1. Hardware

- Portátil Acer Aspire-ES1-711

2.5.2. Software

2.5.2.1. Desarrollo del proyecto

- **Eclipse IDE 2018-09**[5]: Plataforma de desarrollo que proporciona herramientas de programación de código abierto para la gestión y desarrollo de espacios de trabajo, permitiendo ejecutar y depurar aplicaciones.
- **Python IDE - PyDev**[6]: Complemento que permite que Eclipse se use como un IDE de Python.
- **DB Browser for SQLite**[7]: Aplicación diseñada para la creación y administración de bases de datos donde se alojan los textos extraídos.
- **Dropbox**: Servicio de almacenamiento de archivos multiplataforma para guardar copias de seguridad del software y documentos necesarios.
- **Librerías externas de Python**[8]:
  - tkinter: Librería que permite diseñar interfaces gráficas para aplicaciones de escritorio con el lenguaje de programación Python.
  - tweepy: Librería Python para acceder a la API de Twitter.
  - unicodedata: Módulo que posibilita el acceso a la base de datos de caracteres Unicode.
  - re: Librería que proporciona todas las operaciones necesarias para trabajar con las expresiones regulares.
  - os: Módulo que provee de varios métodos para trabajar con las funcionalidades del sistema operativo tales como acceder a una carpeta, crear una carpeta, etc.
  - BeautifulSoup: Biblioteca de Python para analizar documentos HTML.
  - pdfminer: Librería que permite transformar textos de formato PDF a TXT.
  - sqlite3: Librería que permite gestionar la base de datos.
  - google: Librería para búsquedas de webs con el motor de búsqueda Google.
  - azure: Librería para búsquedas de webs con el motor de búsqueda Bing.
  - selenium: Librería para la navegación y extracción de texto de webs dinámicas.

2.5.2.2. Documentación del proyecto

- **Overleaf**[9]: Servicio de LaTeX en línea que permite crear y editar documentos alojándolos en la nube.
- **Visual Paradigm**[10]: Software gratuito que permite la representación de todo tipo de diagramas como casos de uso, diagrama de clases, etc.
- **Planhammer**[11]: Software de planificación de proyectos para la creación de diagramas EDT.
- **Gantt Project**[12]: Aplicación de código abierto que permite diseñar diagramas Gantt para la administración de proyectos.
- **Github**: Software para alojar proyectos, ya sea código o documentación.

2.6. Planificación temporal

2.6.1. EDT

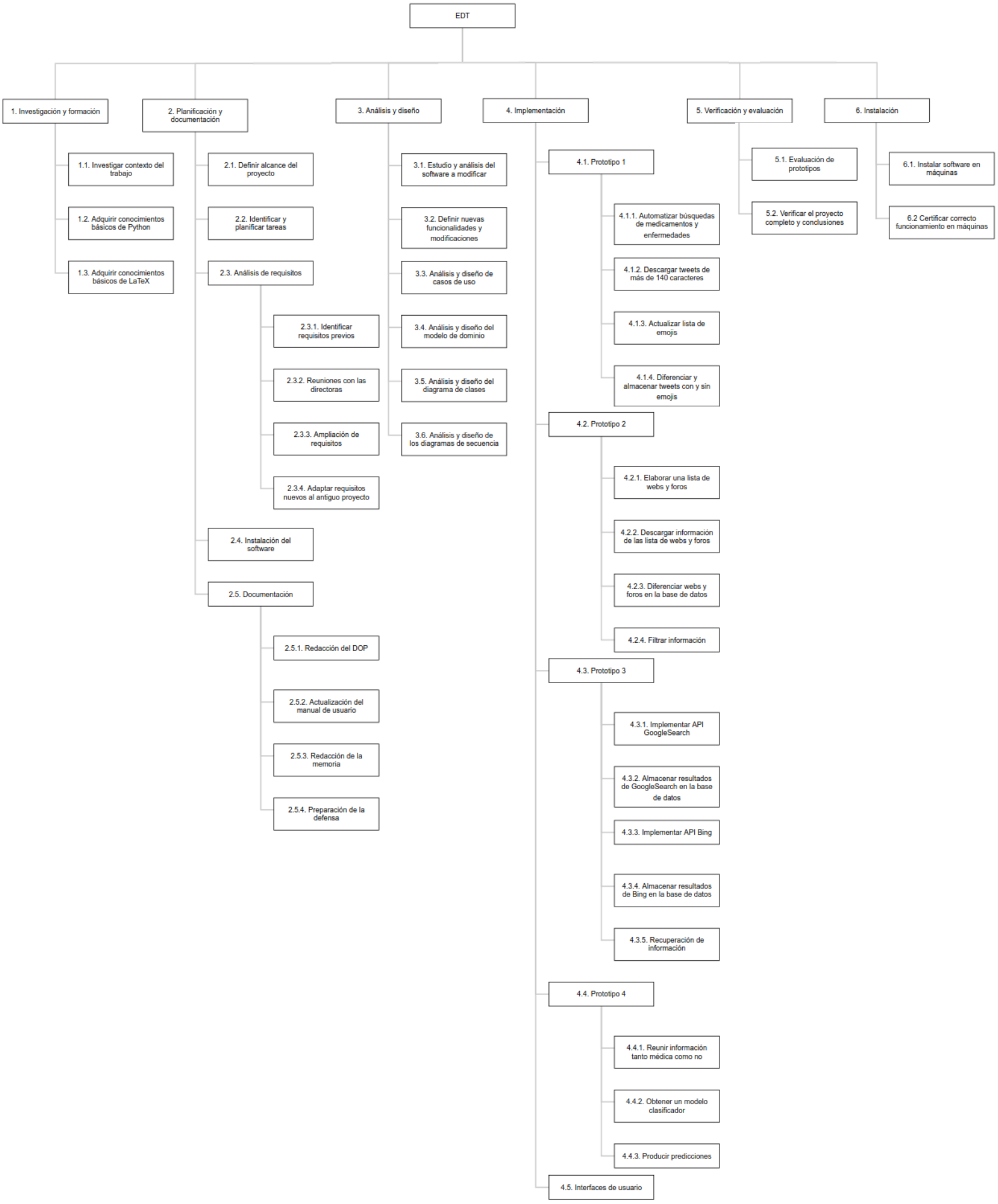


FIGURA 2.4: Diagrama EDT

2.6.2. Descripción de tareas

En esta sección se van a describir cada una de las tareas identificadas en el diagrama EDT anterior.

1. Investigación y formación

Esta es la fase previa de la planificación e implementación del proyecto donde se adquirirán las competencias necesarias para poder abordarlo.

1.1. Investigar contexto del trabajo

En esta tarea se buscará conseguir tener una idea del tema principal del proyecto. Se investigará en Internet publicaciones relacionados con la minería de datos y su aplicación en el ámbito médico.

1.2. Adquirir conocimientos básicos de Python

Una vez determinado el lenguaje con el que se iba a desarrollar el proyecto, se ha precisado de una formación del lenguaje de programación Python para la implementación de la aplicación.

1.3. Adquirir conocimientos básicos de LaTeX

Estudio del la herramienta de composición de textos LaTeX para la redacción de la memoria del trabajo.

2. Planificación y documentación

Esta fase se prolongará desde prácticamente el inicio del proyecto hasta el final del mismo. Esto se debe a que engloba tanto tareas de planificación y preparación para el comienzo de la elaboración del trabajo como las conclusiones del mismo o la redacción de la memoria.

2.1. Definir alcance del proyecto

En esta etapa se establecerán el alcance, qué hay que hacer para desarrollar el proyecto y los objetivos que se quiere conseguir con el proyecto.

2.2. Identificar y planificar tareas

Esta tarea servirá para determinar qué tareas son necesarias, hacer un estudio de cada una de ellas y la elaboración del diagrama EDT.

2.3. Análisis de requisitos

En esta tarea se va a especificar y examinar los requisitos del proyecto. Esta información será fundamental posteriormente para la generación de documentos base para la ejecución de los procesos siguientes.

2.3.1. Identificar requisitos previos

En esta tarea se van a identificar qué requisitos quedaron satisfechos y cuáles no en el trabajo previo.

2.3.2. Reuniones con las directoras

A lo largo del proyecto se van a convocar diferentes reuniones con las directoras del trabajo en las que se irán añadiendo, modificando o eliminando requisitos al proyecto. Cada una de éstas quedará documentada formalmente mediante un acta de reunión.

2.3.3. Ampliación de requisitos

Una vez identificado y acordado el visto bueno por todas las partes implicadas a un nuevo requisito, este será considerado y añadido a los requisitos del proyecto.

2.3.4. Adaptar requisitos nuevos al antiguo proyecto

Esta tarea tiene como fin adecuar las nuevas funcionalidades a las ya existentes en el proyecto anterior.

2.4. Instalación del software

En esta tarea se preparará todo el entorno de desarrollo y se adquirirán y adaptarán todas las tecnologías necesarias para el comienzo de la elaboración del trabajo. La tarea se ve imprescindible ya que el origen de este proyecto es un software previamente desarrollado.

2.5. Documentación

En el trascurso de esta tarea se va a reflejar por escrito todo el progreso del trabajo para, una vez concluido este, dejar constancia del mismo y de cómo se ha ido elaborando.

2.5.1. Redacción del DOP

El Documento de Objetivos del Proyecto va a detallar la descripción del proyecto, los objetivos identificados, el alcance del proyecto, la planificación de las tareas realizar y un análisis de riesgos.

2.5.2. Actualización del manual de usuario

El manual de usuario aportará una explicación de las características de la aplicación, incluyendo las nuevas funcionalidades, y cómo se utiliza de forma correcta el sistema.

2.5.3. Redacción de la memoria

La memoria del proyecto es el documento final en el que se detalla todo el trabajo y las conclusiones una vez concluido.

2.5.4. Preparación de la defensa

La preparación de la defensa consistirá en la composición de la presentación que se va a ensayar y, finalmente, exponer frente a un tribunal.

3. Análisis y diseño

En esta fase se estudiará el software facilitado y se llevarán a cabo todos los diagramas necesarios para describir el nuevo proyecto.

3.1. Estudio y análisis del software a modificar

Esta tarea consiste en analizar y aprender el software proporcionado de cara a identificar dónde se han de hacer las modificaciones o ampliaciones y tener una idea clara del funcionamiento de la aplicación.

3.2. Definir nuevas funcionalidades y modificaciones

En esta tarea se van a especificar y analizar las características nuevas que el software va a requerir o modificaciones de funciones ya existentes.

3.3. Análisis y diseño de casos de uso

Elaboración de los casos de uso para definir y describir cada una de las acciones que un actor primario tiene de usar el sistema.

3.4. Análisis y diseño del modelo de dominio

Elaboración del modelo de dominio para capturar todos los temas relacionados (entidades y datos a almacenar) con el sistema y sus relaciones.

3.5. Análisis y diseño del diagrama de clases

Elaboración del diagrama de clases para mostrar las clases del sistema, sus métodos, sus atributos y las relaciones.

3.6. Análisis y diseño de los diagramas de secuencia

Elaboración de los diagramas de secuencia donde se muestran los envíos de mensajes entre objetos en orden secuencial.

4. Implementación

Esta fase va a ser en la cual se implementen los prototipos previamente identificados y diseñados. Tal y como se ha comentado en apartados anteriores, la implementación será incremental, es decir, los prototipos se desarrollarán uno detrás de otro, obteniendo el trabajo completo una vez terminados todos.

4.1. Prototipo 1: Automatizar el procesamiento y descarga de información de Twitter

El desarrollo del prototipo 1 se basa en las tareas que se van a describir a continuación.

4.1.1. Automatizar búsquedas de medicamentos y enfermedades

Esta tarea se va a encargar de, dada una lista de enfermedades y medicamentos, la aplicación busque tuits que contengan esos términos de forma automatizada.

4.1.2. Descargar tuits de más de 140 caracteres

Plataformas tan grandes e importantes como Twitter se actualizan constantemente para adaptarse a los cambios y nuevas tecnologías que surgen cada vez con más frecuencia. Una de estas actualizaciones fue aumentar el número de caracteres por tuit, de 140 a 280 caracteres. En esta tarea se abordará la modificación de la funcionalidad ya existente para amoldarla a los nuevos mensajes de más de 140 caracteres.

4.1.3. Actualizar lista de *emojis*

Otra de las actualizaciones de Twitter incluyó nuevos *emojis*, los cuales los usuarios podrían usar en sus mensajes. Esta tarea se encargará de actualizar la lista de *emojis* para poder reconocer los *emojis* añadidos.

4.1.4. Diferenciar y almacenar tuits con y sin *emojis*

Esta tarea tiene como finalidad distinguir tuits con y sin *emojis*. Esta discriminación fdará la posibilidad de hacer un estudio de *emojis* que aparecen en los mensajes, ya que pueden aportar información extra aparte del texto en sí.

4.2. Prototipo 2: Automatizar el procesamiento y descarga de información de páginas webs y/o foros

El desarrollo del prototipo 2 se basa en las tareas que se van a describir a continuación.

4.2.1. Elaborar una lista de webs y foros

En esta tarea se buscarán webs y foros de los que extraer información, a priori, del ámbito médico y se guardarán en su lista correspondiente.

4.2.2. Descargar información de las lista de webs y foros

En esta tarea se procesará y extraerá el contenido de los webs y foros que estén contenidos en las listas elaboradas previamente y todas aquellas que el web crawler detecte dentro de todas las páginas webs que se analicen.

4.2.3. Diferenciar webs y foros en la base de datos

En esta tarea se adaptará la base de datos ya existente para acomodarla a las necesidades actuales. El cambio consiste en almacenar la información procedente de páginas webs genéricas y de foros en dos tablas independientes, de forma que sea más cómodo diferenciar la procedencia de un texto.

4.2.4. Filtrar información

Debido a que la probabilidad de que el software visite páginas webs que no tengan relación con los temas que nos interesen y descargue la información se aplicará un filtro a todos los textos. Este filtro consiste en verificar si el texto descargado contiene el nombre de alguna enfermedad o medicamento que consideremos relevante. En caso afirmativo se descargará, en caso negativo, por el contrario, se descartará.

4.3. Prototipo 3: Extraer páginas webs de APIs de motores de búsqueda

El desarrollo del prototipo 3 se basa en las tareas que se van a describir a continuación.

4.3.1. Implementar API GoogleSearch

Esta tarea tiene como fin crear e implementar todo lo necesario para poder hacer uso de la API de Google que nos va a permitir recabar links que posteriormente van a ser procesados y descargados.

4.3.2. Almacenar resultados de GoogleSearch en la base de datos

Esta tarea se encarga de almacenar los datos extraidos de la búsqueda con la API de Google en la base de datos con un identificador que permitirá que la API no devuelva siempre los mismos links.

4.3.3. Implementar API Bing

Esta tarea tiene como fin crear e implementar todo lo necesario para poder hacer uso de la API de Bing que nos va a permitir recabar links que posteriormente van a ser procesados y descargados.

4.3.4. Almacenar resultados de Bing en la base de datos

Esta tarea se encarga de almacenar los datos extraidos de la búsqueda con la API de Bing en la base de datos con un identificador que permitirá que la API no devuelva siempre los mismos links.

4.3.5. Recuperación de información

Se podrá extraer la información almacenada en la base de datos de ambas APIs en formato txt.

4.4. Prototipo 4: Verificar si la información es susceptible de ser médica

El desarrollo del prototipo 4 se basa en las tareas que se van a describir a continuación.

4.4.1. Reunir información tanto médica como no

Esta tarea se encarga de reunir textos que se consideren médicos y textos que no se consideren médicos. Los textos médicos se considerarán aquellos que se encontraban en el software original y los no médicos serán textos descargados de Wikipedia.

4.4.2. Obtener un modelo clasificador

Esta tarea se va a encargar de crear y guardar un modelo clasificador que sepa distinguir entre textos médicos y no médicos.

4.4.3. Producir predicciones

Una vez tenemos ya el modelo clasificador, este realizará predicciones en base a un texto que se le haga llegar y lo clasificará como “medico” o “nomedico”.

4.5. Interfaces de usuario

En esta tarea se adaptará la interfaz de usuario a las nuevas funcionalidades y modificaciones que se han aplicado en el software.

5. Verificación y evaluación

En esta fase se verificará que los distintos prototipos y el proyecto en general funcionan de forma correcta y cumplen con los requisitos.

5.1. Evaluación de prototipos

En esta tarea se llevará a cabo las pruebas que certifiquen que cada uno de los prototipos implementados funcionan tal y como en un principio se planearon.

5.2. Verificar el proyecto completo y conclusiones

La última tarea se ocupará de hacer una evaluación final del trabajo entero y se recabarán las conclusiones del progreso y del resultado final.

6. Instalación

En esta fase final se migrará el proyecto del ordenador donde se ha implementado a los ordenadores del Departamento de Informática y se llevará a cabo los últimos pasos para cumplir los objetivos.

6.1. Instalar software en máquinas

En esta tarea se instalará el software ya terminado y todas las librerías necesarias en las máquinas del Departamento de Informática.

6.2. Certificar correcto funcionamiento en máquinas

Una vez esté el software completo instalado se confirmará que este funciona de forma correcta y esperada.

2.6.3. Diagramas Gantt

El diagrama Gantt permite describir gráficamente la distribución de las tareas a lo largo del tiempo. En este caso se realizarán dos diagramas, uno, al comienzo del proyecto, para la estimación inicial y otro, al finalizarlo, para ver la distribución real que se ha llevado a cabo.

2.6.3.1. Diagrama inicial

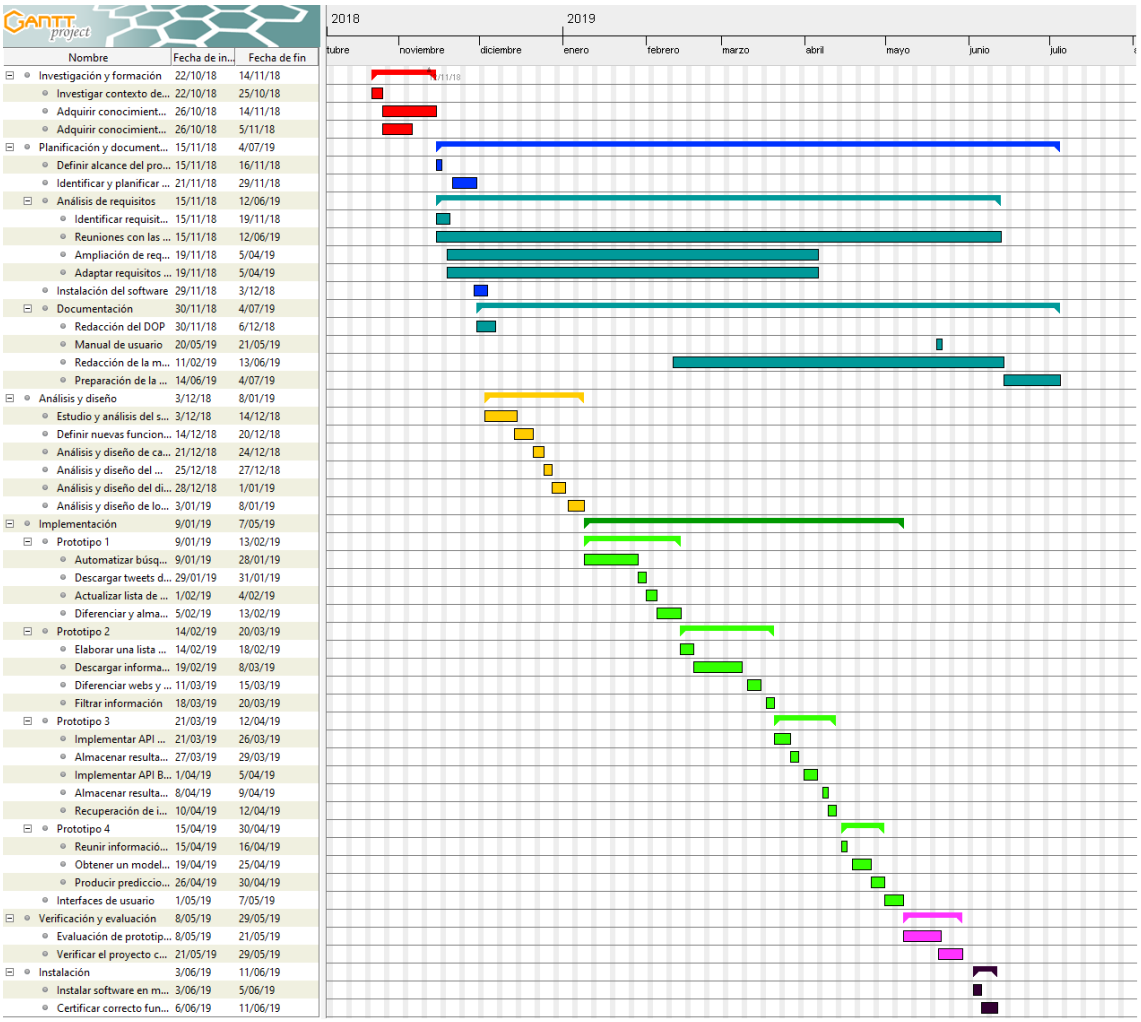
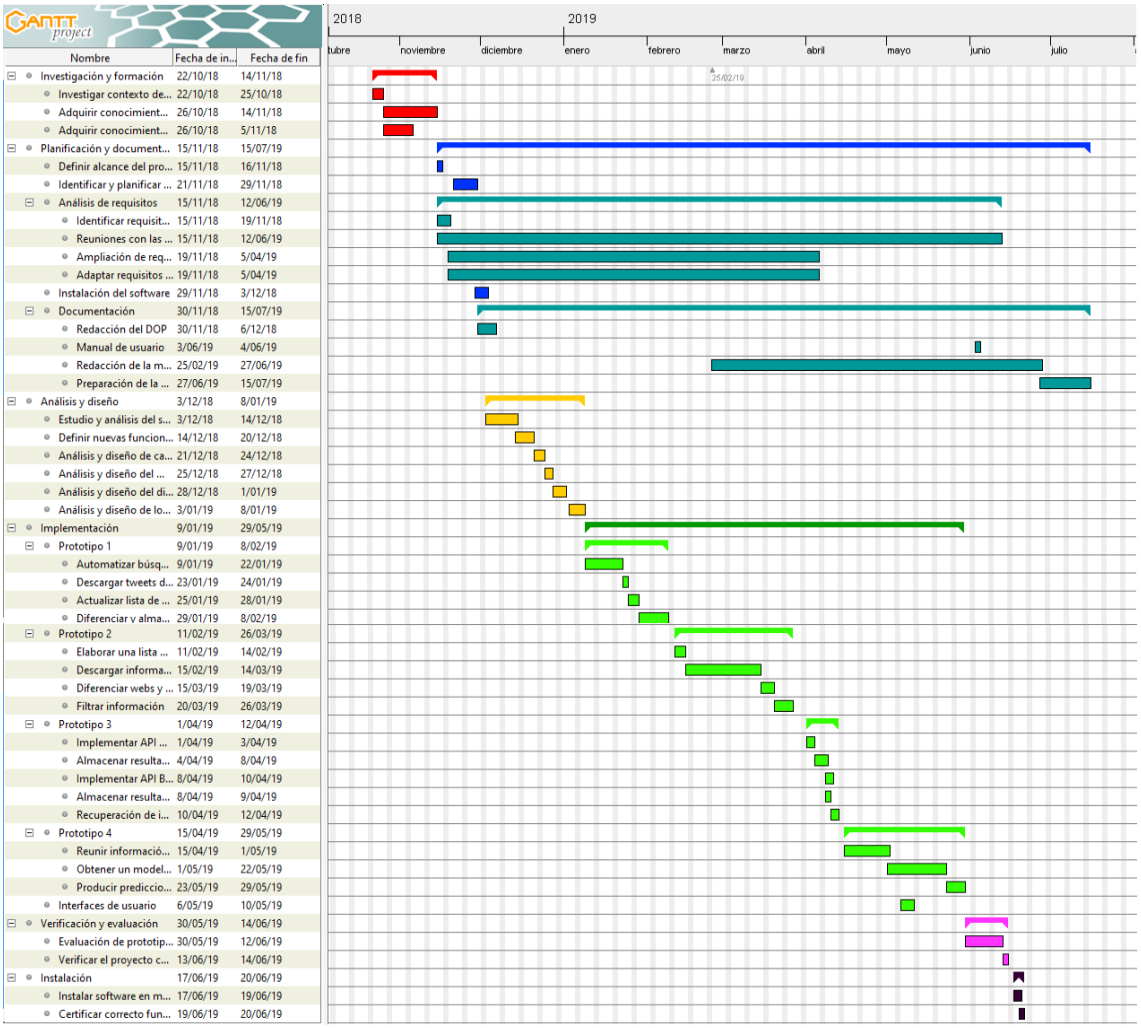


FIGURA 2.5: Diagrama Gantt inicial

2.6.3.2. Diagrama final



Nombre	Fecha de in...	Fecha de fin
Investigación y formación	22/10/18	14/11/18
Investigar contexto de...	22/10/18	25/10/18
Adquirir conocimient...	26/10/18	14/11/18
Adquirir conocimient...	26/10/18	5/11/18
Planificación y document...	15/11/18	15/07/19
Definir alcance del pro...	15/11/18	16/11/18
Identificar y planificar ...	21/11/18	29/11/18
Análisis de requisitos	15/11/18	12/06/19
Identificar requisit...	15/11/18	19/11/18
Reuniones con las ...	15/11/18	12/06/19
Ampliación de req...	19/11/18	5/04/19
Adaptar requisitos ...	19/11/18	5/04/19
Instalación del software	29/11/18	3/12/18
Documentación	30/11/18	15/07/19
Redacción del DOP	30/11/18	6/12/18
Manual de usuario	3/06/19	4/06/19
Redacción de la m...	25/02/19	27/06/19
Preparación de la ...	27/06/19	15/07/19
Análisis y diseño	3/12/18	8/01/19
Estudio y análisis del s...	3/12/18	14/12/18
Definir nuevas funcion...	14/12/18	20/12/18
Análisis y diseño de ca...	21/12/18	24/12/18
Análisis y diseño del ...	25/12/18	27/12/18
Análisis y diseño del di...	28/12/18	1/01/19
Análisis y diseño de lo...	3/01/19	8/01/19
Implementación	9/01/19	29/05/19
Prototipo 1	9/01/19	8/02/19
Automatizar búsq...	9/01/19	22/01/19
Descargar tweets di...	23/01/19	24/01/19
Actualizar lista de ...	25/01/19	28/01/19
Diferenciar y alma...	29/01/19	8/02/19
Prototipo 2	11/02/19	26/03/19
Elaborar una lista ...	11/02/19	14/02/19
Descargar informa...	15/02/19	14/03/19
Diferenciar webs y ...	15/03/19	19/03/19
Filtrar información	20/03/19	26/03/19
Prototipo 3	1/04/19	12/04/19
Implementar API ...	1/04/19	3/04/19
Almacenar resulta...	4/04/19	8/04/19
Implementar API B...	8/04/19	10/04/19
Almacenar resulta...	8/04/19	9/04/19
Recuperación de i...	10/04/19	12/04/19
Prototipo 4	15/04/19	29/05/19
Reunir informació...	15/04/19	1/05/19
Obtener un model...	1/05/19	22/05/19
Producir prediccio...	23/05/19	29/05/19
Interfaces de usuario	6/05/19	10/05/19
Verificación y evaluación	30/05/19	14/06/19
Evaluación de prototip...	30/05/19	12/06/19
Verificar el proyecto c...	13/06/19	14/06/19
Instalación	17/06/19	20/06/19
Instalar software en m...	17/06/19	19/06/19
Certificar correcto fun...	19/06/19	20/06/19

FIGURA 2.6: Diagrama Gantt final

2.6.4. Desviaciones temporales

Las desviaciones temporales surgen cuando se dan imprevistos en la elaboración de un trabajo. Estas desviaciones hacen que el tiempo real de elaboración no se ajuste al que en un principio se había estimado. Tal y como puede apreciarse en las figuras 2.5 y 2.6, los diagramas Gantt inicial y final no son idénticos. Esto se debe a los siguientes factores:

- Avería del ordenador en el que se desarrollaba el proyecto.
- Decisiones erróneas a la hora de desarrollar nuevas funcionalidades.
- Comienzo de prácticas externas voluntarias.

Las diferencias más sustanciales entre ambas distribuciones de tareas están en el inicio de la redacción de la memoria y la implementación del Prototipo 2 y 4. Las desviaciones más notables se deben básicamente por el comienzo de las prácticas externas y por internar, en un principio, crear un modelo clasificador utilizando la librería de Weka con el lenguaje de programación Java, lo cual hizo que el Prototipo 4 se dilatase en el tiempo más de lo esperado.

2.7. Gestión de riesgos

Los riesgos no son ajenos a cualquier tipo de proyecto, en cualquier momento puede surgir un imprevisto que torpedee y, en consecuencia, retrase el desarrollo de un proyecto. He ahí la importancia de identificar estos riesgos con el objetivo de elaborar planes que minimicen su efecto sobre el proyecto.

2.7.1. Planificación de gestión de riesgos

Un plan de riesgos es particular a un proyecto en concreto, puede que haya coincidencias con otros, pero no puede generalizarse ni hacer uso de un plan de riesgos estándar. Por ello, en este proyecto se han identificado los siguientes riesgos:

- Pérdida de la información.
- Incumplimiento en la planificación temporal.
- Problemas con los dispositivos informáticos.
- Falta de conocimiento o problemas con los lenguajes de programación.
- Software proporcionado obsoleto.
- Filtro de información médica poco efectivo.
- *Inter-tagger agreement* excesivamente bajo.
- Cambios en los requisitos del proyecto.
- Trabajo y/o estudios simultáneos al proyecto.
- Problemas de índole personal.

Es obvio que no todos los riesgos tienen una probabilidad idéntica. Realizar un estudio particular de cada riesgo y asignar una probabilidad de que este suceda puede llevar un tiempo excesivo y no es una tarea sencilla, por lo que se ha optado en generalizar la probabilidad de que un riesgo suceda en tres niveles. Dichos niveles se muestran en la Tabla 2.18.

Identificación de probabilidad	
Nivel	Probabilidad
Baja	0-25 %
Media	25-75 %
Alta	75-100 %

Tabla 2.1: Niveles de probabilidad de un riesgo

A su vez, no todos los riesgos tendrán un mismo impacto sobre el proyecto si se llegasen a producir. Para identificar el impacto, tiempo que se retrasaría el proyecto si se da un riesgo, se ha generalizado en 5 niveles en la Tabla 2.2.

Identificación de impacto	
Nivel	Retraso
Insignificante	Menos de 1 día
Menor	Entre 1 y 3 días
Moderado	Entre 3 y 5 días
Mayor	Entre 5 y 7 días
Catastrófico	Más de 7 días

Tabla 2.2: Niveles de impacto de un riesgo

2.7.2. Plan de riesgos

Lo ideal en un proyecto sería que no se diese ningún riesgo, pero, ya que esto es poco probable por no decir imposible, se ha de elaborar un plan de prevención para cada riesgo y otro de contingencia que indique los pasos a seguir si un riesgo ocurre.

Pérdida de la información	
Descripción	Pérdida de parte del proyecto, ya sea implementación o documentación.
Plan de prevención	Realizar copias de seguridad de forma diaria en la nube o, aunque sea menos recomendable, en dispositivo de almacenamiento externos.
Plan de contingencia	Identificar qué datos se han perdidos y recuperar la última copia de seguridad.
Nivel de probabilidad	Media
Nivel de impacto	Moderado

Tabla 2.3: Pérdida de la información

Incumplimiento en la planificación temporal	
Descripción	Retraso de alguna de las tareas por causa personal o ajena.
Plan de prevención	Realizar una planificación realista de cada tarea e ir desarrollando el proyecto de forma que se cumplan los plazos establecidos.
Plan de contingencia	Replanificar la tarea de forma que se aumente el tiempo estimado para la realización de ésta.
Nivel de probabilidad	Alta
Nivel de impacto	Moderado

Tabla 2.4: Incumplimiento en la planificación temporal

Problemas con los dispositivos informáticos	
Descripción	Problemas o averías de algún dispositivo informático que se esté usando para el desarrollo del proyecto.
Plan de prevención	Manipular los dispositivos de forma que no sufran ningún daño físico y disponer de un antivirus y cortafuegos que proteja el apartado del software.
Plan de contingencia	Reparar los dispositivos y, en caso de no ser posible esta opción, adquirir nuevos.
Nivel de probabilidad	Baja
Nivel de impacto	Menor

Tabla 2.5: Problemas con los dispositivos informáticos

Falta de conocimiento o problemas con los lenguajes de programación	
Descripción	Dudas técnicas en un inicio con el lenguaje de programación Python.
Plan de prevención	Leer artículos y visionar vídeos que sirvan de aprendizaje.
Plan de contingencia	Consultar dudas en Internet, ya sea páginas webs o foros.
Nivel de probabilidad	Alta
Nivel de impacto	Menor

Tabla 2.6: Falta de conocimiento o problemas con los lenguajes de programación

Software proporcionado obsoleto	
Descripción	El programa que se facilitó no funciona por completo o algunas de sus funcionalidades no funcionan como se estimaba en un principio.
Plan de prevención	Verificar de forma cuidadosa que la ejecución del proyecto por completo, probando todas las funcionalidades, es tal y como se especificaba en la memoria del trabajo del proyecto.
Plan de contingencia	Adaptar las funcionalidades a la actualidad procurando que todas ellas funcionen tal y como lo hacían en su momento.
Nivel de probabilidad	Baja
Nivel de impacto	Catastrófico

Tabla 2.7: Software proporcionado obsoleto

Filtro de información médica poco efectivo	
Descripción	El modelo empleado clasifica correctamente en un porcentaje muy bajo los textos que se le pasan como posibles textos médicos. Un caso posible es que siempre clasifique todos los textos con la misma clase.
Plan de prevención	Selección de un clasificador lo más adecuado posible y reunir cuanta más información médica y no médica, que estas, sean lo más claras posibles.
Plan de contingencia	Agrupar más información médica y buscar textos no médicos más variados, es decir, que no traten únicamente de un tema.
Nivel de probabilidad	Media
Nivel de impacto	Moderado

Tabla 2.8: Filtro de información médica poco efectivo

Inter-tagger agreement excesivamente bajo	
Descripción	El *inter-tagger agreement* mide el acuerdo entre las etiquetas asignadas al mismo elemento por dos anotadores diferentes. Que el acuerdo sea bajo indicará que la máquina tendrá una capacidad baja para poder clasificar correctamente los textos.
Plan de prevención	Definir con exactitud desde un principio qué es lo que se considera texto médico y qué es lo que no.
Plan de contingencia	Crear un modelo clasificador que se ajuste más a las necesidades.
Nivel de probabilidad	Media
Nivel de impacto	Moderado

Tabla 2.9: *Inter-tagger agreement* excesivamente bajo

Cambios en los requisitos del proyecto	
Descripción	Detección de nuevas necesidades o la modificación/eliminación de una ya existente.
Plan de prevención	Desarrollar el proyecto de forma que no se asuma un coste temporal alto el hecho de añadir nuevas funcionalidades.
Plan de contingencia	Analizar la nueva propuesta y, una vez estudiada, comenzar la implementación.
Nivel de probabilidad	Alta
Nivel de impacto	Moderado

Tabla 2.10: Cambios en los requisitos del proyecto

Trabajo y/o estudios simultáneos al proyecto	
Descripción	Invertir un tiempo considerable de nuestro día a día en otra actividad laboral o académica.
Plan de prevención	Tener en cuenta las horas invertidas en otras actividades y administrar las restantes para que sea posible progresar en el proyecto.
Plan de contingencia	Si es necesario, realizar un replanteamiento temporal del proyecto.
Nivel de probabilidad	Media
Nivel de impacto	Mayor

Tabla 2.11: Trabajo y/o estudios simultáneos al proyecto

Problemas de índole personal	
Descripción	Acontecimiento de un hecho que se pueda catalogar como problema familiar o personal.
Plan de prevención	Tratar de evitar enfermedades y conservar un buen estado de ánimo, ya que los problemas ajenos no se pueden prevenir.
Plan de contingencia	Centrarse en el trabajo lo máximo que sea posible en una de esas situaciones.
Nivel de probabilidad	Baja
Nivel de impacto	Mayor

Tabla 2.12: Problemas de índole personal

2.8. Evaluación económica

2.8.1. Costes

2.8.1.1. Mano de obra

El coste de mano de obra es el dinero que le corresponde al programador del proyecto. El sueldo que el programador percibe se va a estimar en 15€ la hora.

Coste de la mano de obra		
Sueldo	Horas totales	Coste Mano de Obra Total
15 €/h	420	6300 €

Tabla 2.13: Coste de la mano de obra

2.8.1.2. Indirectos

Los costes indirectos son aquellos característicos del desarrollo de un proyecto, pero que aun así son necesarios para que este prospere. Para este proyecto se han identificado gastos en electricidad e Internet. Es difícil calcular el coste exacto de ambas por lo que se va a establecer un precio fijo mensual en el tiempo que se ha elaborado el proyecto.

Coste indirecto			
Herramienta	Precio	Meses	Coste parcial
Electricidad	20 €	8	160 €
Internet	15 €	8	120 €
Coste Indirecto Total			280 €

Tabla 2.14: Coste indirecto

2.8.1.3. Hardware

El coste en hardware se atribuye a las herramientas utilizadas a lo largo del proceso de creación del proyecto. Se ha estimado que la vida útil del ordenador usado es de 4 años, es decir, 48 meses de vida útil y su coste fue de 460€. El cálculo de este coste vendrá dado por la siguiente fórmula:

$$Coste\,Hardware = \frac{Precio * Meses\,de\,uso}{Meses\,de\,vida\,útil}$$

Coste de hardware				
Herramienta	Precio	Meses de uso	Meses de vida útil	Coste Hardware Total
Acer Aspire-ES1-711	460 €	7	48	67,08 €

Tabla 2.15: Coste de hardware

2.8.1.4. Software

El coste en software será el correspondiente al gasto en programas para desarrollar el proyecto o para la elaboración de la documentación.

Coste de software	
Software	Precio
Eclipse IDE	0 €
Python IDE	0 €
DB Browser for SQLite	0 €
Dropbox	0 €
Librerías Python	0 €
Overleaf	0 €
Visual Paradigm	0 €
Planhammer	0 €
Gantt Project	0 €
Github	0 €
Coste Software Total	0 €

Tabla 2.16: Coste de software

2.8.2. Coste total

Una vez desglosados todos los costes, en este apartado se va a calcular el coste total del proyecto que será ni más ni menos que la suma de todos los costes del apartado anterior.

Coste total	
Tipo	Coste
Coste de mano de obra	6300 €
Coste indirecto	280 €
Coste software	0 €
Coste hardware	67,08 €
Coste Total Proyecto	6647,08 €

Tabla 2.17: Coste total

2.8.3. Amortización

Este trabajo en cuestión no podría generar ningún tipo de beneficio ya que, al ser propuesto por la misma universidad, no está permitida su comercialización y no podría contemplarse una posible venta del software.

Aún así, se sabe del potencial de la herramienta y la utilidad que le podría dar un centro de salud, por lo que en el hipotético caso de que el software pudiera venderse, los gastos que habría que amortizar serían de 6647,08 €. El valor de la aplicación se ha estimado en 800 €/unidad.

Precio unitario	IVA	Precio final	Unidades a vender	Ingresos
800 €	21 %	968 €	7	6776 €
Coste Total				6647,08 €

Tabla 2.18: Amortización

Capítulo 3

Análisis de antecedentes

En este apartado se darán unas breves explicaciones de herramientas ya existentes que llevan a cabo alguna de las funcionalidades de este trabajo, ya sea rescatar mensajes de Twitter, descargar textos de webs y foros genéricos o clasificar textos del ámbito médico de forma masiva.

3.1. "Towards Internet-Age Pharmacovigilance: Extracting Adverse Drug Reactions from User Posts to Health-Related Social Networks"

El objetivo de este trabajo [17] era identificar efectos adversos en mensajes que publicaban usuarios en la red social relacionada con la salud DailyStrength. Esta red social fue elegida como fuente ya que permite a los usuarios crear perfiles, mantener amigos y unirse a varios grupos de apoyo relacionados con enfermedades para hacer consultas a usuarios que hayan estado o estén en el mismo estado de salud.

La búsqueda se centró en 4 medicamentos en concreto, por lo que no se tuvo que reconocer nombres de medicamentos. Después de reunir los mensajes de los usuarios, se catalogaron como *adverse effect*, *beneficial effect*, *indication* u *other*.

Se evaluó el sistema con 3.150 comentarios etiquetados que no estaban reservados para el desarrollo del sistema. Los resultados de este estudio fueron 78.3 % de *precision* y 69.9 % recall, para una f-measure de 73.9 %.

3.2. "Recuperación de tuits relacionados con el uso de drogas usando técnicas de extracción de terminología"

En este trabajo desarrollado por investigadores de un hospital e investigadores de un Departamento Universitario de Informática se lleva a cabo una recuperación de tuits relacionados con el uso no terapéutico de drogas. La motivación del trabajo era extraer conocimiento de utilidad clínica desde Twitter en el ámbito del uso no medicinal de las drogas. Por una parte se quería extraer terminología sobre drogas usada por los usuarios de Twitter y, por otra, reunir un conjunto de tuits que en un futuro pudieran ser de utilidad.

El proceso consiste en dado un conjunto de términos semilla inicial se rescatan tuits usando la API Twitter, se realiza un cálculo de candidatos y aquellos más significativos se añadirán al conjunto de términos semilla para futuras búsquedas. De este modo, el conjunto semilla va mejorando y creciendo en número de forma iterativa.

El estudio concluye con un experimento utilizando la palabra "droga" como semilla. La precisión que se obtuvo fue mayor al 80 % en un tema como las drogas que es tan ambiguo, esto es, que no siempre que alguien se refiera al tema de las drogas será en términos no terapéuticos.

3.3. "Malpractice and Malcontent: Analyzing Medical Complaints in Twitter"

Este trabajo [18] está orientado a recopilar información vía Twitter acerca de quejas o errores médicos que alguien hubiera sufrido. Para la recopilación de textos se hizo búsquedas en la API de Twitter de términos como *sue the doctor* ("demandar al doctor") o *nurse make a mistake* ("enfermera comete un error"). En total se identificaron 170 mensajes que se podrían incluir dentro del ámbito de la seguridad de un paciente.

El estudio encontró que los errores mayoritariamente eran auto-informados o informados por familiares. Uno de los aspectos que más llamaron la atención fue el hecho de encontrar muchos errores graves, algunos de los cuales, según el usuario, podrían causar la muerte.

3.4. "GENVL and WWWW: Tools for taming the web"

World Wide Web Worm [19] fue desarrollado por Oliver McBryan, en la Universidad de Colorado en Boulder a fines de 1993 y se considera uno de los primeros motores de búsqueda.

El gusano creó una base de datos de 300.000 objetos multimedia que se podrían obtener o buscar por palabras clave a través de la World Wide Web. Por otra parte, consiguió indexar cerca de 110.000 páginas web a partir de 1994.

FIGURA 3.1: Interfaz World Wide Web Worm
Fuente: http://www.internethistorypodcast.com/2016/11/was-the-world-wide-web-worm-the-first-web-search-engine/

3.5. "PyBot: An Algorithm for Web Crawling"

PyBot [2] es un simple web crawler implementado en Python 2.7 en 2011. Este crawler utiliza el algoritmo de búsqueda Breadth First Search (BFS) que es el más popular para rastrear la web. BFS utiliza la estrategia en la que el nodo raíz se expande primero, luego todos los sucesores del nodo raíz se expanden a continuación, luego sus sucesores, etc. El nodo raíz es la URL y todos los hipervínculos se visitan y se descargan, y así sucesivamente.

FIGURA 3.2: Interfaz PyBot [2]

PyBot se probó en 2010 con la web www.curtin.edu.my y consiguió rastrea el sitio, recupera todos los hipervínculos y lo guardarlos en un formato CSV de Excel. Además de rastrear, descargó las páginas y guardó información adicional, como todas las páginas rastreadas, páginas colgantes, páginas no colgadas, páginas visitadas de robots, páginas de robots.txt y enlaces externos salientes.

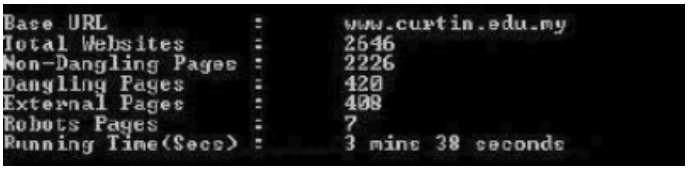


FIGURA 3.3: Resultado de la ejecución de PyBot rastreando www.curtin.edu.my [2]

Capítulo 4

Captura de requisitos

4.1. Casos de uso

Los casos de uso describen qué acciones puede realizar una persona que vaya a utilizar el sistema. Una persona que use el sistema se denominará actor y las acciones que pueda llevar a cabo se llamarán casos de uso.

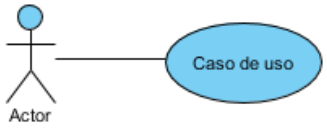


FIGURA 4.1: Ejemplo de casos de uso

A continuación, se identificarán el o los actores que pueden tomar parte en el sistema y los casos de uso de este proyecto. Por último, por cada caso de uso identificado en el diagrama se hará una breve descripción explicando en qué consiste esa funcionalidad.

4.1.1. Jerarquía de actores

La jerarquía de actores distinguirá los distintos tipos de usuarios que podrán hacer uso de la aplicación. En este caso nos encontramos con solo un tipo de usuario ya que no hace falta estar identificado para acceder a las diferentes funcionalidades del proyecto.

FIGURA 4.2: Jerarquía de actores

4.1.2. Diagrama de casos de uso

En este apartado se va a mostrar el diagrama que refleja los casos de uso de este proyecto.

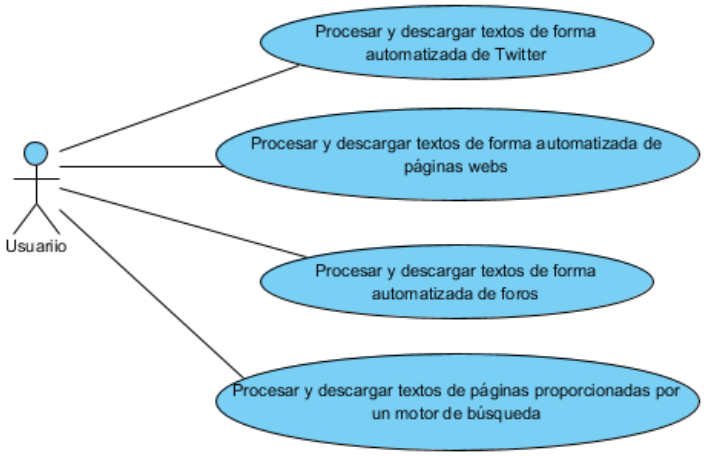


FIGURA 4.3: Diagrama de casos de uso

4.1.2.1. Procesar y descargar textos de forma automatizada de Twitter

Dada una lista de medicamentos o enfermedades procesa y descarga de forma automatizada y secuencial la información de aquellos tuits que contengan algún término de la lista.

4.1.2.2. Procesar y descargar textos de forma automatizada de páginas webs

Dada una lista de páginas webs relacionadas con el ámbito médico, procesa y descarga el contenido de forma automatizada y secuencial con el objetivo de que sea de utilidad en un estudio futuro.

4.1.2.3. Procesar y descargar textos de forma automatizada de foros

Dada una lista de foros relacionados con el ámbito médico, procesa y descarga el contenido de forma automatizada y secuencial con el objetivo de que sea de utilidad en un estudio futuro.

4.1.2.4. Procesar y descargar textos de páginas proporcionadas por un motor de búsqueda

Dado un término médico (enfermedad o medicamento) el motor de búsqueda devuelve un conjunto de direcciones web relacionados con dicho término para que procese y descargue la información de esas páginas.

Capítulo 5

Análisis y diseño

5.1. Análisis

- **Prototipo 1**: Automatizar el procesamiento y descarga de información de Twitter
  - Procesar y tratar tuits.
  - Distinguir entre tuits con y sin emoticonos.
  - Descargar tuits.
    - Por medicamento/enfermedad
    - Por lista de medicamentos/enfermedades
    - Por usuarios seguidos por @TFGFVIGILANCIA
  - Almacenar tuits en la base de datos.
  - Convertir tuits almacenados en la base de datos en documentos de texto.
- **Prototipo 2**: Automatizar el procesamiento y descarga de información de páginas webs
  - Procesar y tratar textos.
  - Clasificar textos.
    - Médicos
    - No médicos
  - Descargar textos.
  - Almacenar textos en la base de datos.
  - Convertir datos almacenados en la base de datos en documentos de texto.
- **Prototipo 3**: Extraer páginas webs de APIs de motores de búsqueda
  - Procesar y tratar textos.
  - Clasificar textos.
    - Médicos
    - No médicos
  - Descargar textos.
  - Almacenar textos en la base de datos.
  - Convertir datos almacenados en la base de datos en documentos de texto.

5.2. Diseño

Como ya se ha explicado previamente, este trabajo fin de grado tiene como base otro hecho con anterioridad. Para que el proyecto tenga una continuidad, se ha decidido seguir el diseño del proyecto base modificando y añadiendo los elementos necesarios para este trabajo.

Para poder visualizar de forma más clara qué alteraciones ha sufrido el diseño, los elementos nuevos o modificados se distinguirán de color verde.

5.2.1. Diagramas de clase

En este apartado se van a mostrar los diagramas de clase que se han diseñado. Se van a dividir por prototipos, por lo que se han diseñado tres diagramas de clase, uno por cada uno de los tres primeros prototipos. La razón por la que el cuarto prototipo no tiene un diagrama propio es que está integrado en el segundo prototipo y en el tercero. Por último, para una mejor visualización, todo atributo o variable que no se especifique su tipo se supondrá que es de tipo String.

5.2.1.1. Prototipo 1: Automatizar el procesamiento y descarga de información de Twitter

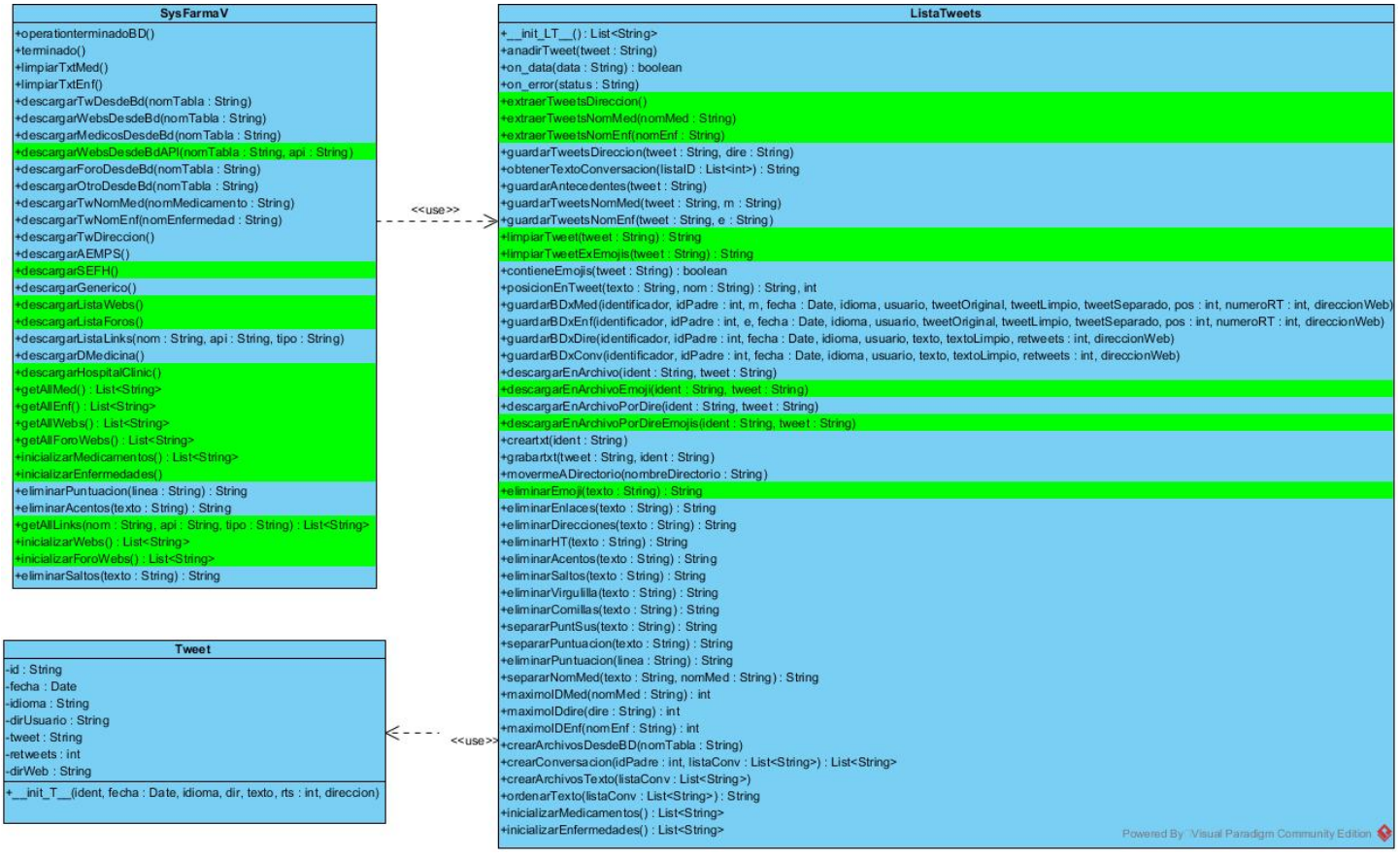


FIGURA 5.1: Diagrama de clase prototipo 1

5.2.1.2. Prototipo 2: Automatizar el procesamiento y descarga de información de páginas webs y/o foros

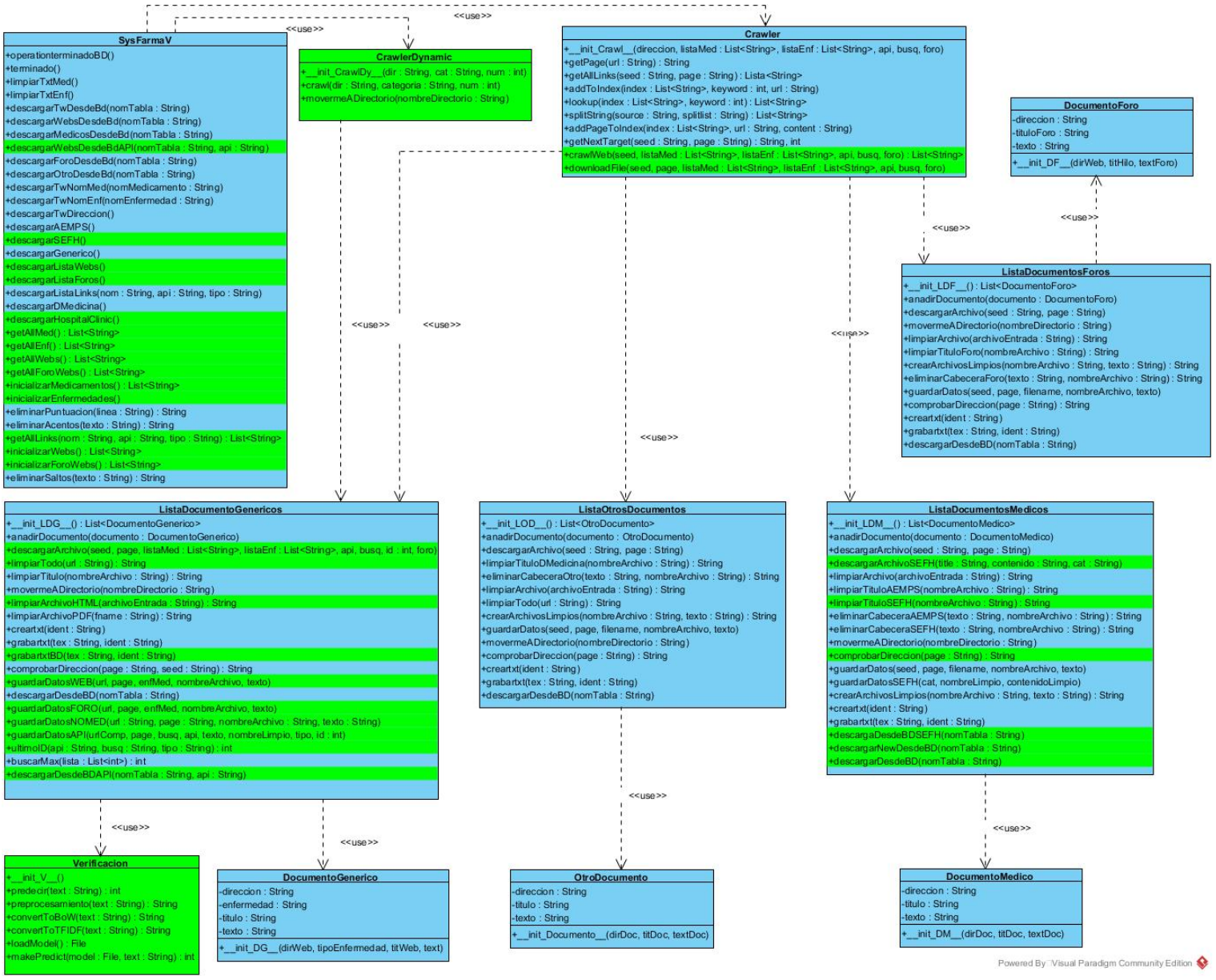


FIGURA 5.2: Diagrama de clase prototipo 2

5.2.1.3. Prototipo 3: Extraer páginas webs de APIs de motores de búsqueda

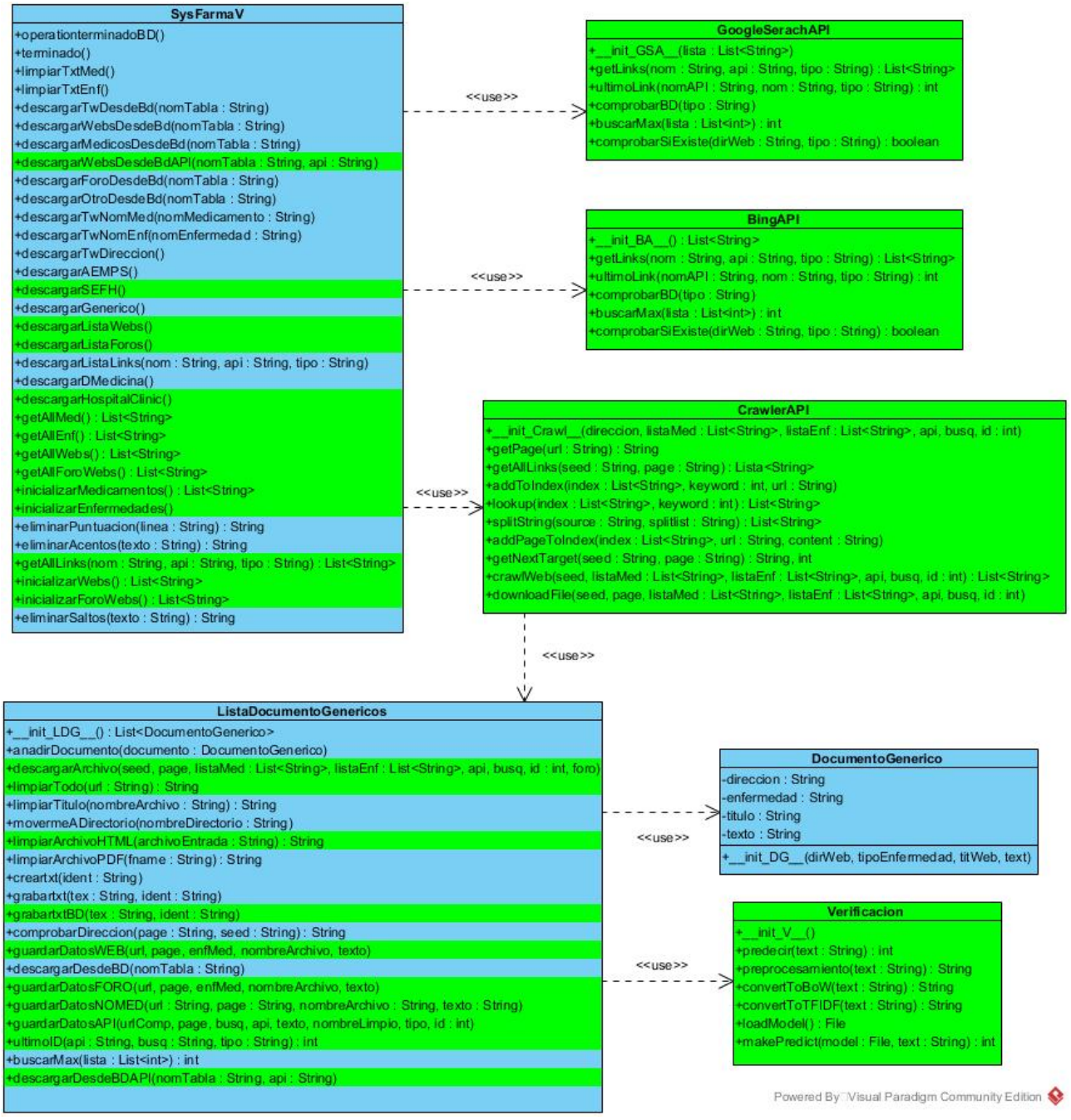


FIGURA 5.3: Diagrama de clase prototipo 3

5.2.2. Diagrama Entidad/Relación

Al mantener el diseño anterior las entidades relativas a Twitter, AEMPS, SEFH, Forum-Clinic y DMedicina no han sufrido cambios. Ya que se han tratado nuevas páginas webs, se han añadido las entidades para la nueva web de SEFH y para HospìtalClinic. Por último, se han incluido todas las entidades necesarias para alojar la información tanto médica como no que se descargue de webs genéricas.

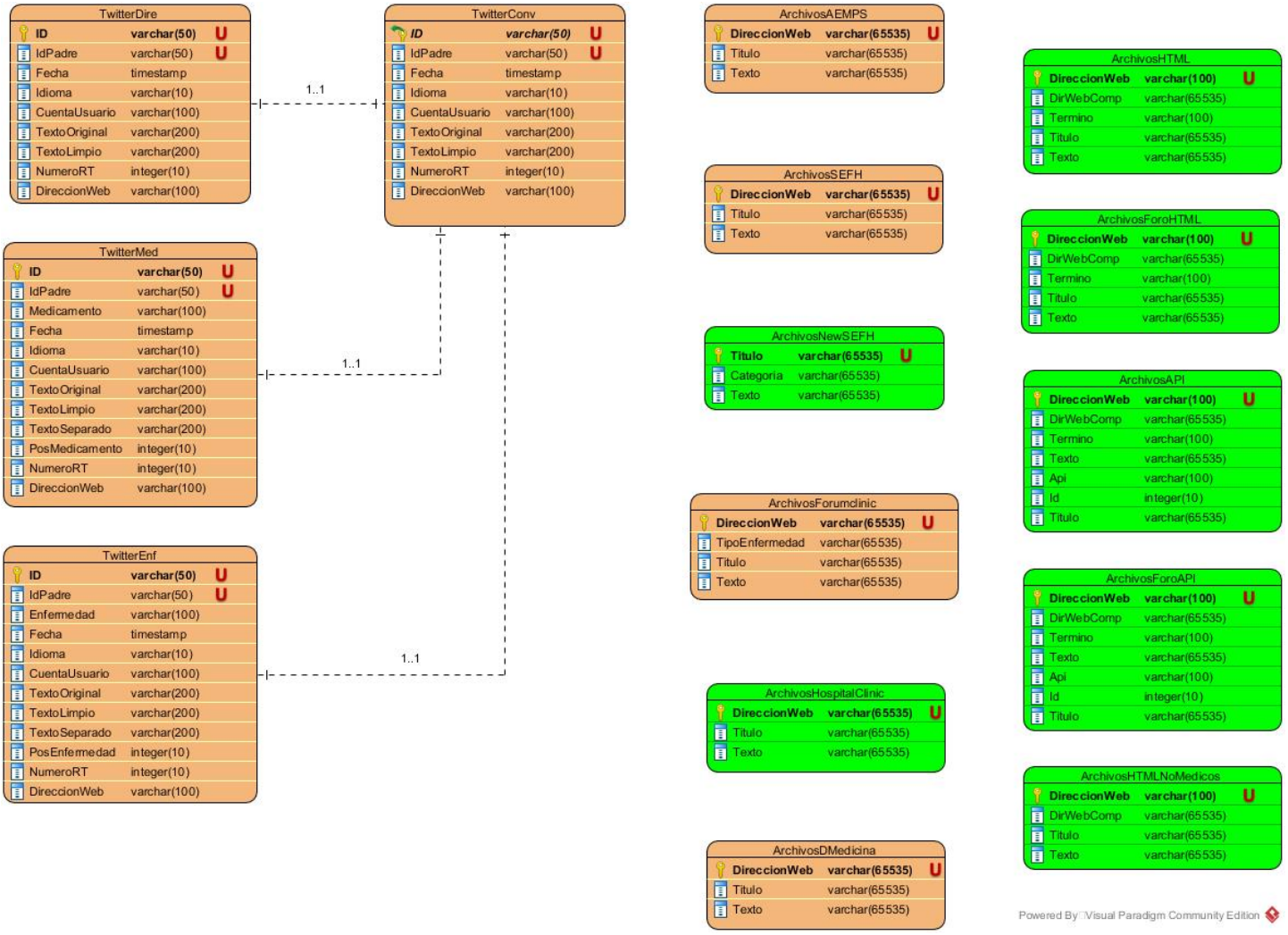


Figura 5.4: Diagrama entidad/relación

Capítulo 6

Desarrollo

6.1. Actualización del software

En un principio se comprobó si el funcionamiento del software en la actualidad era igual que cuando se implementó. Tras unas sencillas pruebas de ejecución se vió que entre el desarrollo de ambos trabajos ciertas funcionalidades dejaron de realizar la función que en su día realizaban. En concreto, la descarga de textos de la página web de la SEFH y la de ForumClinic, por lo que se procedió a actualizar ambas funciones.

6.1.1. Descarga de textos de la SEFH

El primer paso fue visitar la página web en cuestión, ya que la implementación parecía correcta y no se creía que ese fuese el problema. Tras una breve navegación se comprobó que la página web había pasado a ser dinámica, es decir, todo su contenido se alojaba en una única url, https://www.sefh.es/boletin-sefh.php. Este hecho hace que el crawler de la aplicación no encontrase enlaces para poder recorrer y extraer el contenido.

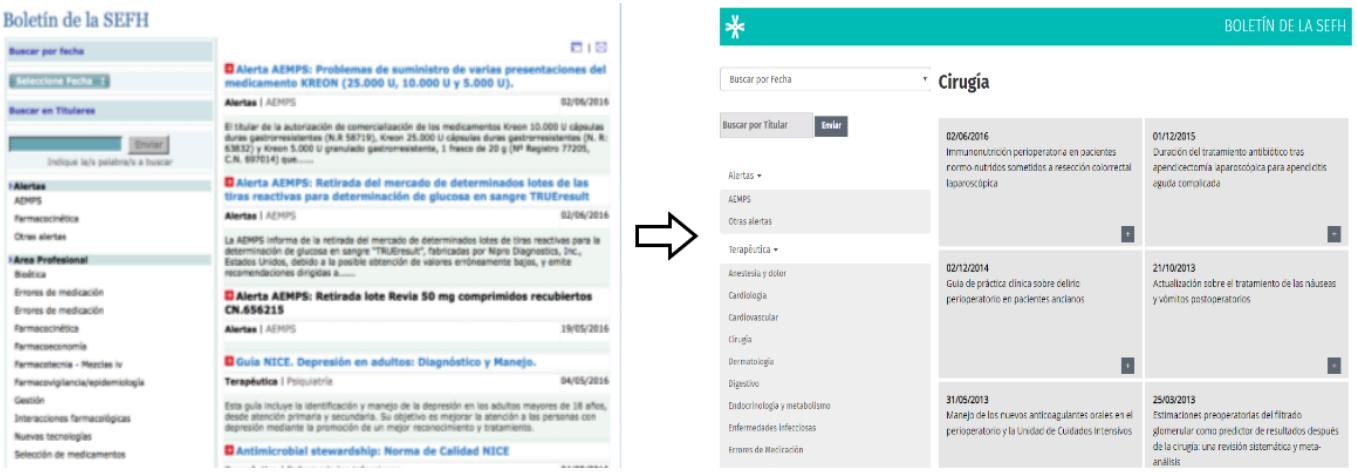


FIGURA 6.1: Cambio página web https://www.sefh.es/boletin-sefh.php

La solución tomada fue crear un crawler dinámico que fuese haciendo una navegación por los contenidos que se considerasen relevantes y extraerlos de esta forma. Para ello se encontraron dos vías, Scrapy+Splash, más profesional, y Chrome Puppetter, Quick&Dirty. Una vez investigadas un poco ambas, se escogió la solución más profesional haciendo uso de la librería Selenium aprovechando que ya se tenían conocimientos básicos sobre la misma.

El proceso nuevo consiste en una navegación automatizada por las diferentes noticias de la web que están ordenadas por categorías y se sigue el mismo procedimiento que con el contenido de otras webs.

6.1.2. Descarga de textos de ForumClinic

Para abordar este problema, con la experiencia del caso anterior, se volvió a visitar la web de la que se descargaban los textos, http://www.forumclinic.org/es/foros. Esta búsqueda en el navegador te redirigía directamente a http://www.forumclinic.org/, una clínica de salud online en francés, es decir, la propiedad del dominio había cambiado.

Investigando sobre si ForumClinic como tal había desaparecido se encontró un blog creado por la misma persona que ForumClinic. Bloc de Clínic es un blog donde se redactan noticias o hechos relevantes que se han dado en el Hospital Clinic de Barcelona.

FIGURA 6.2: Página web http://blog.hospitalclinic.org/es/

Considerando que el contenido de las entradas del blog eran relevantes y podían contener información interesante se hizo un estudio similar al anterior hecho con ForumClinic para poder extraer el contenido de esta nueva página web.

6.2. Automatizar la descarga de textos

El proceso de automatización de la descarga de textos se ha abordado en dos apartados diferentes. Por una parte se ha implementado para la descarga vía webs y foros y, por otra parte, para la descarga de textos vía Twitter.

6.2.1. Páginas webs y foros

En los módulos de webs y foros se ha creado la opción de hacer búsquedas de texto de forma automática. Pese a que ambos módulos, el de webs y foros, estén separados en la interfaz de usuario, esta funcionalidad está implementada y funciona de forma análoga en ambos. En el siguiente diagrama de flujo se puede ver el proceso que se va a seguir para la descarga de un texto de una web genérica:

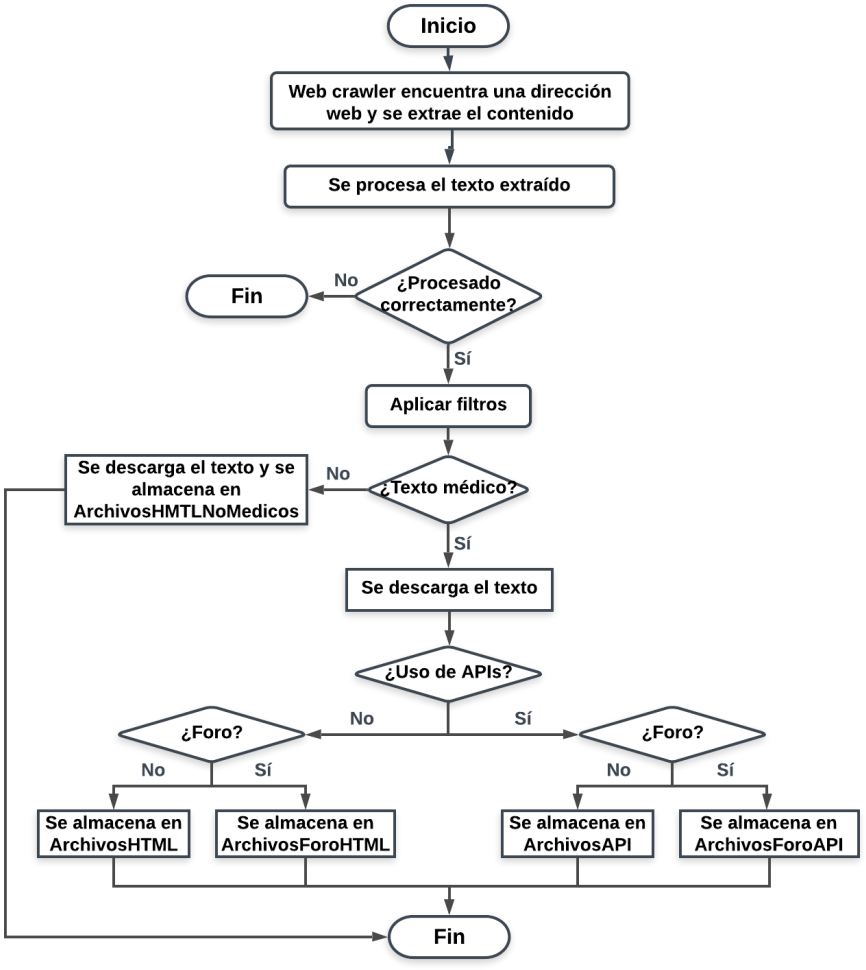


Figura 6.3: Diagrama de flujo del procesamiento, descarga y almacenamiento de un texto

Para comenzar se han seleccionado y reunido en dos ficheros de texto una lista de webs y otra de foros, cada una en un documento distinto, que se han considerado susceptibles de contener información relevante para el proyecto. Estos documentos podrán ser modificados por el usuario, por lo que cualquiera, añadiendo un link en una línea del fichero, podrá extraer datos médicos de la página que considere oportuno.

Considerando que los lugares de donde va a ser extraída la información están definidos y almacenados en el documento, cada una de las páginas webs pasará por el crawler de una en una en el orden en que estén en el fichero. Todo contenido web que el crawler haga llegar se procesará y clasificará para considerarlo relevante. Existen tres posibles casos a la hora de intentar descargar un texto:

1. **Correctamente procesado y clasificado como médico.**

   El contenido de la web será descargada y almacenada en la tabla ArchivosHTML de la base de datos. Esta tabla contiene los siguientes campos:

   - DireccionWeb: Los 80 primeros caracteres de la url de la página web. Este campo se ha incluido para no tener registros de webs con direcciones extremadamente largas.
   - DirWebComp: Dirección completa de la página web.
   - Termino: Medicamento o enfermedad que se ha hallado en el texto.
   - Titulo: Nombre que se le va a dar al documento de texto al descargar el registro.
   - Texto: Contenido descargado de la página web.

2. **Correctamente procesado y clasificado como no médico.**

   Al no ser considerado médico, el texto no es de interés para este proyecto, aun así el contenido de la web será almacenado en la tabla ArchivosHTMLNoMedicos por si en un futuro fuera de interés. Esta tabla contiene los siguientes campos:

   - DireccionWeb: Los 80 primeros caracteres de la url de la página web. Este campo se ha incluido para no tener registros de webs con direcciones extremadamente largas.
   - DirWebComp: Dirección completa de la página web.
   - Titulo: Nombre que se le va a dar al documento de texto al descargar el registro.
   - Texto: Contenido descargado de la página web.

3. **No procesado**: El contenido no se ha conseguido procesar por lo que se saltará dicho contenido y pasará al siguiente.

El hecho de que las páginas que el crawler recorre son desconocidas, no es posible preverlo, hace que no se puedan eliminar cabeceras, ya que la cabecera de cada página web es distinta, y la limpieza del contenido y el titulo del archivo sea muy genérica.

6.2.2. Twitter

En el módulo de Twitter se ha automatizado la búsqueda tanto para medicamentos como para enfermedades. Ambos casos se han desarrollado de una forma similar ya que la única diferencia entre ambas es los términos que se van a utilizar en la búsqueda.

El primer paso ha sido reunir una lista de medicamentos y enfermedades que se van a considerar relevantes para este proyecto. Para ello se han almacenado en dos documentos de texto, uno para medicamentos y otro para enfermedades, de la ontología médica SNOMED.

Una vez ya reunidos todos los términos, en la interfaz de usuario tenemos las opciones “Buscar por Lista de Medicamentos” y “Buscar por Lista de Enfermedades” en los que se realizará la búsqueda automática. Esta búsqueda consiste en, término a término, realizar una búsqueda mediante la API de Twitter la cual nos va a devolver un conjunto de tuits que incluyan ese término.

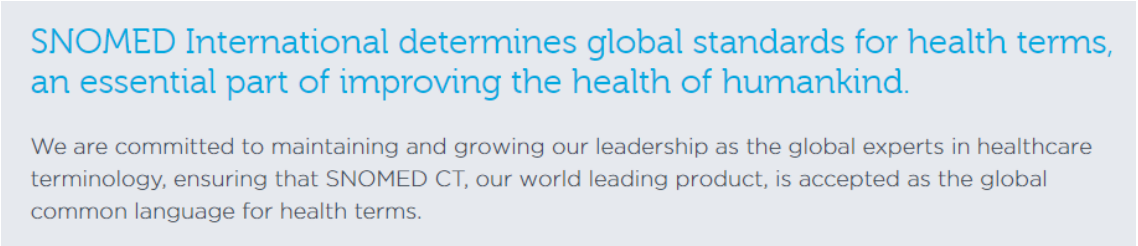


FIGURA 6.4: Definición de SNOMED por la propia entidad
Fuente: http://www.snomed.org/

La red social amplió el limite de caracteres en un tuit, por lo que en esta nueva búsqueda se han tenido en cuenta aquellos tuits que estén compuestos por más de 140 caracteres.

El conjunto de mensajes que devuelva la búsqueda serán procesados, eliminando símbolos comunes en Twitter como la almohadilla (#) o la arroba (@). En este tratamiento de los tuits se ha hecho una distinción entre aquellos que tenían emoticonos, también conocidos como *emojis*, y los que no debido a que el proyecto previo eliminaba estos caracteres especiales directamente. Los emoticonos en la actualidad se pueden definir como una representación gráfica de un estado de ánimo o de una realidad.

Para comenzar se actualizó el listado de emoticonos que proporciona Unicode ya que en este período de tiempo se han dado diversas actualizaciones en las que se han incluido nuevos caracteres. Twitter los ha incluido en su plataforma, por lo que cualquier usuario puede hacer uso de ellos en los mensajes que escriba. En este proceso se tendrá en cuenta la aparición de emoticonos y se almacenarán en directorios diferentes aquellos mensajes que los incluyan y aquellos que no. Una vez procesado los textos y terminado el proceso de búsqueda, se habrán almacenado todos los tuits que contengan algún término de la lista de enfermedades o medicamentos en la base de datos.

6.3. Recabar links de páginas webs mediante APIs de motores de búsqueda

Tal y como se ha explicado previamente en el módulo de recabar información de páginas webs y foros, el procedimiento tiene una parte semi-manual. Los hechos de conseguir los links de las páginas webs e incluir estos mismos en el documento donde se alojan la lista de webs necesariamente requieren de una persona para hacerlo. Este hecho motivó la búsqueda de un procedimiento el cual requiriese una interacción mínima del usuario con el programa.

Comúnmente los usuarios que quieren encontrar cierta información recurren a motores de búsqueda. Algunos motores de búsqueda han desarrollado sus propias APIs con las que se permite ejecutar algunas de sus funciones desde un programa informático. Google y Bing, dos de los motores de búsqueda más conocidos en la actualidad, tienen una API con la que recabar links a partir de la búsqueda de un término.

En ambos tipos de búsqueda la información descargada se ha almacenado en la base de datos creando dos tablas nueva, ArchivosAPI y ArchivosForoAPI, que tienen la siguiente estructura:

- DireccionWeb: Los 80 primeros caracteres de la url de la página web. Este campo se ha incluido para no tener registros de webs con direcciones extremadamente largas.

- DirWebComp: Dirección completa de la página web.
- Termino: Medicamento o enfermedad utilizado en la búsqueda.
- Texto: Contenido descargado de la página web.
- Api: Nombre de la herramienta con la que se ha realizado la búsqueda (googlesearch/bing).
- Id: Indica la posición en los resultados de la dirección web en la búsqueda de un término con una API determinada. Es decir, si el resultado de una búsqueda devuelve 10 direcciones web, el Id de el primer resultado es 1, el Id del segundo 2 y así sucesivamente.
- Titulo: Nombre que se le va a dar al documento de texto al descargar el registro.

Como el propósito era descargar información de diferentes fuentes, se diferenció entre los textos descargados de páginas webs genéricas y de foros. Para que la búsqueda de un término devolviese páginas de foros se añadió la postilla 'foro ' por defecto, por lo que si el término a buscar en foros es "dalsy" el parámetro 'query' finalmente será "foro dalsy". Este elemento será el indicativo para identificar si un texto se ha descargado de una web genérica o de un foro.

6.3.1. Google Search

Google es el motor de búsqueda más utilizado en el mundo, así que se buscó la forma de obtener los resultados que devolvería Google en la búsqueda de un término. La solución que se eligió fue la de utilizar el módulo *googlesearch* [20] que incluye el paquete de Google para Python. Este módulo permite hacer búsquedas personalizadas y con unos limites de peticiones muy aceptables.

Los parámetros de la búsqueda que se han utilizado para este caso son los siguientes:

- **query**: Término sobre el cual se quiere realizar la búsqueda.
- **tld**: Dominio de nivel superior. Sirve para especificar si, por ejemplo, se quiere realizar la búsqueda en google.es o google.com
- **lang**: Idioma.
- **num**: Número de resultados que se quiere obtener.
- **start**: Primer resultado a recuperar.
- **stop**: Último resultado a recuperar.
- **pause**: Tiempo de espera entre las solicitudes HTTP.

```
search(query, tld="es", lang='es', num=50, start=ult, stop=ult+50, pause=2)
```

FIGURA 6.5: Búsqueda de un término con GoogleSearch

En la figura 6.5 puede verse los valores que se han dado a los parámetros. El primero de ellos, 'query', término sobre el cual se quiere hacer la búsqueda. En los parámetros 'tld' y 'lang' tendrán valor "es" para indicar que el idioma que se pide es el español. El número de resultados a obtener por cada búsqueda se ha fijado en 50, por lo que será el valor que toma 'num'. El valor "ult" de la variable 'start' viene de la posición del último resultado de un término en concreto que está almacenado en la base de datos. Por ejemplo, si el último resultado de la búsqueda del medicamento "dalsy" registrado es el número 5, en el momento que se vuelva a buscar "dalsy" con GoogleSearch 'start' tomará valor 6 y el primer resultado será el número 6, ya que las cinco

anteriores ya se han recorrido. El valor de 'stop' será el primer resultado que se va a devolver ('start') más el número de resultados que se quiere ('num'). Por último, 'pause' tomará valor 2 (segundos). Puede ser que este tiempo de espera retrase el proceso, pero un lapso de tiempo más corto podría hacer que Google bloquease la IP al realizar demasiadas peticiones seguidas.

Tras el proceso de recabar los links de la página web viene el extraer el contenido de las mismas. Para las páginas webs que vengan de una búsqueda de una de las APIs se va a usar un crawler particular. La única diferencia entre este crawler y el que se usa para recorrer enlaces en las demás webs es la profundidad a la que el crawler accede. En este caso se ha decidido que el crawler solo llegue hasta el segundo nivel de profundidad y, así, evitar acceder a páginas webs que nada tenga que ver con la página web raíz.

6.3.2. Bing

Bing Web Search API [21] es un servicio desarrollado por Microsoft Azure que permite realizar consultas a los usuarios. La función que hace esta API es de gran utilidad para este proyecto ya que permite conseguir links de páginas webs genéricas en relación a un término.

Microsoft Azure ofrece varias alternativas gratuitas y de pago, cada una de ellas con diferentes limitaciones de uso. En este caso se ha optado por la opción gratuita que ofrece 3 transacciones por segundo y 3000 por mes gratis.

Detalles de precios de Bing Search API v7

INSTANCIA	TRANSACCIONES POR SEGUNDO (TPS)	CARACTERÍSTICAS	PRECIO
Gratis	3 TPS	Bing Image Search Bing News Search Bing Video Search Bing Visual Search Bing Web Search	3.000 transacciones gratis por mes

FIGURA 6.6: Detalles de la oferta gratuita de Bing Search API
Fuente: https://azure.microsoft.com/es-es/pricing/details/cognitive-services/search-api/

Para este proyecto, en la búsqueda de links se han especificado ciertos parámetros de consulta para que la solicitud se ajuste lo máximo posible a lo que requiere el proyecto. Los parámetros indicados en la petición tienen el siguiente significado:

- **query**: Término sobre el cual se quiere realizar la búsqueda.
- **offset**: Número de resultados que se omiten previos a los resultados. Si offset=10, los resultados que devolverá serán del décimo en adelante.
- **count**: Número de resultados que se quieren recibir.
- **mkt**: País desde el cual se realiza la petición. Este parámetro servirá para que la petición devuelva páginas webs en el idioma español.

```
web_data = client.web.search(query=query, offset=ult, count=50, mkt="es-ES")
```

FIGURA 6.7: Búsqueda de un término en Bing Search API

En la figura 6.7 se muestra que valor toman los parámetros en esta búsqueda. Para comenzar 'query' tomará el valor del término del cual se quiere hacer la búsqueda. El valor ult que se

asigna a 'offset' corresponde a la posición del último resultado de un término en concreto que está almacenado en la base de datos. Se podría decir que 'offset' cumple la misma función par Bing que 'start' para GoogleSearch. Con el objetivo de reunir una gran cantidad de información por cada búsqueda, se ha asignado el valor 50 a 'count'. Por último, el 'mkt' será "es-ES" con el propósito de que los resultados a devolver sean en castellano, aunque no lo garantiza.

Una vez se tenga el conjunto de páginas webs que nos ha devuelto la API, se pasarán al crawler específico para estas páginas webs, CrawlerAPI, para descargar la información que se vaya encontrando.

6.4. Filtrar la información procesada

Es conocido por prácticamente todo el mundo que en Internet es posible encontrar información acerca de cualquier tema. Este hecho hay que considerarlo en este proyecto, el cual solo esta enfocado en el ámbito médico, es decir, toda información que se encuentre que no tenga relación con el ámbito médico no se debe considerar relevante. Por ese motivo, toda la información que la herramienta de este proyecto procese deberá superar uno de los dos siguientes filtros para considerarla provechosa.

6.4.1. Por medicamento o enfermedad

Este primer filtro comprueba que el texto procesado incluya alguna enfermedad o medicamento que se han considerado relevantes. una vez se haga la comprobación se contemplan los siguientes dos escenarios:

- En caso afirmativo: El texto de descargará y se almacenará en la base de datos en la tabla ArchivosHTML. El contenido de la tabla ArchivosHTML será después considerada para posibles estudios futuros acerca de la medicina.
- En caso negativo: El texto de descargará y se almacenará en la base de datos en la tabla ArchivosHTMLNoMedicos. Pese a que el contenido de la tabla ArchivosHTMLNoMedicos no sirve para este proyecto, se almacenará de todas formas por un futuro próximo.

6.4.2. Por un modelo clasificador

El segundo filtro consiste en un modelo clasificador que consiga distinguir entre aquellos textos que sean médicos y los que no. La tarea de este modelo clasificador, a grandes rasgos, consiste en clasificar un texto que se le haga llegar en base a lo que haya aprendido previamente.

Este aprendizaje consiste en entrenar el clasificador con textos que se consideran médicos y otros que no. Para conseguir que un clasificador esté bien entrenado se necesitan suficientes ejemplos de ambos casos, por esta razón la primera tarea fue reunir un conjunto de datos para entrenar el clasificador. En una primera instancia se pensó en aprovechar los documentos del software anterior, ya que se podían considerar médicos y los no médicos se reunirían de la enciclopedia libre Wikipedia que, pese a que nadie asegura que la información que tenga sera totalmente correcta y veraz, se va a considerar una fuente válida. Este primer intento no resultó efectivo ya que entre los textos médicos se encontraban textos no muy apropiados como tuits.

Viendo que esta opción no surgió el efecto esperado se decidió reunir los dos conjuntos de textos, médicos y no médicos, de la propia Wikipedia [22]. Por una parte, se descargó aquellos

artículos de la Wikipedia cuyos títulos aparecen en SNOMED CT [23], que serán los que se consideren médicos. Por otro lado, se siguió el mismo procedimiento para artículos con un título que no aparcen en SNOMED CT, considerados no médicos.

A continuación, se crea el clasificador entrenándolo con el conjunto de datos reunido. El clasificador a implementar es el Random Forest [24], que es un algoritmo que combina árboles de decisión lo más básicos posibles para, a partir de estos, llegar a árboles más complejos.

Las razones principales para elegir Random Forest son:

- Entre los algoritmos actuales es uno de los que mejor precisión reporta.
- Funciona de manera eficiente en grandes bases de datos

A la hora de hacer las predicciones, las hojas de cada árbol son etiquetadas con una clasificación y aquella etiqueta mayoritaria será la clase reportada como la predicción.

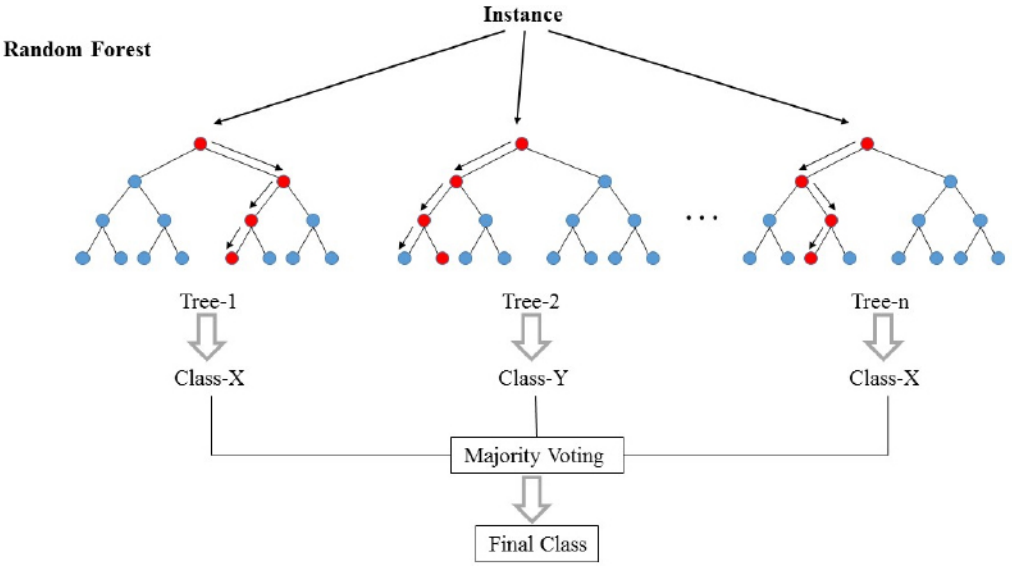


Figura 6.8: Representación gráfica de Random Forest
Fuente: http://www.nrronline.org/article.asp?issn=1673-5374;year=2018;volume=13;issue=6;spage=962;epage=970;aulast=Dimitriadis

El proceso de filtrado consiste en que el clasificador haga predicciones sobre los textos ya procesados y limpios que se le hagan llegar desde la herramienta. El clasificador realiza la predicción y en función del resultado se contemplará los mismos escenarios que el filtrado por medicamento o enfermedad.

Capítulo 7

Verificación y evaluación

7.1. Verificación

En este apartado se detallará las pruebas realizadas prototipo por prototipo para comprobar el correcto funcionamiento de la herramienta completa. Por cada prueba se especificará tanto qué resultado se debe esperar de una prueba como el resultado real.

7.1.1. Prototipo 1

7.1.1.1. Procesar y descargar tuits de un medicamento o enfermedad

- **Resultado esperado**: La búsqueda del término en la API de Twitter devolverá un conjunto de tuits que se procesarán y descargarán en un directorio en formato de texto plano.
- **Resultado real**: Si se encuentran tuits con ese término, se intentan procesar. En caso de conseguirlo, el tuit se descarga y se deposita en el directorio correspondiente; en caso contrario, el tuit se descarta y se continua el proceso.

  En el caso de llegar al límite de tuits descargados por la API en un periodo de tiempo, el proceso se detendrá y reanudará su marcha una vez se habilite la descarga de nuevo.

7.1.1.2. Procesar y descargar tuits de una lista un medicamentos o enfermedades

- **Resultado esperado**: Se hace una búsqueda por cada término de la lista de medicamentos o enfermedades. Esto devolverá un conjunto de tuits, uno por cada término de la lista, que se procesarán y descargarán en un directorio en un texto plano cada uno.
- **Resultado real**: Si se encuentran tuits con ese término, se intentan procesar. En este caso es más complicado encontrar tuits de ciertos medicamentos o enfermedades ya que algunos de ellos tienen nombres no muy comunes y fáciles de escribir por lo que no es fácil encontrar a usuarios que escriban los nombres de forma correcta.

  Si un tuit es procesado, se descarga y se deposita en el directorio correspondiente, sino se descarta y se continua con el siguiente tuit. Por lo general, en esta funcionalidad siempre se llega al límite de descarga, lo que hace que el proceso sea lento y temporalmente costoso.

7.1.1.3. Procesar y descargar tuits de usuarios seguidos por una cuenta

- **Resultado esperado**: Se recopila los tuits de los usuarios seguidos por @TFGFVIGILANCIA. Esto devolverá un conjunto de tuits que se procesarán y descargarán en un directorio en un texto plano cada uno.
- **Resultado real**: El conjunto de tuits se procesa y se descarga en el directorio correspondiente.

7.1.1.4. Almacenamiento de los tuits procesados y descargados

- **Resultado esperado**: Un tuit que finalmente se consigue descargar es almacenado en la base de datos.
- **Resultado real**: Todo tuit procesado y descargado se almacena en la tabla correspondiente de la base de datos.

7.1.1.5. Distinguir entre tuits con y sin emoticonos

- **Resultado esperado**: Aquellos tuits con emoticonos y sin emoticonos se descargarán en directorios diferentes.
- **Resultado real**: Se identifican los mensajes con y sin emoticonos y se descargan en el directorio correspondiente.

7.1.2. Prototipo 2

7.1.2.1. Procesar y descargar textos de páginas predefinidas (AEMPS, HospitalClinic y DMedicina

- **Resultado esperado**: Se descargan los textos utilizando el *web crawler* de cada una de las páginas que se consideran relevantes y se depositan en su directorio correspondiente.
- **Resultado real**: Los textos procesados y considerados relevantes de los enlaces recorridos por el *web crawler* son descargados en el directorio correspondiente.

7.1.2.2. Procesar y descargar textos de página predefinida dinámica (SEFH)

- **Resultado esperado**: Se descargan los textos de la página dinámica y se depositan en su directorio correspondiente.
- **Resultado real**: Se accede a los textos mediante navegación automatizada y se descargan. Hay que considerar el inconveniente de la lentitud de la funcionalidad. El proceso es más lento que el de las otras páginas predefinidas debido al tiempo empleado en la navegación por la página web y técnica de extracción de textos.

7.1.2.3. Procesar y descargar textos de listas de webs y foros

- **Resultado esperado**: Se procesan y descargan textos de páginas webs genéricas almacenadas en un documento de texto.
- **Resultado real**: Aquellos textos que se consiguen procesar y superen el proceso de filtrado, se descargan y se depositan en su directorio correspondiente.

  El hecho de que el *web crawler* recorra páginas weba tan diversas hace que el procesamiento tenga que ser muy genérico y que un número considerable de textos no consigan ser tratados correctamente, lo que hace que la cantidad de información descargada disminuya.

7.1.2.4. Almacenamiento de los textos procesados y descargados

- **Resultado esperado**: Un texto que finalmente se consigue descargar es almacenado en la base de datos.
- **Resultado real**: Todo texto procesado y descargado se almacena en la tabla correspondiente de la base de datos.

7.1.3. Prototipo 3

7.1.3.1. Procesar y descargar textos de webs y foros con GoogleSearch

- **Resultado esperado**: Se recopila una lista de links realizando una búsqueda de un término con GoogleSearch y se descargan los textos de las páginas de la lista.
- **Resultado real**: Se procesan y descargan, si superan el proceso de filtrado, en el directorio correspondiente los textos de las páginas webs devueltas por la búsqueda.

  Es un proceso relativamente rápido pese al tiempo de espera asignado para que Google no bloquee la IP. Por otra parte, al ser webs genéricas, el procesamiento sufre las mismas dificultades que con la lista de webs y foros.

7.1.3.2. Procesar y descargar textos de webs y foros con Bing

- **Resultado esperado**: Se recopila una lista de links realizando una búsqueda de un término con la API de Bing y se descargan los textos de las páginas de la lista.
- **Resultado real**: Se procesan y descargan, si superan el proceso de filtrado, en el directorio correspondiente los textos de las páginas webs devueltas por la búsqueda.

  Esta funcionalidad tiene la particularidad el límite que Bing establece para una cuenta gratuita a la hora de realizar búsquedas, por lo que solo se podrán hacer 3000 transacciones al mes.

7.1.3.3. Almacenamiento de los textos procesados y descargados

- **Resultado esperado**: Un texto que finalmente se consigue descargar es almacenado en la base de datos.
- **Resultado real**: Todo texto procesado y descargado se almacena en la tabla correspondiente de la base de datos distinguiendo en cada caso el motor de búsqueda con el que se han encontrado.

7.2. Evaluación

La evaluación del proyecto comienza seleccionando 20 textos descargados de las diferentes fuentes que ofrece la herramienta aleatoriamente. Estos textos han sido etiquetados como *medico* o *nomedico* por las dos directoras del trabajo de forma independiente.

Tras el proceso de etiquetado se calcula el *inter-tagger agreement*, que mide el acuerdo entre las etiquetas asignadas. Del total de 20 textos, en 15 de ellos hubo coincidencia a la hora de etiquetar por parte de las directoras. De este hecho se concluye que el *inter-tagger agreement* es del 75 %, o lo que es lo mismo, la capacidad máxima que tendrá la herramienta de distinguir correctamente textos.

Al haber discrepancias entre las dos etiquetadoras, los textos 5 en debate se clasifican de un lado u otro para crear lo conocido como *gold standard*. El *gold standard* [25] se considera aquella prueba ejemplar en términos de calidad o corrección con la que se compararán los resultados obtenidos. Finalmente, el *gold standard* contiene 16 textos etiquetados como *medico* y 4 como *nomedico*. A continuación, se realizan las predicciones con el modelo clasificador creado. La comparación entre los resultados de las predicciones con el *gold standard* se representa en la siguiente matriz de confusión:

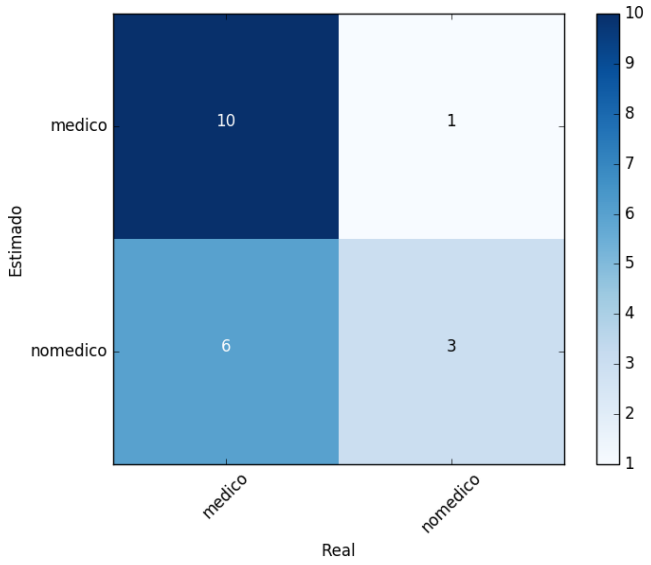


Figura 7.1: Matriz de confusión
Fuente código: https://www.kaggle.com/grfiv4/plot-a-confusion-matrix

Por último, se elabora un informe donde se calculan las figuras de mérito (*precision*, *recall* y *f-score*) que determinan la calidad del modelo clasificador.

Informe de clasificación				
	Precision	Recall	F-Score	Cantidad
medico	0.91	0.62	0.74	16
nomedico	0.33	0.75	0.46	4
Avg / Total	0.79	0.65	0.68	20

Tabla 7.1: Informe de clasificación

Capítulo 8

Conclusiones y trabajo futuro

8.1. Trabajo futuro

El trabajo futuro más obvio es dar utilidad a los textos médicos que la herramienta consigue descargar de internet. Existen cantidad de trabajos o estudios que requieren de grandes volúmenes de información. Únicamente habría que encontrar uno relativo al tema del que trata este trabajo como, por ejemplo, la detección de efectos adversos de medicamentos, tal y como se realizo en el trabajo antecesor. Aprovechando que el software es capaz de distinguir entre tuits con y sin emoticonos, uno de los estudios interesantes podría ir orientado a realizar un *Sentimental Analysis* de los mensajes de los usuarios de Twitter acerca de un medicamento.

Otra de las posibilidades sería mejorar el procesamiento de textos ya que, tal y como se ha dicho previamente, la herramienta no consigue procesar correctamente ciertos contenidos de páginas webs. Por otro lado, se podrían aplicar mejoras en la descarga de textos buscando alternativas a las funcionalidades que están implementadas que ofrezcan menos limitaciones y tiempos de espera en los apartados de Twitter y las APIs.

En cuanto a la interfaz de usuario podría desarrollarse una más atractiva visualmente aportándole un aire más moderno, más interactivo con el usuario, etc. ya que la actual se podría considerar una interfaz básica, pese a que cumple su función principal.

8.2. Conclusiones sobre el trabajo

Como comienzo, manifestar que los objetivos del proyecto han sido cumplidos. Se ha actualizado el software que me llegó en un inicio, se han automatizado los procesos de descarga de textos y se ha conseguido desarrollar una verificación de textos médicos. Por contra, remarcar las limitaciones de la herramienta a la hora de descargas información, el coste temporal grande de algunas funcionalidades y el procesamiento de textos no óptimo.

Un aspecto a remarcar es la mala gestión de los tiempos del trabajo, ya que dificultades a la hora de implementar ciertas funcionalidades retrasaron el tiempo esperado de finalización del proyecto. Esto ha hecho que la presión por terminar a tiempo el trabajo aumentase. Una mejor planificación temporal o búsquedas de alternativas habría sido más adecuado.

8.3. Conclusiones personales

Una vez ya concluido el trabajo, la sensación general es satisfacción tras completar los objetivos principales del TFG y ver que el esfuerzo y tiempo invertidos han dado sus frutos.

Durante el desarrollo de este trabajo de fin de grado me he enfrentado a multitud de dificultades: elaborar un trabajo de estas características de forma individual, tener que interpretar y entender el software previo (qué hace, cómo lo hace, por qué lo hace de esa manera), nuevos lenguajes de programación, nuevos áreas de investigación, etc. Tras haber abordado todos ellos, opino que los he conseguido lidiar de forma medianamente satisfactoria.

El conocimiento adquirido estos meses en cuestiones como el lenguaje Python, desarrollo completo del proyecto o los ámbitos de los rastreadores web y la minería de datos estoy seguro que será beneficiosa para mi futuro. Por último, únicamente expresar el deseo de que el proyecto sea de utilidad en algún proyecto o investigación futura.

Apéndice A

Anexo I. Casos de uso extendidos y diagramas de secuencia

Descargar textos AEMPS

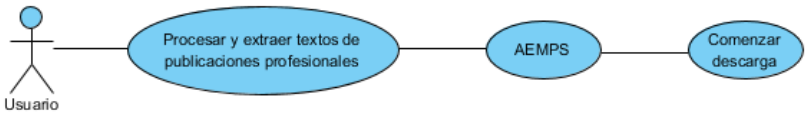


Figura A.1: Caso de uso extendido: Descargar textos AEMPS

Procesar, descargar y almacenar los textos desde la página web de la AEMPS	
Descripción	Procesa, descarga y almacena el contenido de la página web http://www.aemps.gob.es.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto.
Poscondiciones	La tabla ArchivosAEMPS de la base de datos se ha actualizado y se ha descargado el contenido de la página web.

Tabla A.1: Descargar textos AEMPS

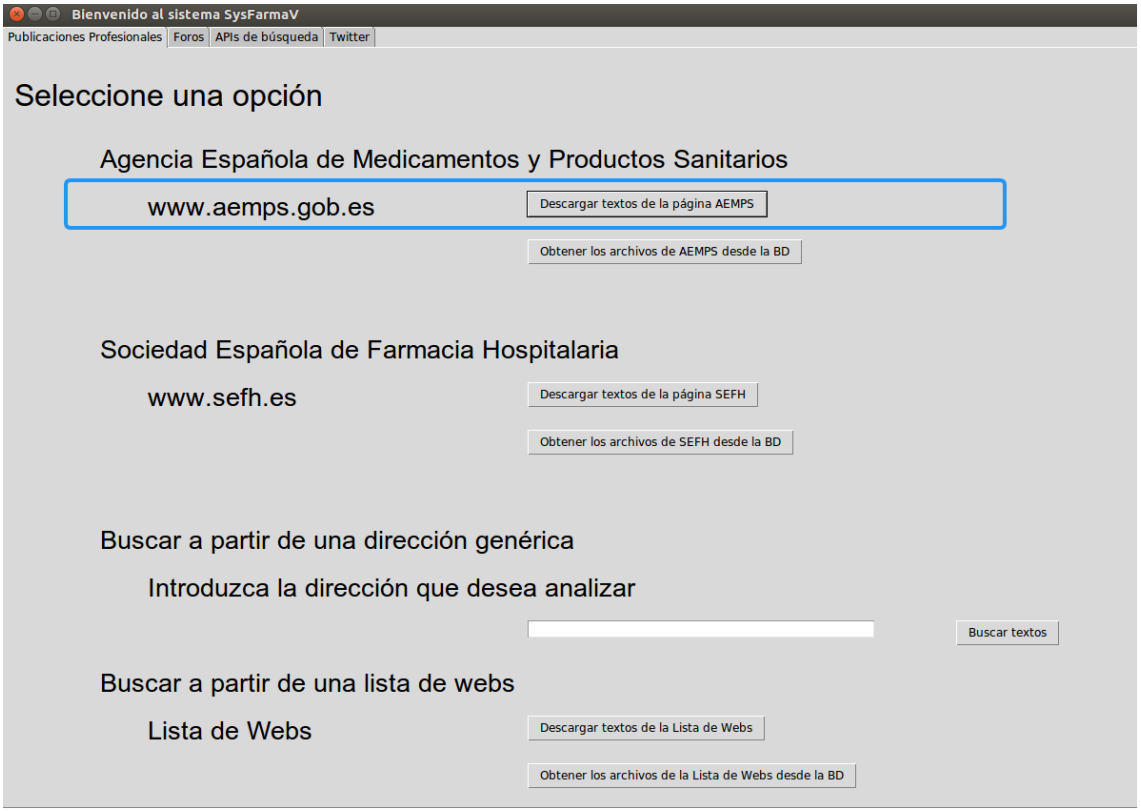


FIGURA A.2: Interfaz gráfica: Descargar textos AEMPS

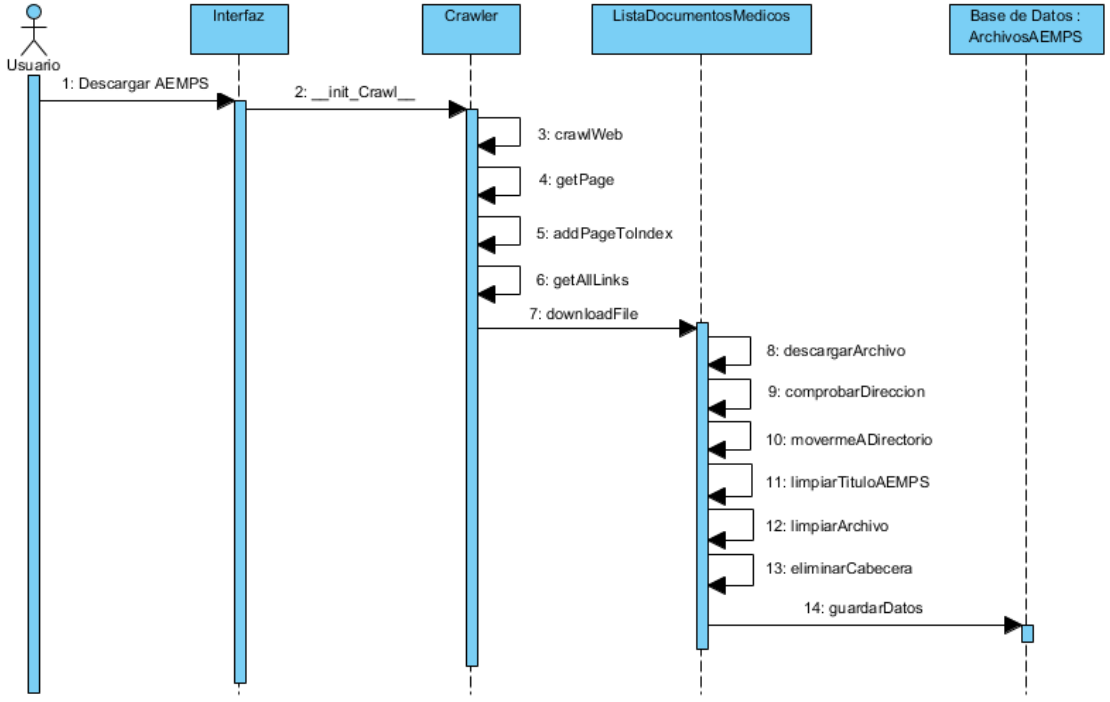


FIGURA A.3: Diagrama de secuencia: Descargar textos AEMPS

Descargar textos AEMPS desde la base de datos

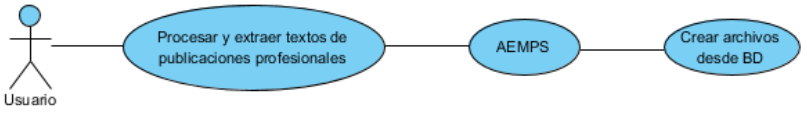


FIGURA A.4: Caso de uso extendido: Descargar textos AEMPS desde la base de datos

Convertir en documentos de texto la información de AEMPS de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de la página web http://www.aemps.gob.es de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosAEMPS contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosAEMPS y los registros de la tabla ArchivosAEMPS se pasan a documentos de texto.

TABLA A.2: Descargar textos AEMPS desde la base de datos

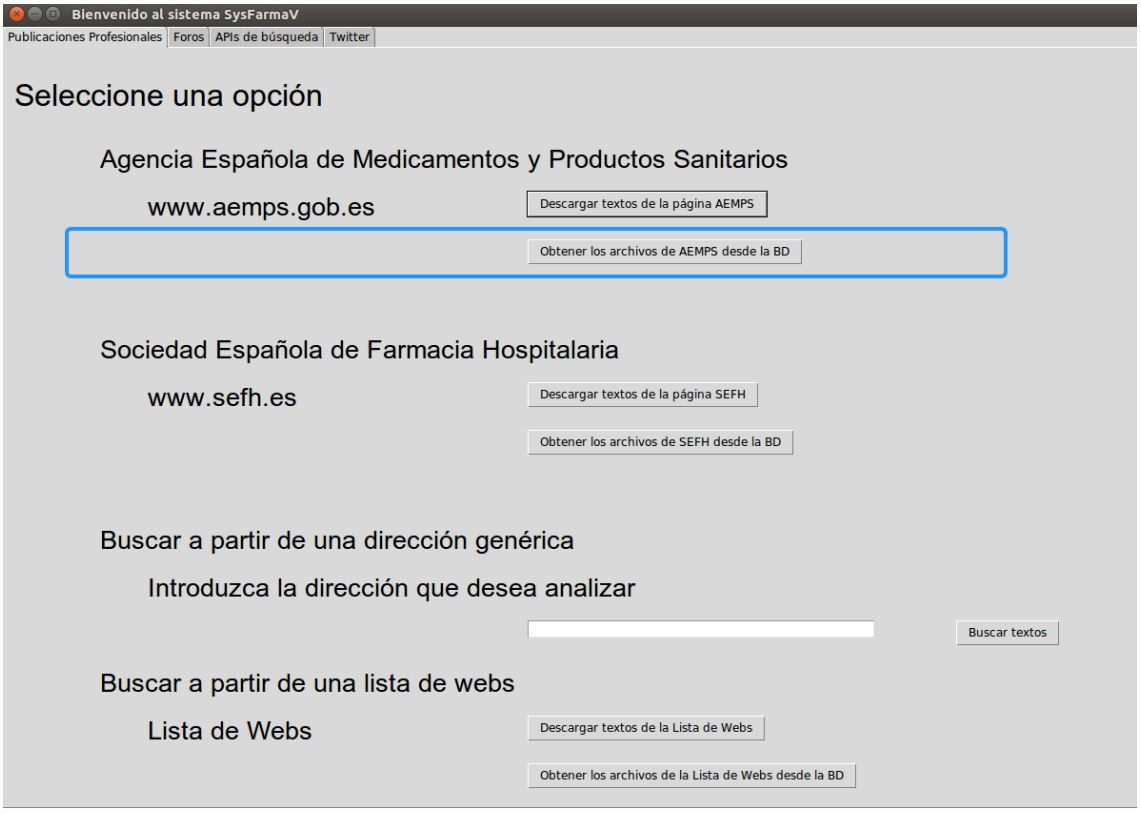


FIGURA A.5: Interfaz gráfica: Descargar textos AEMPS desde la base de datos

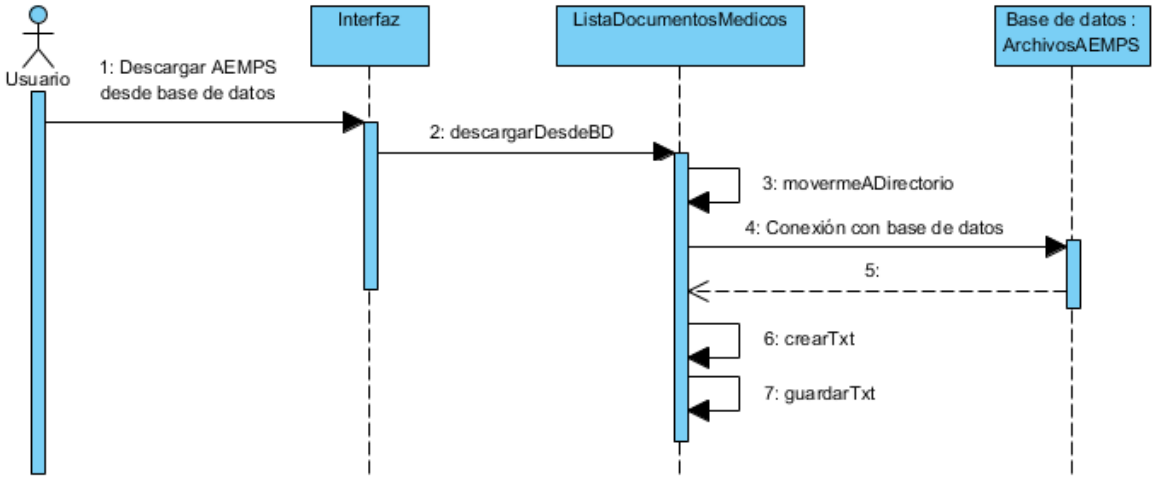


FIGURA A.6: Diagrama de secuencia: Descargar textos AEMPS desde la base de datos

Descargar textos SEFH

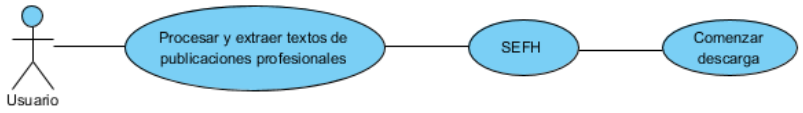


FIGURA A.7: Caso de uso extendido: Descargar textos SEFH

Procesar, descargar y almacenar los textos desde la página web de la SEFH	
Descripción	Procesa, descarga y almacena el contenido de la página web https://www.sefh.es/boletin-sefh.php.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto.
Poscondiciones	La tabla ArchivosSEFH de la base de datos se ha actualizado y se ha descargado el contenido de la página web.

TABLA A.3: Descargar textos SEFH

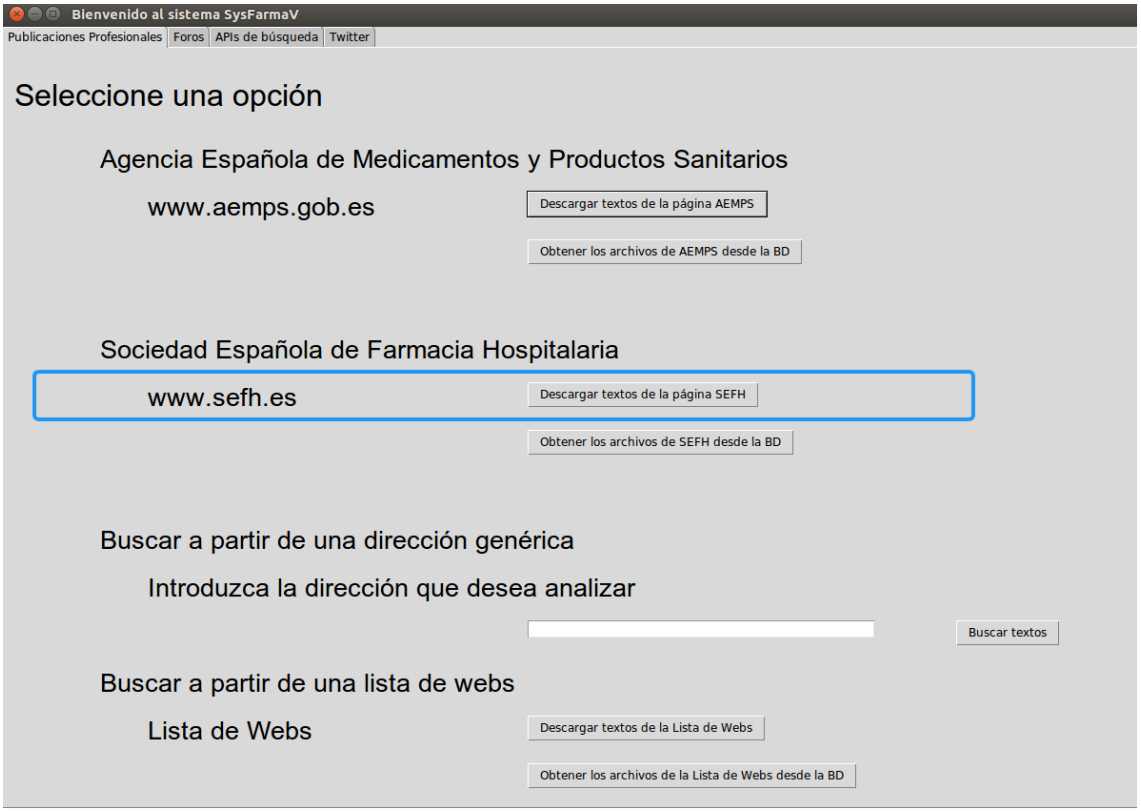


FIGURA A.8: Interfaz gráfica: Descargar textos SEFH

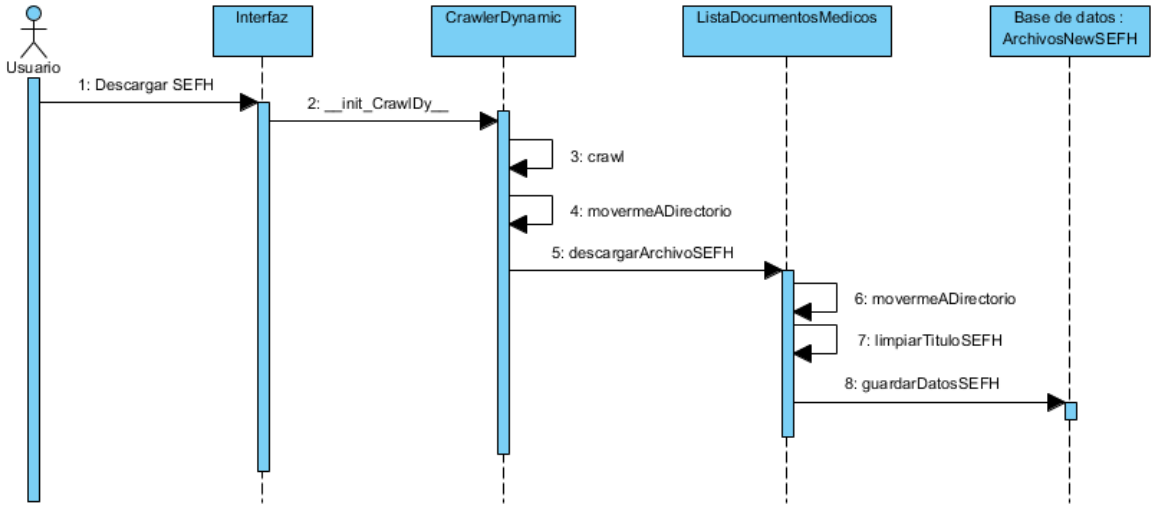


FIGURA A.9: Diagrama de secuencia: Descargar textos SEFH

Descargar textos SEFH desde la base de datos

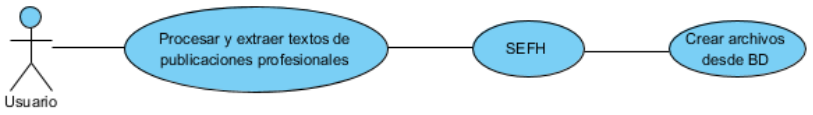


FIGURA A.10: Caso de uso extendido: Descargar textos SEFH desde la base de datos

Convertir en documentos de texto la información de SEFH de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de la página web https://www.sefh.es/boletin-sefh.php de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosSEFH contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosSEFH y los registros de la tabla ArchivosSEFH se pasan a documentos de texto.

TABLA A.4: Descargar textos SEFH desde la base de datos

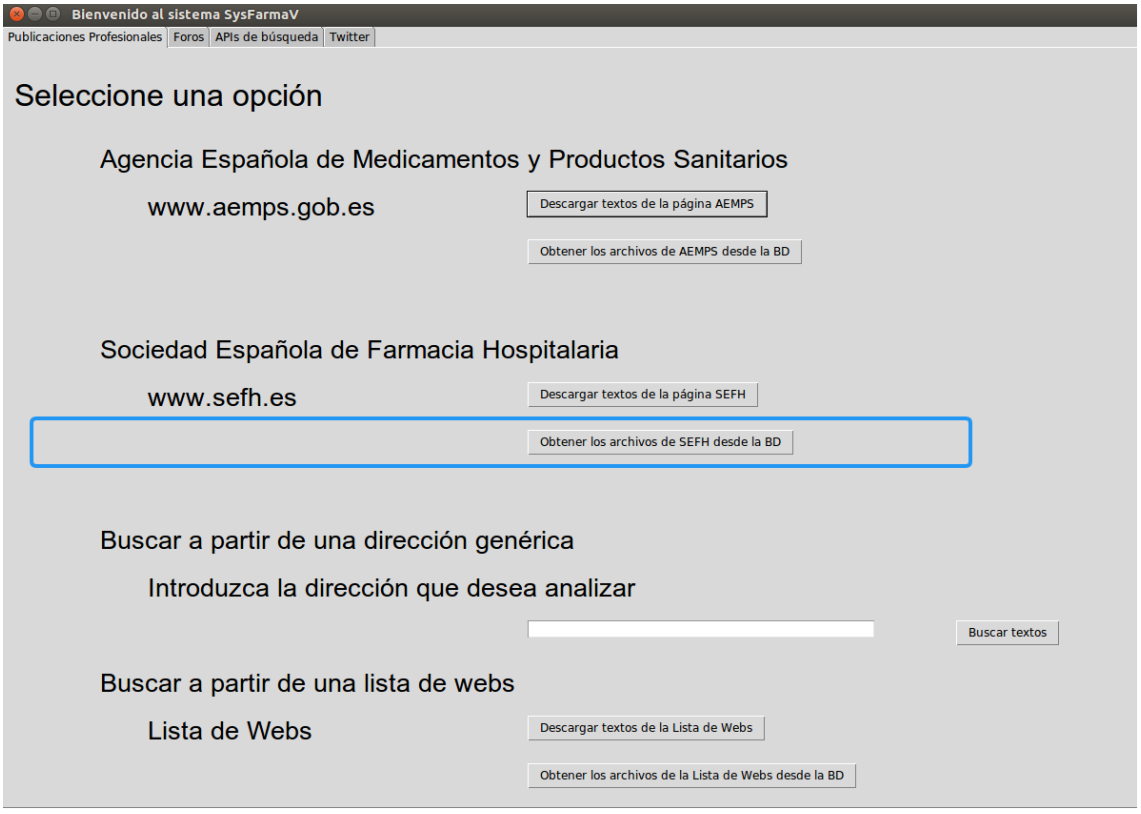


FIGURA A.11: Interfaz gráfica: Descargar textos SEFH desde la base de datos

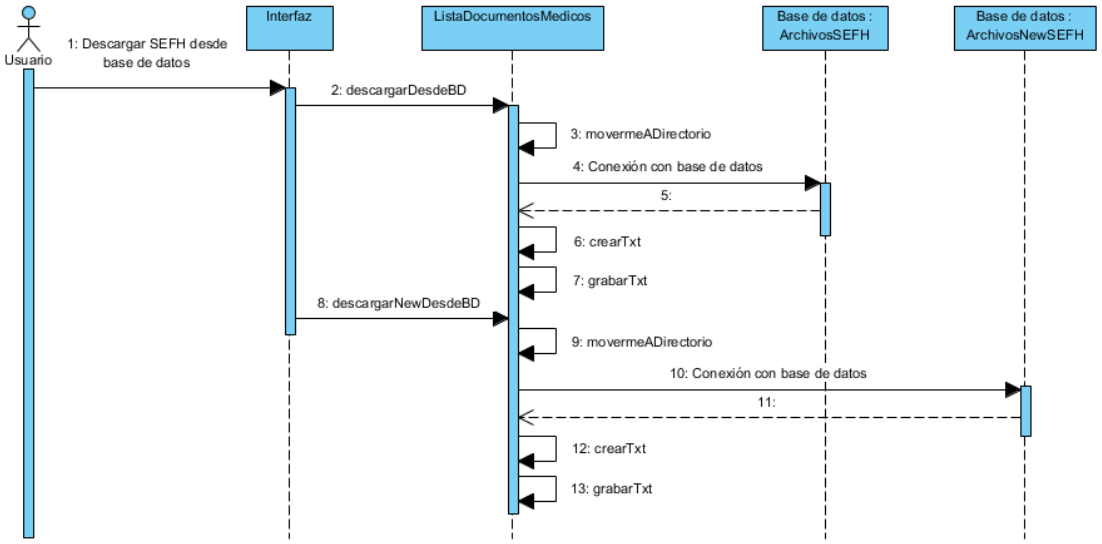


FIGURA A.12: Diagrama de secuencia: Descargar textos SEFH desde la base de datos

Descargar textos de una web genérica

FIGURA A.13: Caso de uso extendido: Descargar textos de una web genérica

Procesar, descargar y almacenar los textos de una web genérica	
Descripción	Procesa, descarga y almacena el contenido de la página web que el usuario introduzca.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, la página web que el usuario ha introducido, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tabla ArchivosHTML y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de la página web.

TABLA A.5: Descargar textos de una web genérica

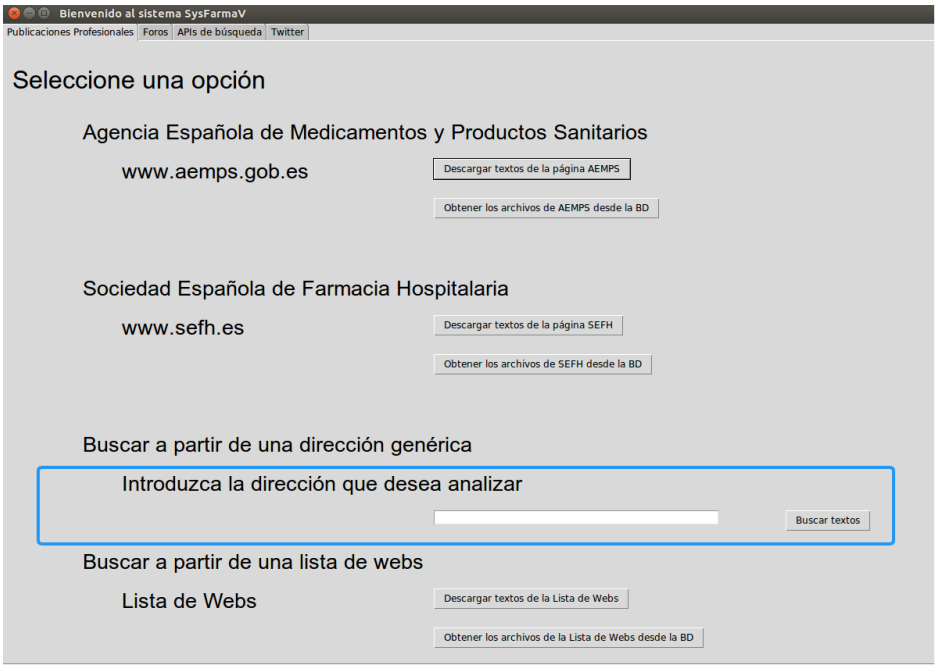


FIGURA A.14: Interfaz gráfica: Descargar textos de una web genérica

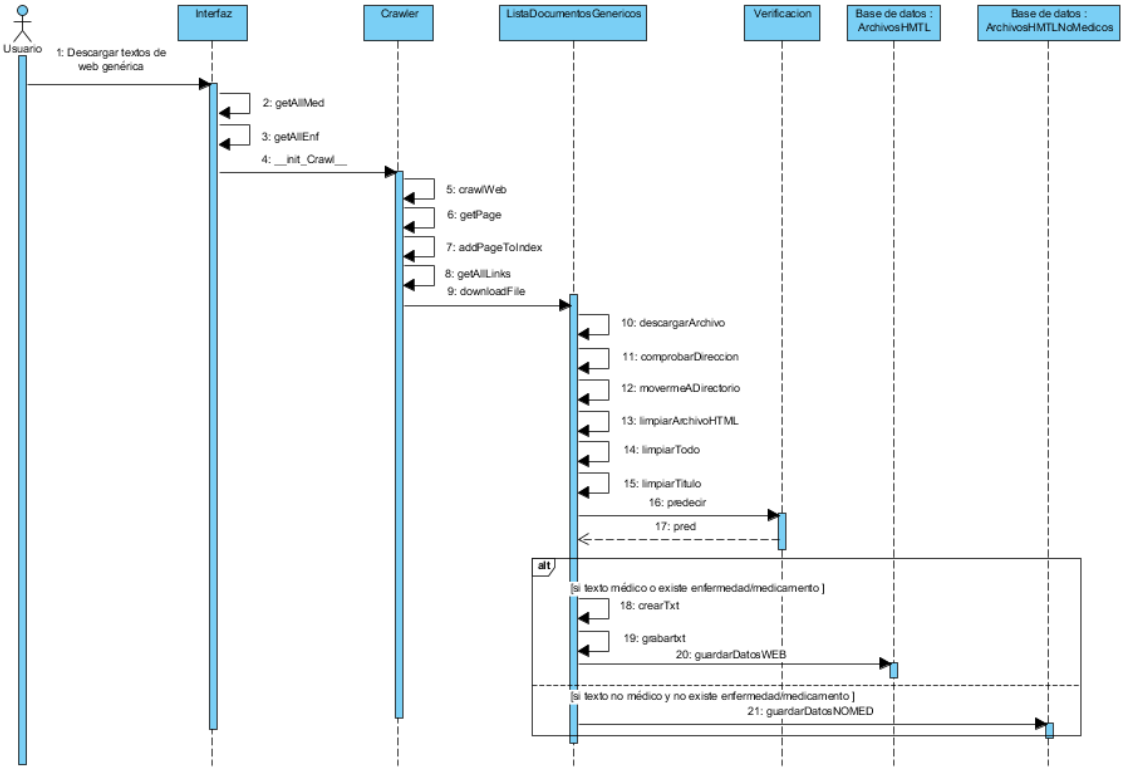


FIGURA A.15: Diagrama de secuencia: Descargar textos de una web genérica

Descargar textos de una lista de páginas webs

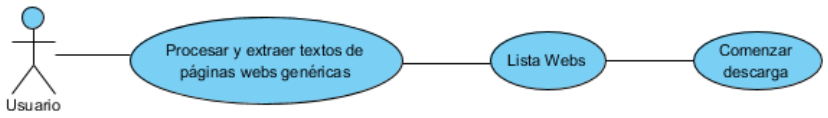


FIGURA A.16: Caso de uso extendido: Descargar textos de una lista de páginas webs

Procesar, descargar y almacenar los textos de un conjunto de páginas webs	
Descripción	Procesa, descarga y almacena el contenido de la lista de páginas webs que contiene el archivo ListaWebs.txt
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, las páginas webs que contiene el archivo ListaWebs.txt, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tabla ArchivosHTML y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de las páginas webs.

TABLA A.6: Descargar textos de una lista de páginas webs

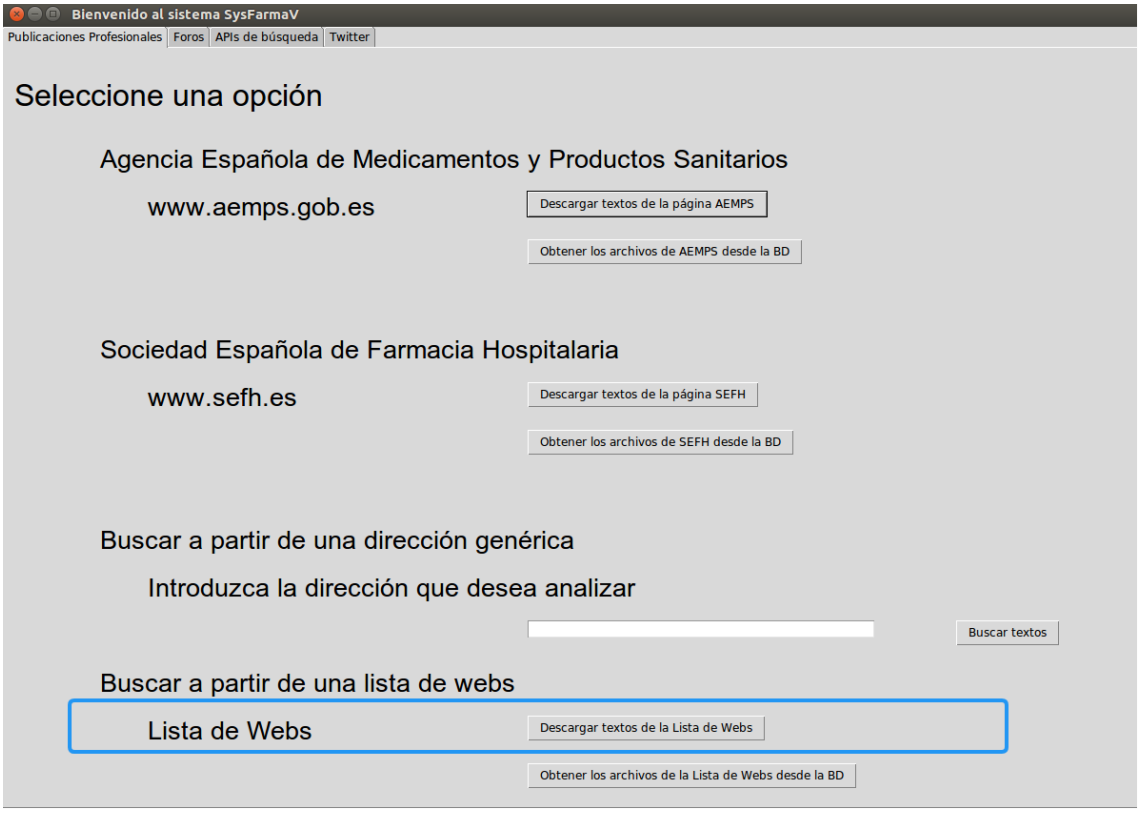


FIGURA A.17: Interfaz gráfica: Descargar textos de una lista de páginas webs

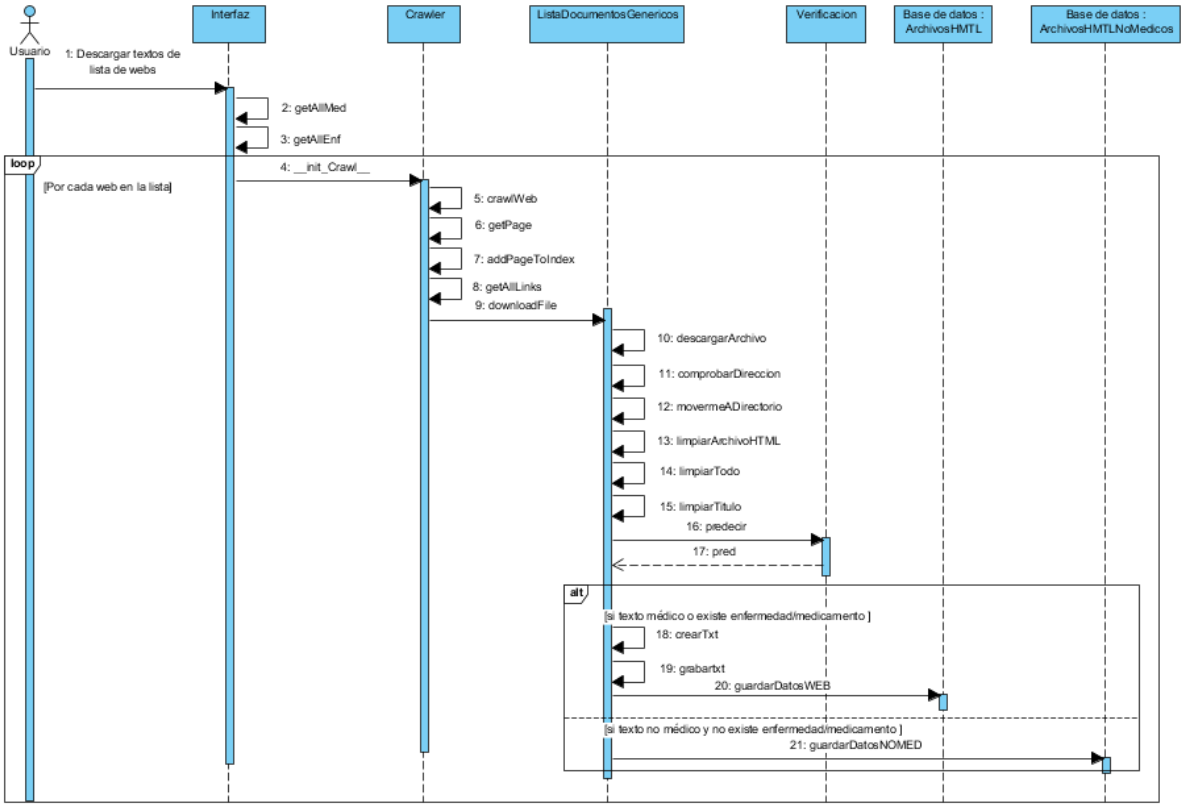


FIGURA A.18: Diagrama de secuencia: Descargar textos de una lista de páginas webs

Descargar textos genéricos desde la base de datos

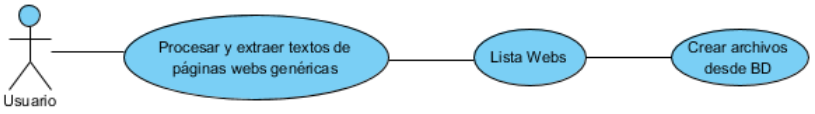


FIGURA A.19: Caso de uso extendido: Descargar textos genéricos desde la base de datos

Convertir en documentos de texto la información de webs genéricas de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de las páginas webs genéricas de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosHTML contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosHTML y los registros de la tabla ArchivosHTML se pasan a documentos de texto.

TABLA A.7: Descargar textos genéricos desde la base de datos

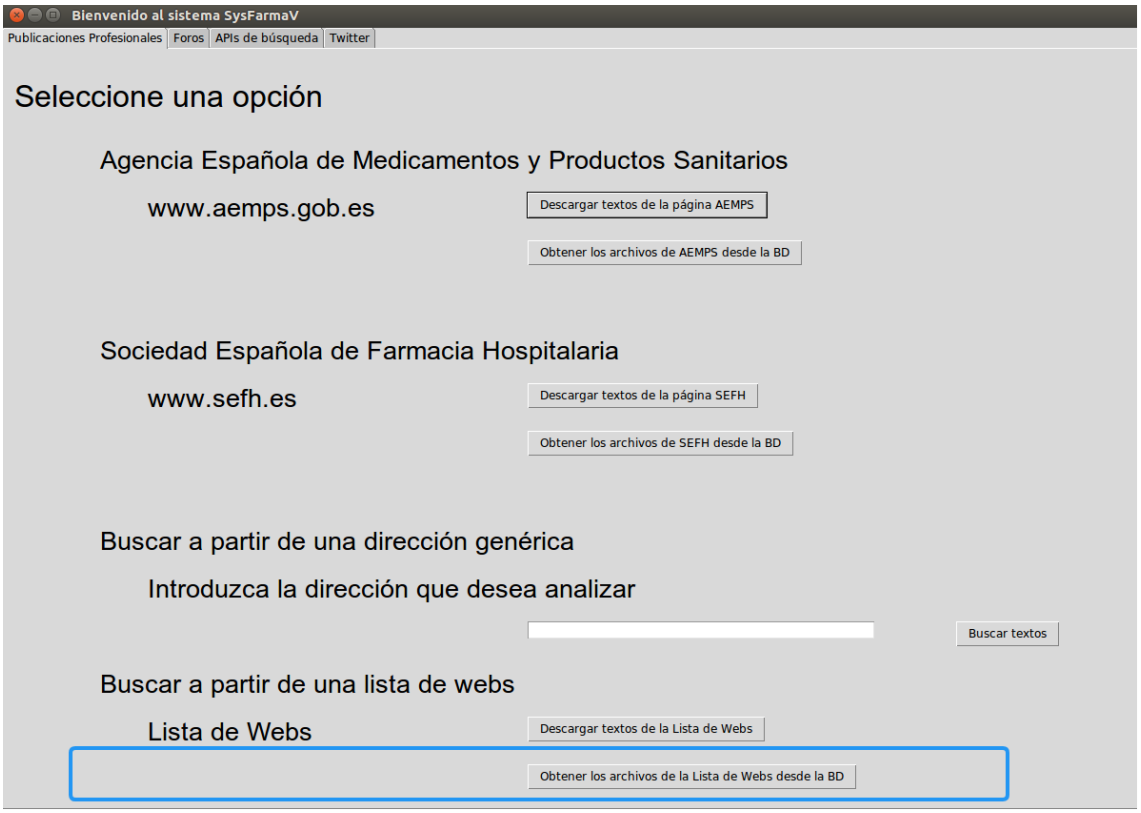


FIGURA A.20: Interfaz gráfica: Descargar textos genéricos desde la base de datos

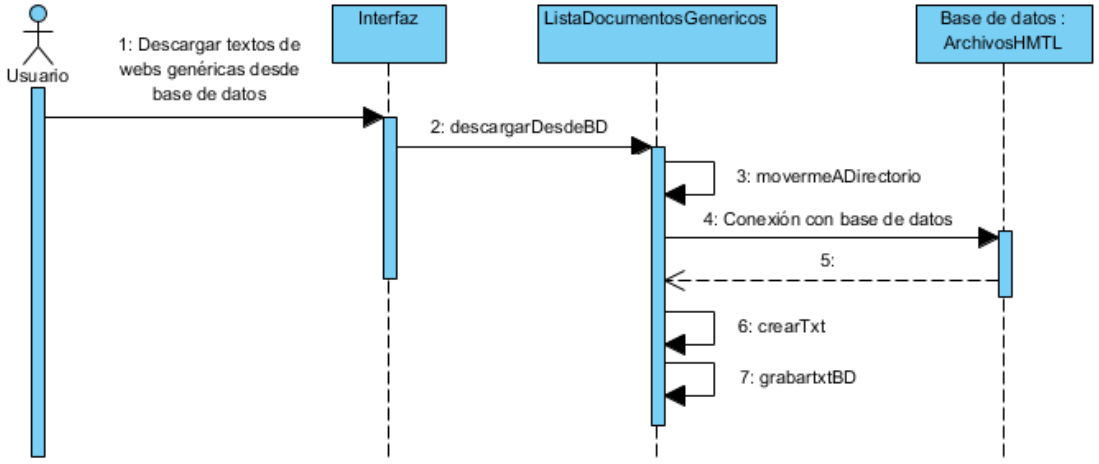


FIGURA A.21: Diagrama de secuencia: Descargar textos genéricos desde la base de datos

Descargar textos HospitalClinic

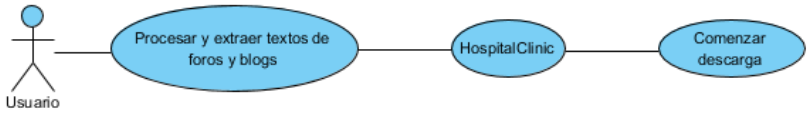


FIGURA A.22: Caso de uso extendido: Descargar textos HospitalClinic

Procesar, descargar y almacenar los textos desde la página web HospitalClinic	
Descripción	Procesa, descarga y almacena el contenido de la página web http://blog.hospitalclinic.org/es/
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto.
Poscondiciones	La tabla ArchivosHospitalClinic de la base de datos se ha actualizado y se ha descargado el contenido de la página web.

TABLA A.8: Descargar textos HospitalClinic

FIGURA A.23: Interfaz gráfica: Descargar textos HospitalClinic

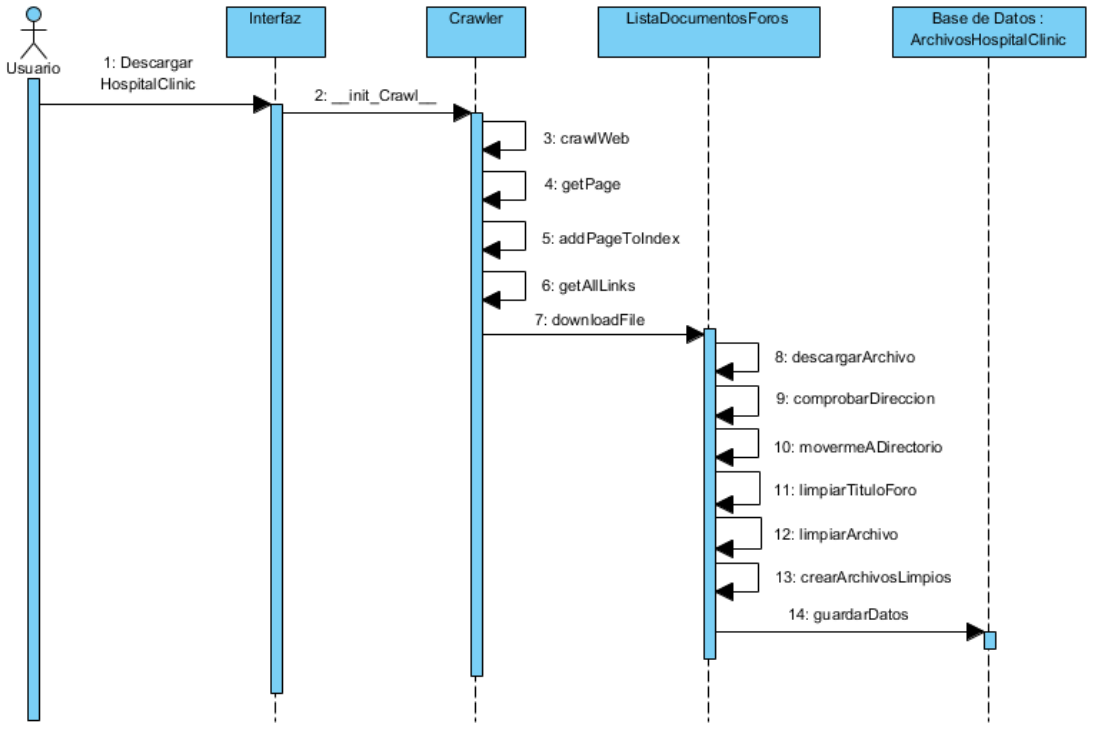


FIGURA A.24: Diagrama de secuencia: Descargar textos HospitalClinic

Descargar textos HospitalClinic desde la base de datos

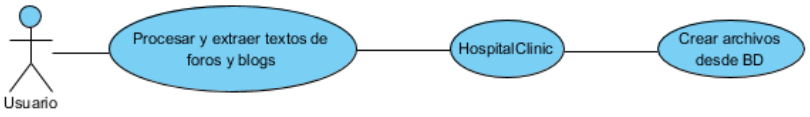


FIGURA A.25: Caso de uso extendido: Descargar textos HospitalClinic desde la base de datos

Convertir en documentos de texto la información de HospitalClinic de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de la página web http://blog.hospitalclinic.org/es/ de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosHospitalClinic contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosHospitalClinic y los registros de la tabla ArchivosHospitalClinic se pasan a documentos de texto.

TABLA A.9: Descargar textos HospitalClinic desde la base de datos

FIGURA A.26: Interfaz gráfica: Descargar textos HospitalClinic desde la base de datos

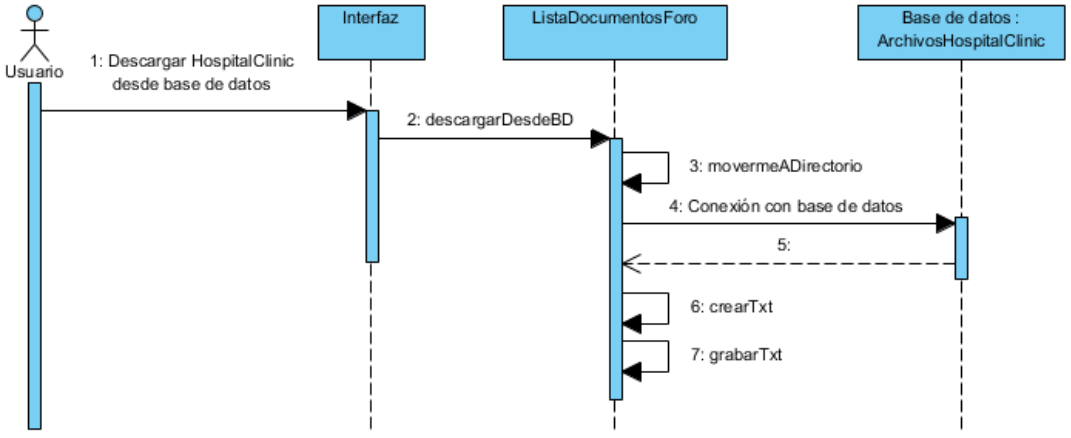


FIGURA A.27: Diagrama de secuencia: Descargar textos HospitalClinic desde la base de datos

Descargar textos DMedicina

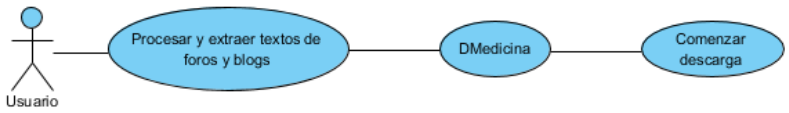


FIGURA A.28: Caso de uso extendido: Descargar textos DMedicina

Procesar, descargar y almacenar los textos desde la página web DMedicina	
Descripción	Procesa, descarga y almacena el contenido de la página web https://dmedicina.com/
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto.
Poscondiciones	La tabla ArchivosDMedicina de la base de datos se ha actualizado y se ha descargado el contenido de la página web.

TABLA A.10: Descargar textos DMedicina

FIGURA A.29: Interfaz gráfica: Descargar textos DMedicina

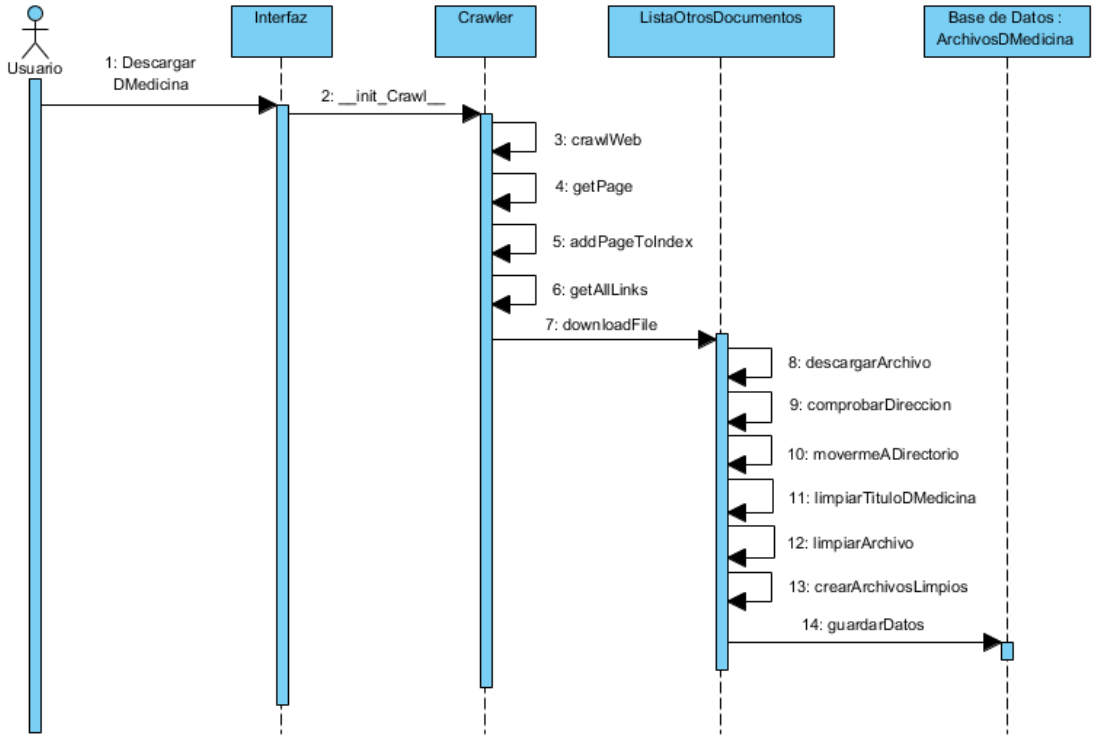


FIGURA A.30: Diagrama de secuencia: Descargar textos DMedicina

Descargar textos DMedicina desde la base de datos

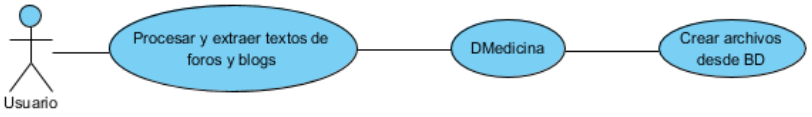


FIGURA A.31: Caso de uso extendido: Descargar textos DMedicina desde la base de datos

Convertir en documentos de texto la información de DMedicina de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de la página web https://dmedicina.com/ de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosDMedicina contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosDMedicina y los registros de la tabla ArchivosDMedicina se pasan a documentos de texto.

TABLA A.11: Descargar textos DMedicina desde la base de datos

FIGURA A.32: Interfaz gráfica: Descargar textos DMedicina desde la base de datos

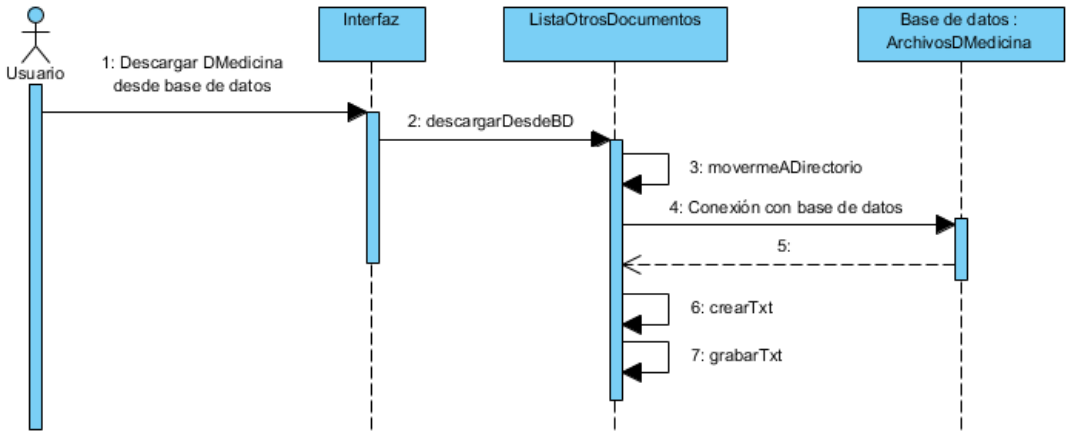


FIGURA A.33: Diagrama de secuencia: Descargar textos DMedicina desde la base de datos

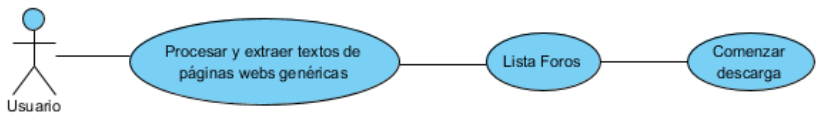


FIGURA A.34: Caso de uso extendido: Descargar textos de una lista de foros y blogs

Procesar, descargar y almacenar los textos de un conjunto de foros y blogs	
Descripción	Procesa, descarga y almacena el contenido de la lista de foros que contiene el archivo ListaForos.txt
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, los foros que contiene el archivo ListaForos.txt, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tablas ArchivosForoHTML y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de los foros.

TABLA A.12: Descargar textos de una lista de foros y blogs

FIGURA A.35: Interfaz gráfica: Descargar textos de una lista de foros y blogs

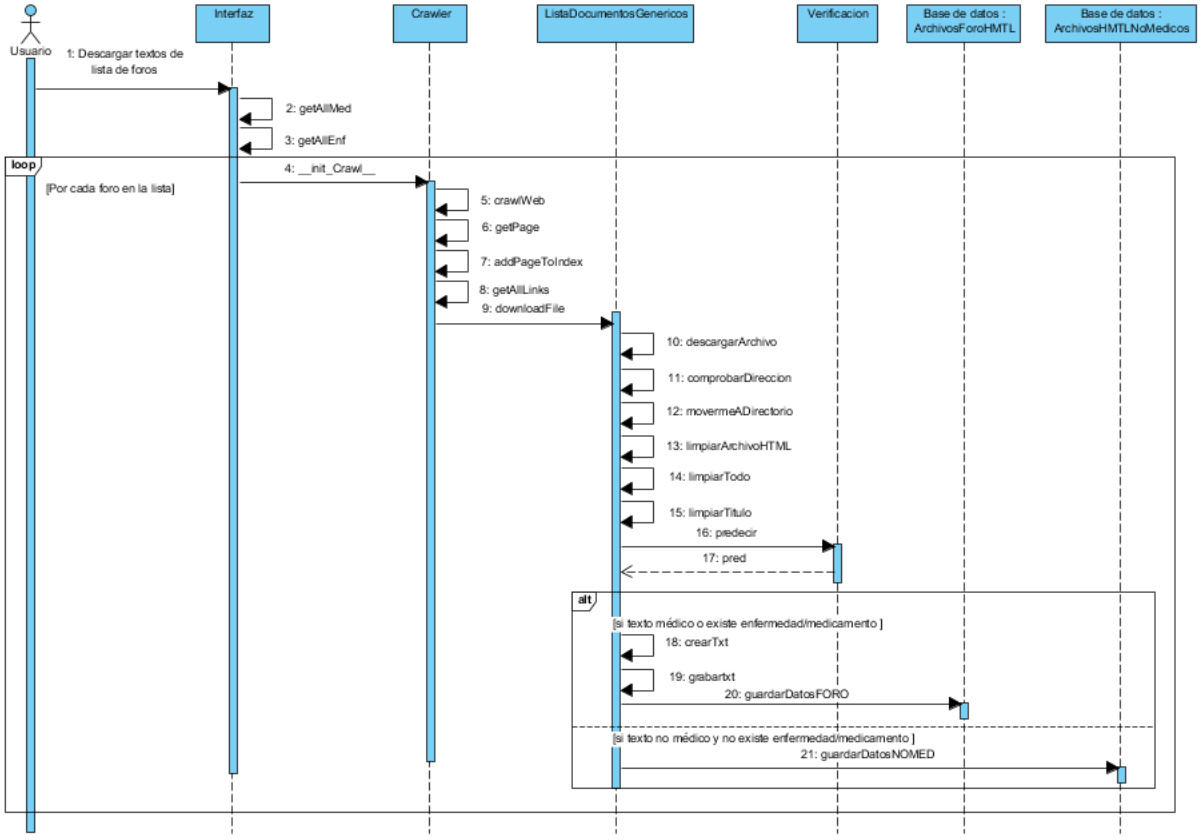


FIGURA A.36: Diagrama de secuencia: Descargar textos de una lista de foros y blogs

Descargar textos genéricos de foros/blogs desde la base de datos

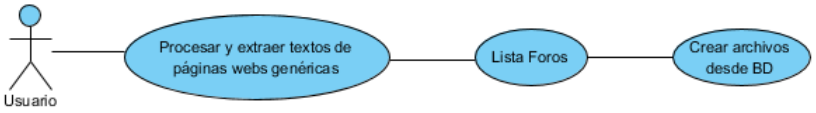


FIGURA A.37: Caso de uso extendido: Descargar textos genéricos de foros/blogs desde la base de datos

Convertir en documentos de texto la información de foros y blogs de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de los foros de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosForoHTML contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosForoHTML y los registros de la tabla ArchivosForoHTML se pasan a documentos de texto.

TABLA A.13: Descargar textos genéricos de foros/blogs desde la base de datos

FIGURA A.38: Interfaz gráfica: Descargar textos genéricos de foros/blogs desde la base de datos

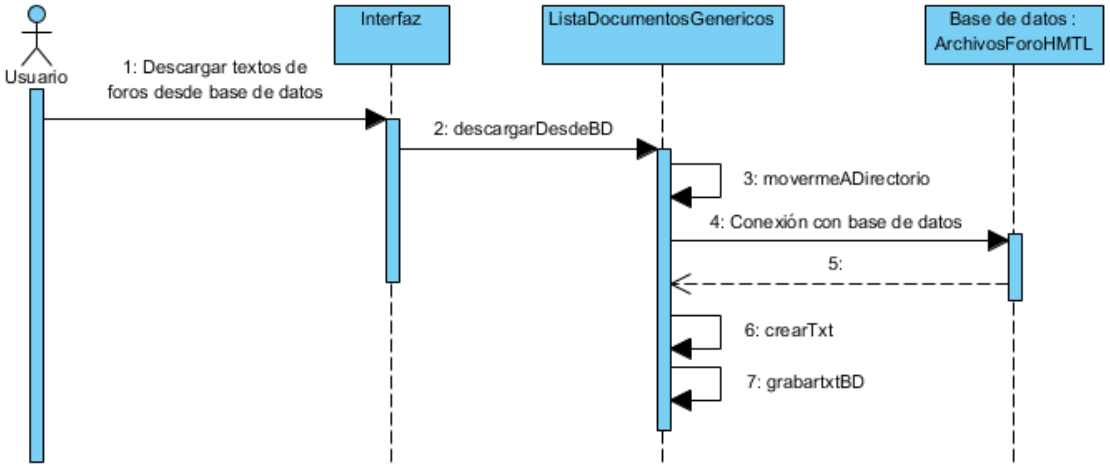


FIGURA A.39: Diagrama de secuencia: Descargar textos genéricos de foros/blogs desde la base de datos

Descargar textos de webs vía GoogleSearch

FIGURA A.40: Caso de uso extendido: Descargar textos de webs vía GoogleSearch

Procesar, descargar y almacenar los textos de las webs que devuelva la API GoogleSearch	
Descripción	Procesa, descarga y almacena el contenido de la lista de páginas webs que devuelve la búsqueda de un término en GoogleSearch.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tablas ArchivosAPI y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de las páginas webs.

TABLA A.14: Descargar textos de webs vía GoogleSearch

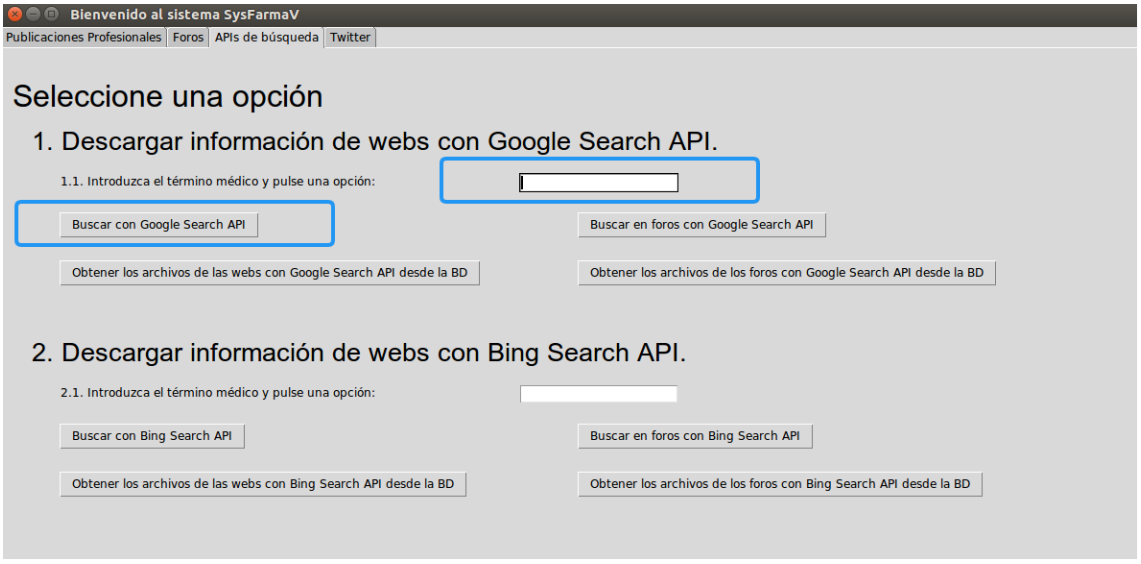


FIGURA A.41: Interfaz gráfica: Descargar textos de webs vía GoogleSearch

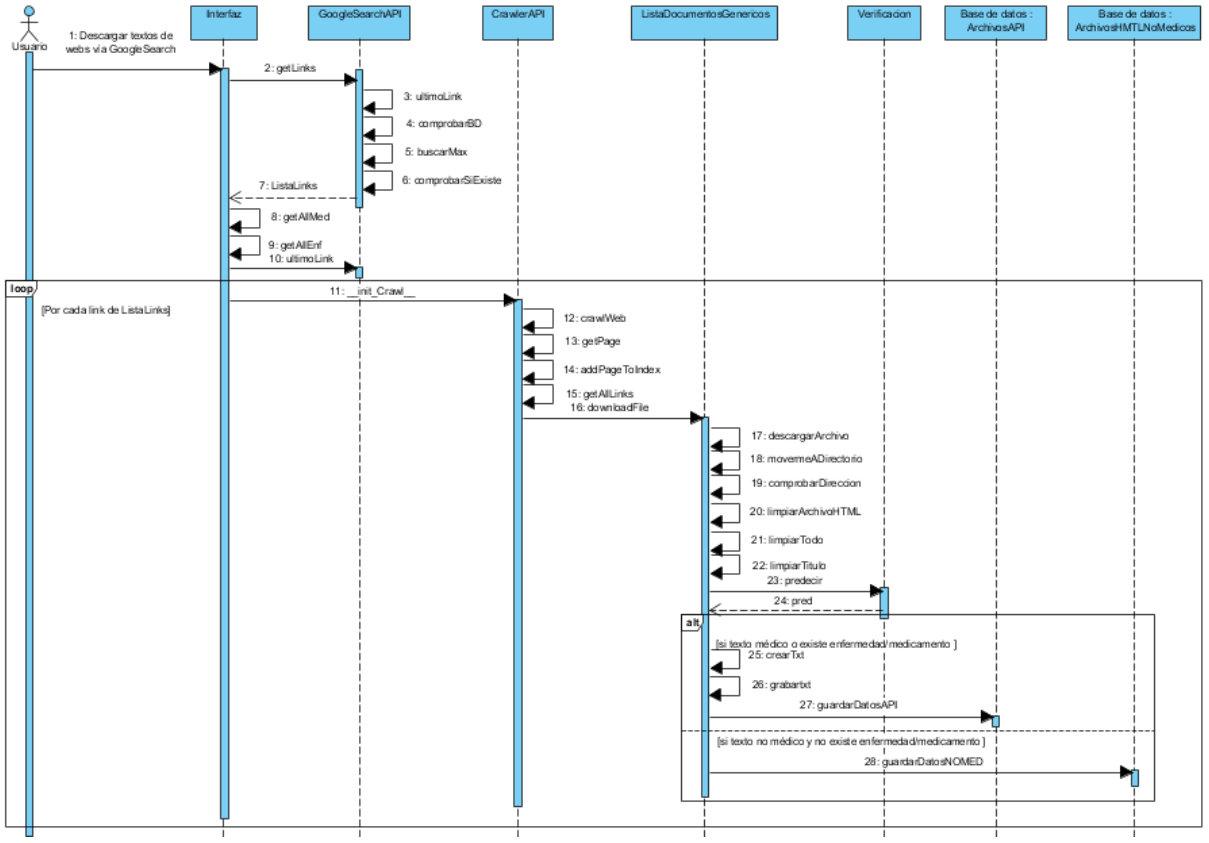


FIGURA A.42: Diagrama de secuencia: Descargar textos de webs vía GoogleSearch

Descargar textos de webs vía GoogleSearch desde la base de datos

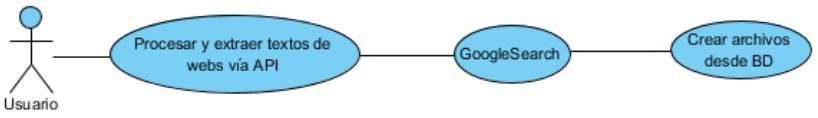


FIGURA A.43: Caso de uso extendido: Descargar textos de webs vía GoogleSearch desde la base de datos

Convertir en documentos de texto la información de webs vía GoogleSearch de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de las páginas webs que ha devuelto GoogleSearch de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosAPI contiene al menos un registro de una búsqueda con GoogleSearch.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosAPI y los registros de una búsqueda con GoogleSearch de la tabla ArchivosAPI se pasan a documentos de texto.

Tabla A.15: Descargar textos de webs vía GoogleSearch desde la base de datos

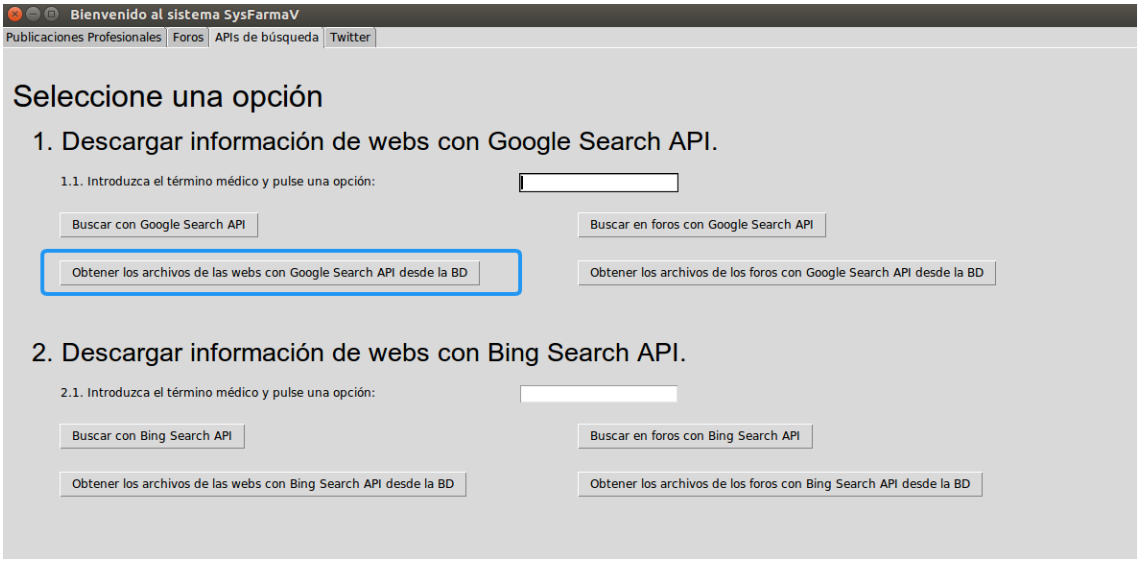


Figura A.44: Interfaz gráfica: Descargar textos de webs vía GoogleSearch desde la base de datos

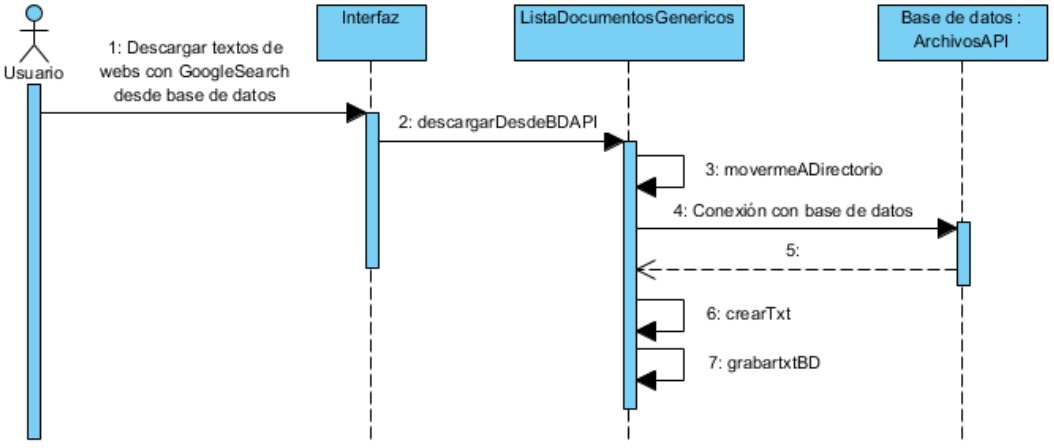


Figura A.45: Diagrama de secuencia: Descargar textos de webs vía GoogleSearch desde la base de datos

Descargar textos de foros vía GoogleSearch

FIGURA A.46: Caso de uso extendido: Descargar textos de foros vía GoogleSearch

Procesar, descargar y almacenar los textos de los foros que devuelva la API GoogleSearch	
Descripción	Procesa, descarga y almacena el contenido de la lista de foros que devuelve la búsqueda de un término en GoogleSearch.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tablas ArchivosAPI ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de los foros.

TABLA A.16: Descargar textos de foros vía GoogleSearch

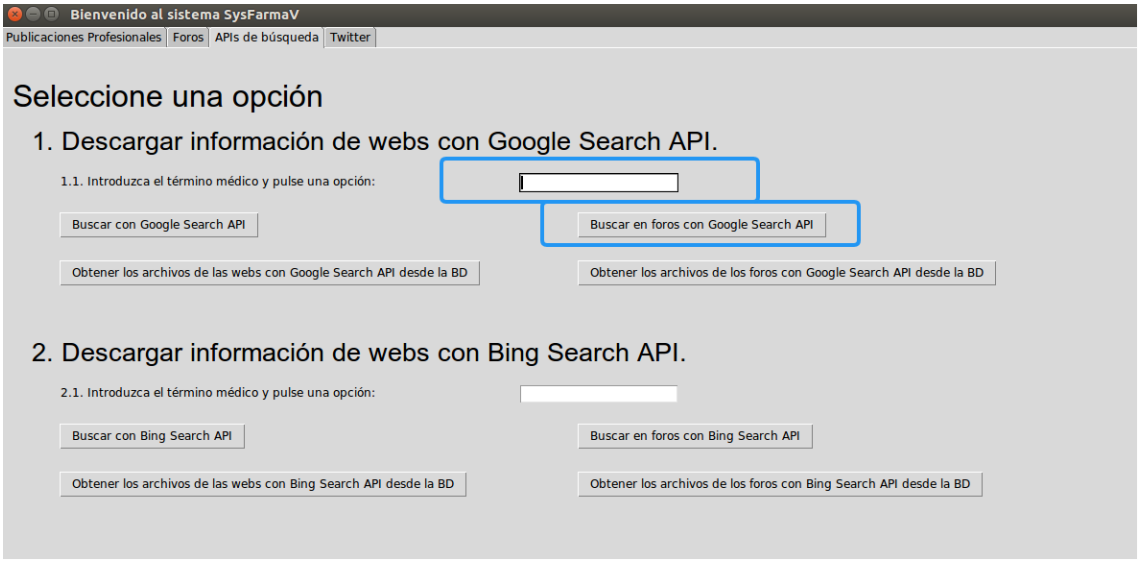


FIGURA A.47: Interfaz gráfica: Descargar textos de foros vía GoogleSearch

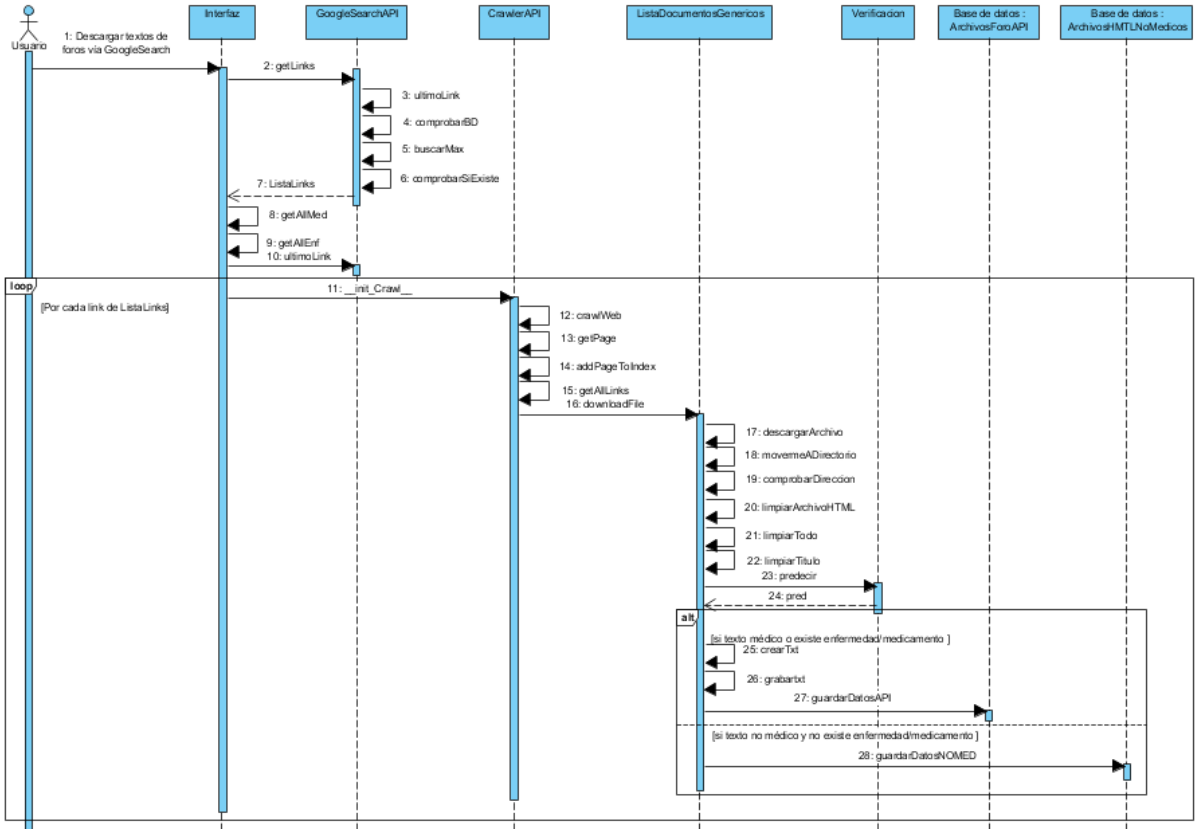


FIGURA A.48: Diagrama de secuencia: Descargar textos de foros vía GoogleSearch

Descargar textos de foros vía GoogleSearch desde la base de datos

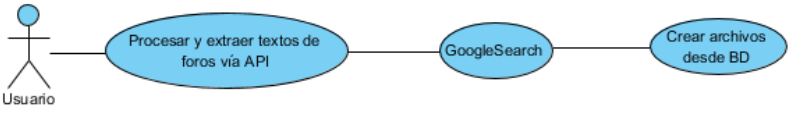


FIGURA A.49: Caso de uso extendido: Descargar textos de foros vía GoogleSearch desde la base de datos

Convertir en documentos de texto la información de foros vía GoogleSearch de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de los foros que ha devuelto GoogleSearch de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosForoAPI contiene al menos un registro de una búsqueda con GoogleSearch.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosForoAPI y los registros de una búsqueda con GoogleSearch de la tabla ArchivosForoAPI se pasan a documentos de texto.

TABLA A.17: Descargar textos de foros vía GoogleSearch desde la base de datos

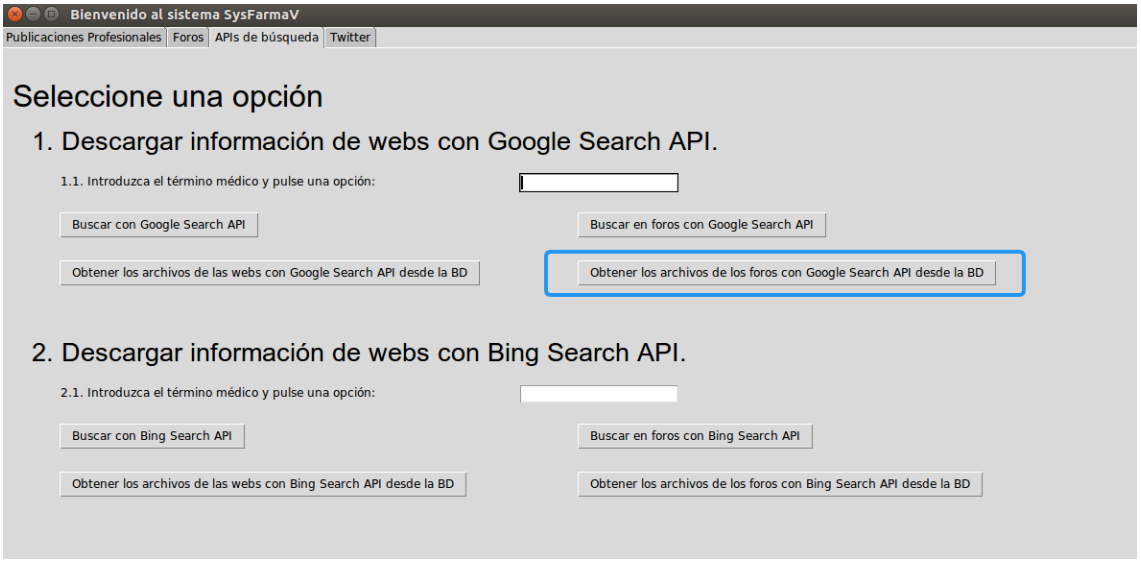


FIGURA A.50: Interfaz gráfica: Descargar textos de foros vía GoogleSearch desde la base de datos

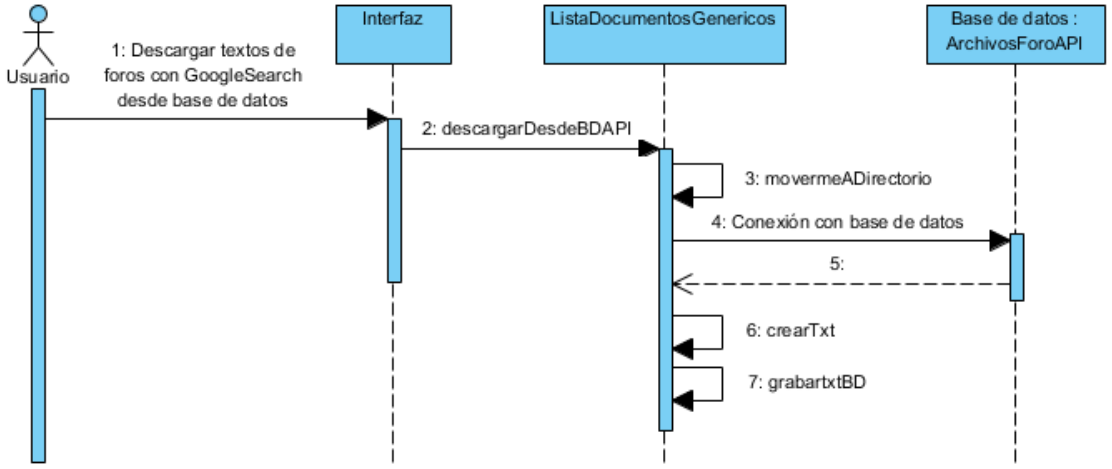


FIGURA A.51: Diagrama de secuencia: Descargar textos de foros vía GoogleSearch desde la base de datos

Descargar textos de webs vía API de Bing

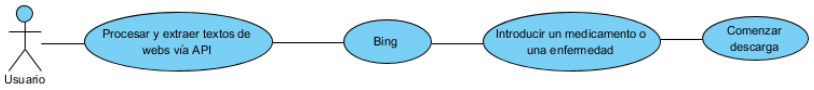


FIGURA A.52: Caso de uso extendido: Descargar textos de webs vía API de Bing

Procesar, descargar y almacenar los textos de las webs que devuelva la API de Bing	
Descripción	Procesa, descarga y almacena el contenido de la lista de páginas webs que devuelve la búsqueda de un término en la API de Bing.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tablas ArchivosAPI y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de las páginas webs.

TABLA A.18: Descargar textos de webs vía API de Bing

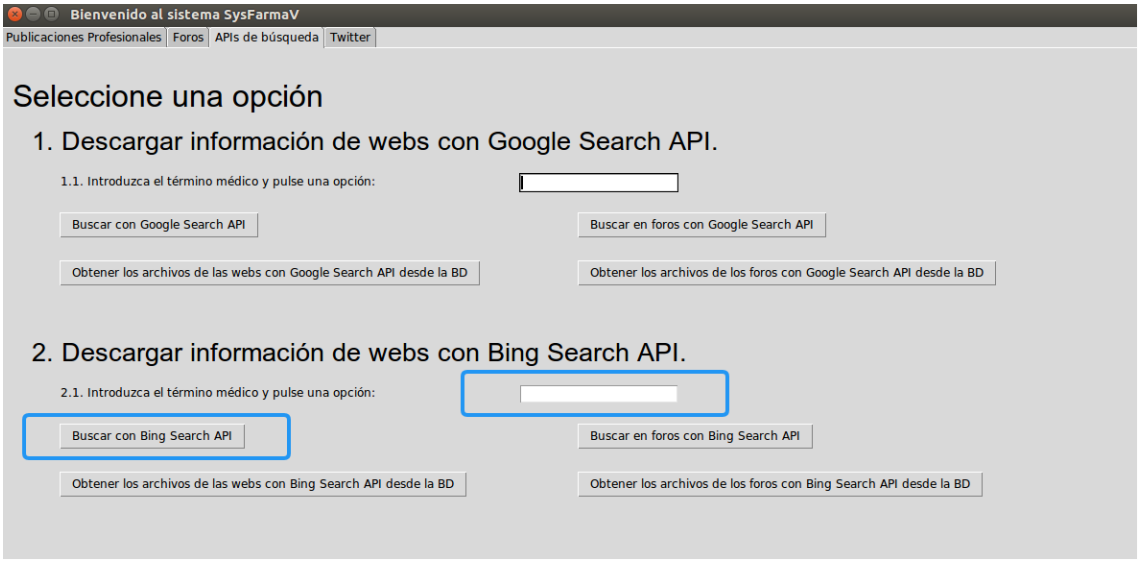


FIGURA A.53: Interfaz gráfica: Descargar textos de webs vía API de Bing

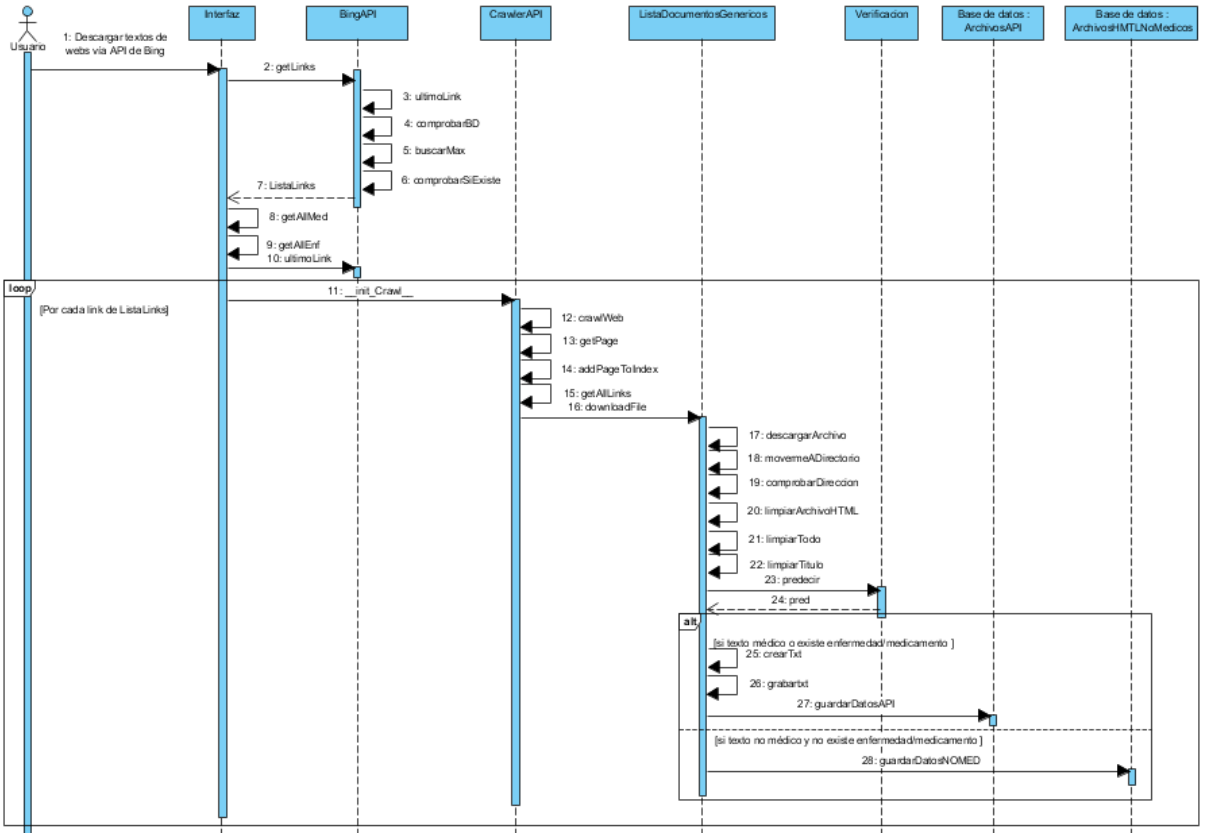


FIGURA A.54: Diagrama de secuencia: Descargar textos de webs vía API de Bing

Descargar textos de webs vía API de Bing desde la base de datos

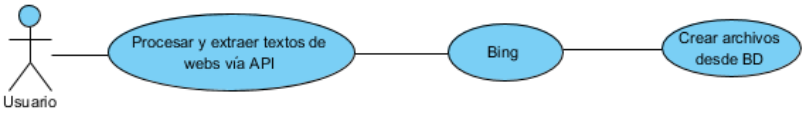


FIGURA A.55: Caso de uso extendido: Descargar textos de webs vía API de Bing desde la base de datos

Convertir en documentos de texto la información de webs vía GoogleSearch de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de las páginas webs que ha devuelto la API de Bing de la base de datos.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto y la tabla ArchivosAPI contiene al menos un registro de una búsqueda con Bing.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosAPI y los registros de una búsqueda con Bing de la tabla ArchivosAPI se pasan a documentos de texto.

TABLA A.19: Descargar textos de webs vía API de Bing desde la base de datos

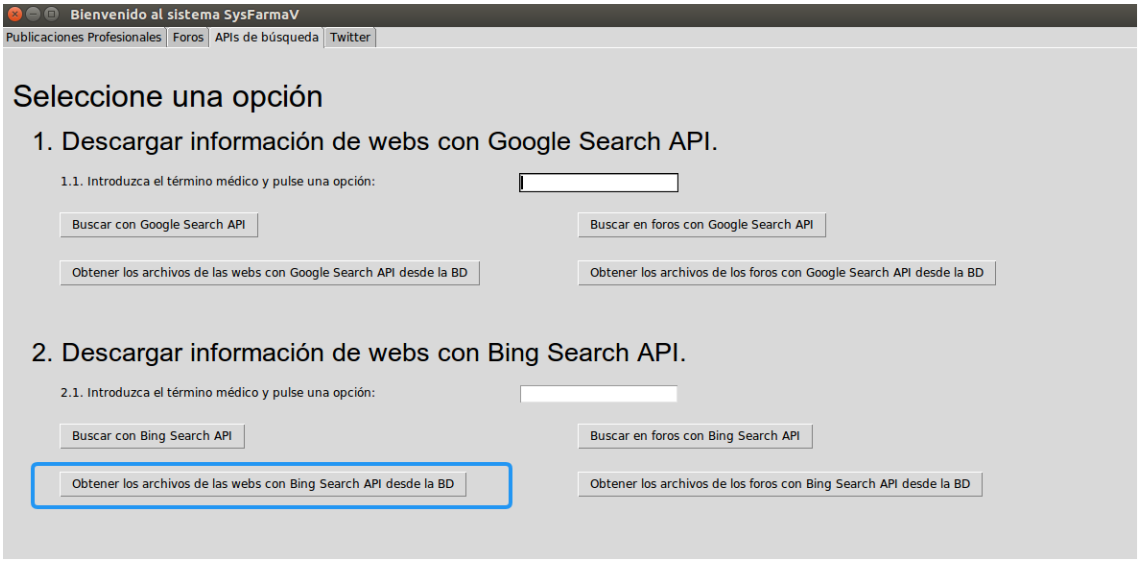


FIGURA A.56: Interfaz gráfica: Descargar textos de webs vía API de Bing desde la base de datos

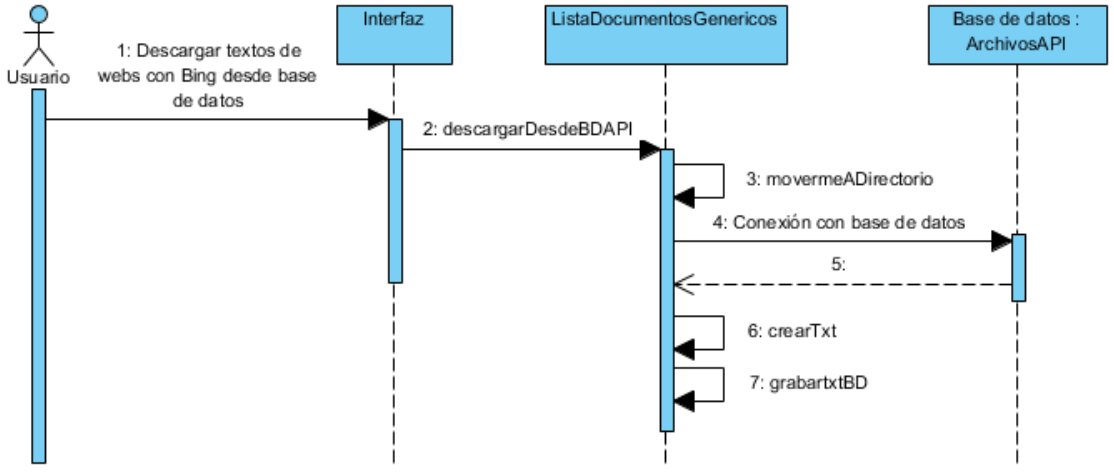


FIGURA A.57: Diagrama de secuencia: Descargar textos de webs vía API de Bing desde la base de datos

Descargar textos de foros vía API de Bing

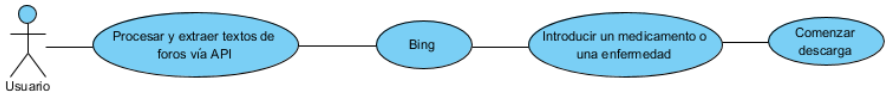


FIGURA A.58: Caso de uso extendido: Descargar textos de foros vía API de Bing

Procesar, descargar y almacenar los textos de los foros que devuelva la API de Bing	
Descripción	Procesa, descarga y almacena el contenido de la lista de foros que devuelve la búsqueda de un término en la API de Bing.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, los archivos Enfermedades.txt y Medicamentos.txt y un modelo clasificador.
Poscondiciones	Las tablas ArchivosAPI y ArchivosHTMLNoMedicos de la base de datos se han actualizado y se ha descargado el contenido de los foros.

TABLA A.20: Descargar textos de foros vía API de Bing

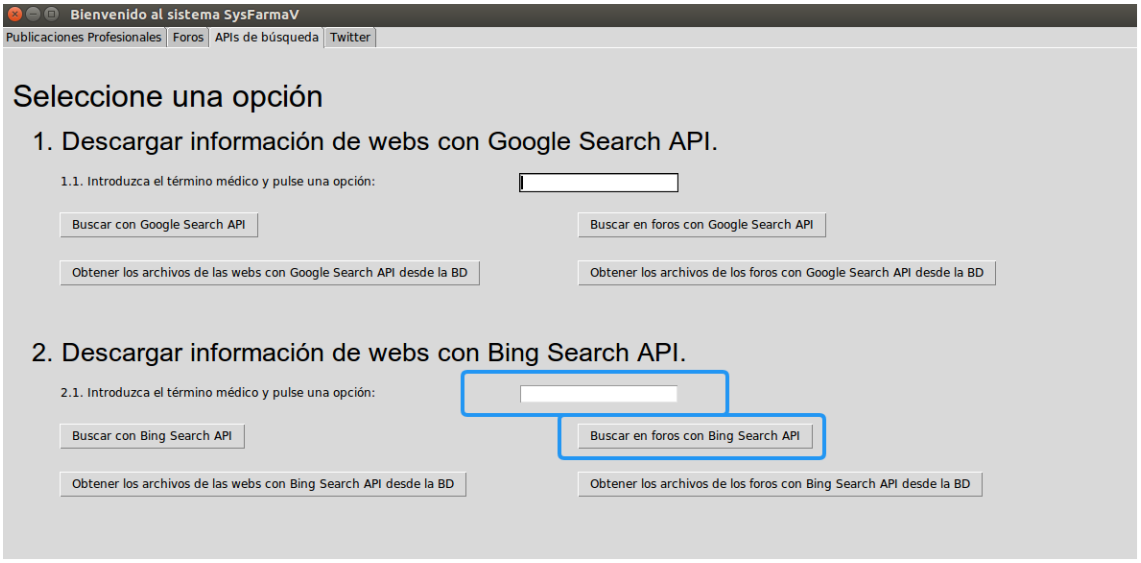


FIGURA A.59: Interfaz gráfica: Descargar textos de foros vía API de Bing

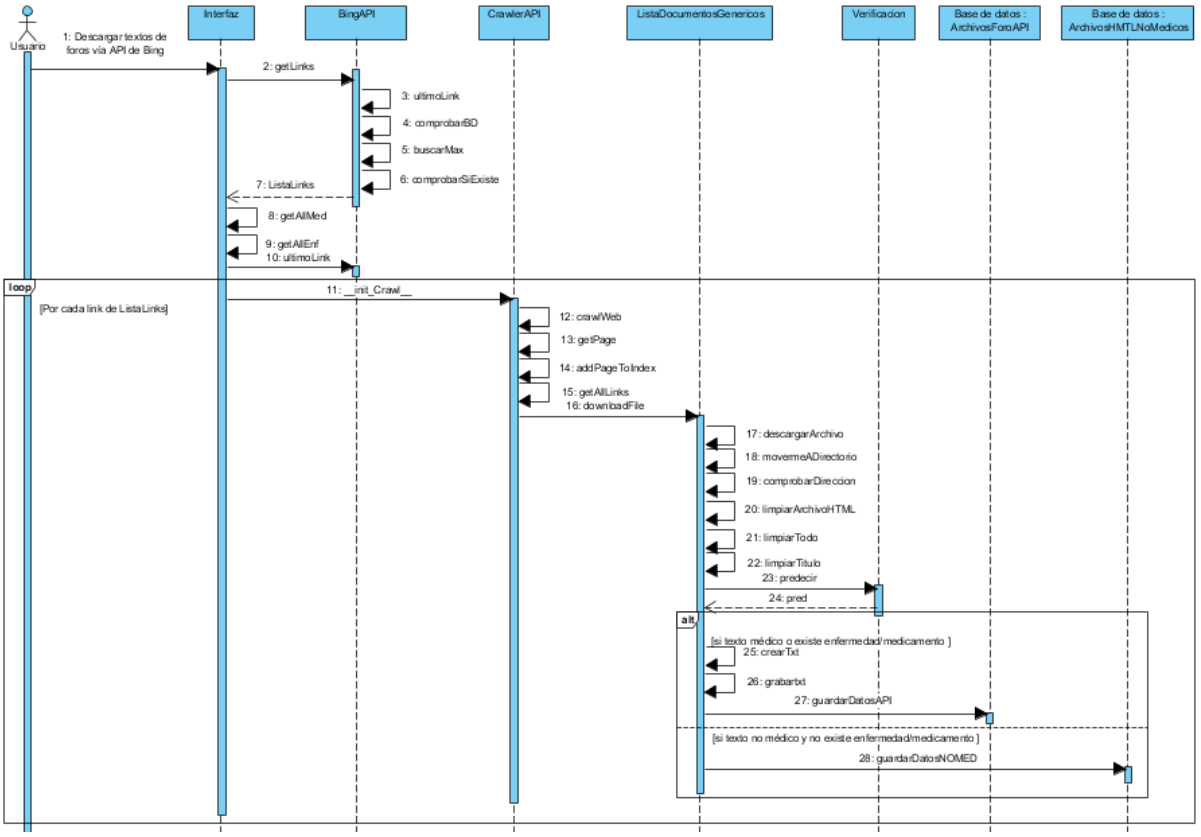


FIGURA A.60: Diagrama de secuencia: Descargar textos de foros vía API de Bing

Descargar textos de foros vía API de Bing desde la base de datos

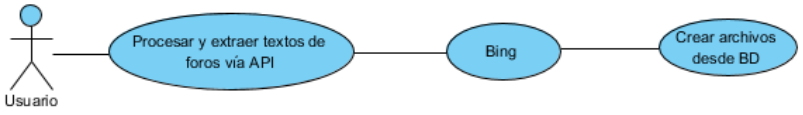


FIGURA A.61: Caso de uso extendido: Descargar textos de foros vía API de Bing desde la base de datos

Convertir en documentos de texto la información de foros vía API de Bing de la base de datos	
Descripción	Convierte en documentos de texto los datos descargados de los foros que ha devuelto la API de Bing de la base de datos.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla ArchivosForoAPI contiene al menos un registro de una búsqueda con la API de Bing.
Poscondiciones	Se crea el directorio ArchivosDB-ArchivosForoAPI y los registros de una búsqueda con la API de Bing de la tabla ArchivosForoAPI se pasan a documentos de texto.

TABLA A.21: Descargar textos de foros vía API de Bing desde la base de datos

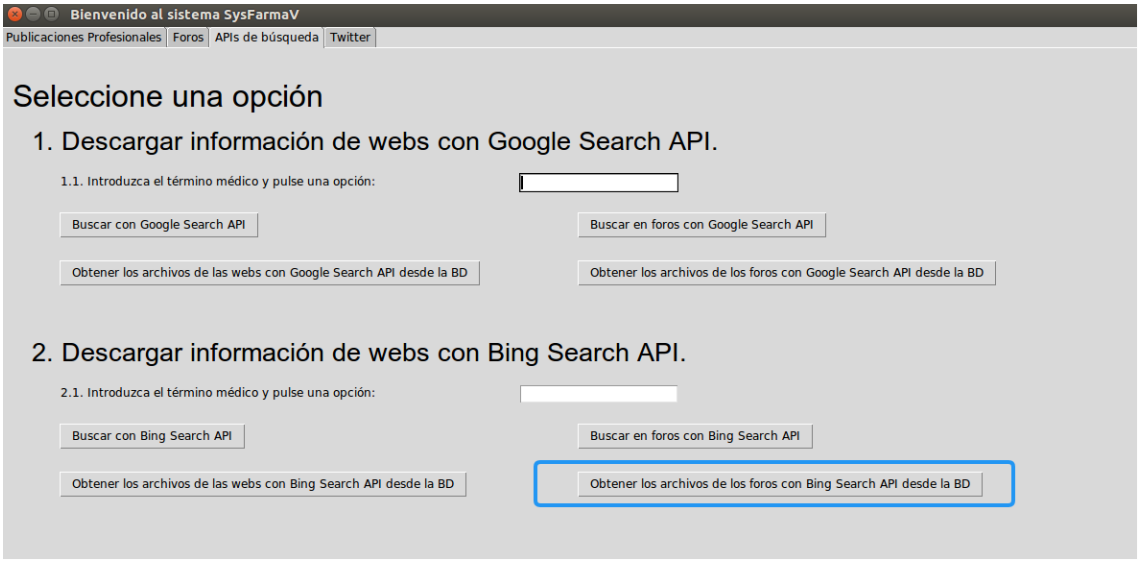


FIGURA A.62: Interfaz gráfica: Descargar textos de foros vía API de Bing desde la base de datos

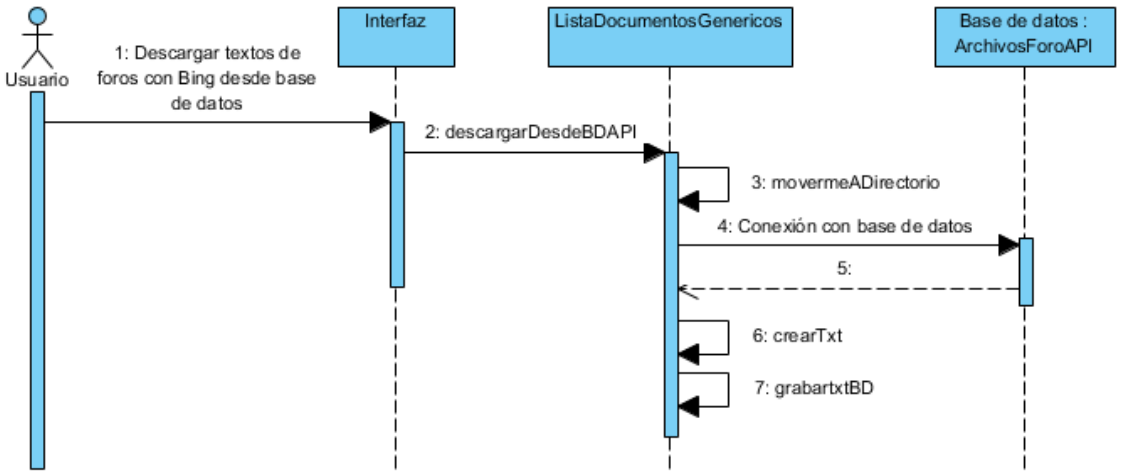


FIGURA A.63: Diagrama de secuencia: Descargar textos de foros vía API de Bing desde la base de datos

Descargar textos de Twitter a partir del nombre de un medicamento

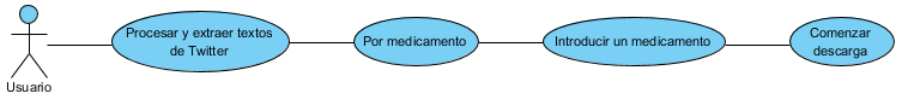


FIGURA A.64: Caso de uso extendido: Descargar textos de Twitter a partir del nombre de un medicamento

Procesar, descargar y almacenar los tuits publicados que contengan el nombre del medicamento	
Descripción	Procesa, descarga y almacena los tuits publicados que contengan el medicamento que el usuario haya introducido.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto y una API de Twitter con una clave de autentificación para desarrolladores.
Poscondiciones	Las tablas TwitterMed y TwitterConv de la base de datos se han actualizado y se han descargado los tuits.

TABLA A.22: Descargar textos de Twitter a partir del nombre de un medicamento

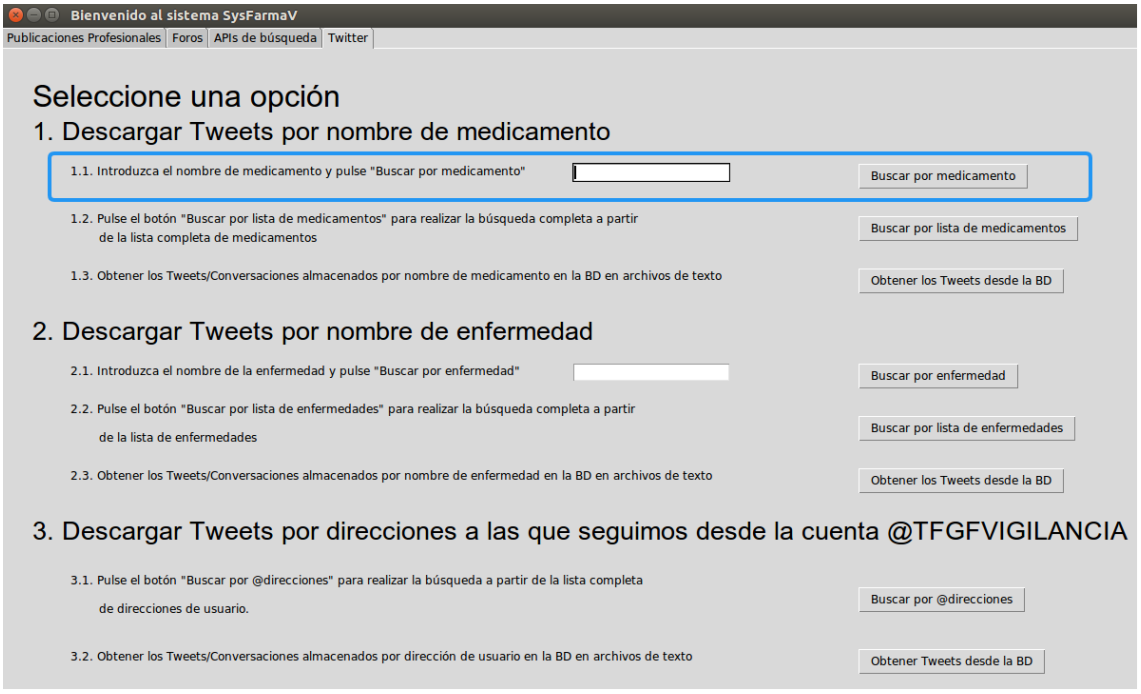


FIGURA A.65: Interfaz gráfica: Descargar textos de Twitter a partir del nombre de un medicamento

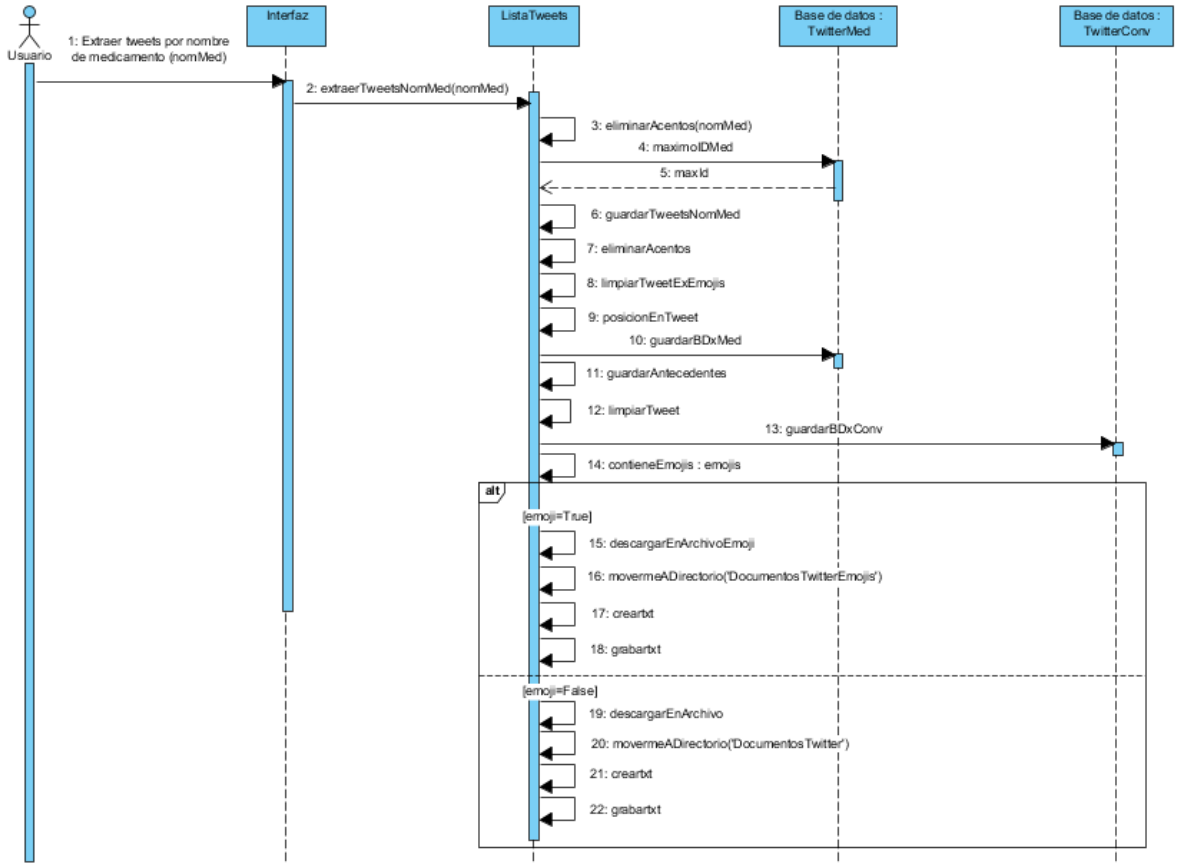


FIGURA A.66: Diagrama de secuencia: Descargar textos de Twitter a partir del nombre de un medicamento

Descargar textos de Twitter a partir de una lista de medicamentos

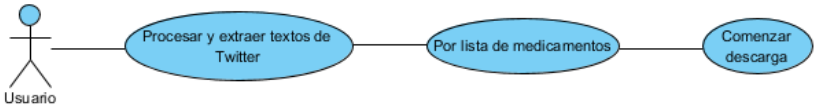


FIGURA A.67: Caso de uso extendido: Descargar textos de Twitter a partir de una lista de medicamentos

Procesar, descargar y almacenar tuits que contengan algún medicamento de la lista Medicamentos.txt	
Descripción	Procesa, descarga y almacena los tuits publicados que contengan los medicamento que aparecen en el archivo Medicamentos.txt
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, el archivo Medicamentos.txt y una API de Twitter con una clave de autentificación para desarrolladores.
Poscondiciones	Las tablas TwitterMed y TwitterConv de la base de datos se han actualizado y se han descargado los tuits.

TABLA A.23: Descargar textos de Twitter a partir de una lista de medicamentos

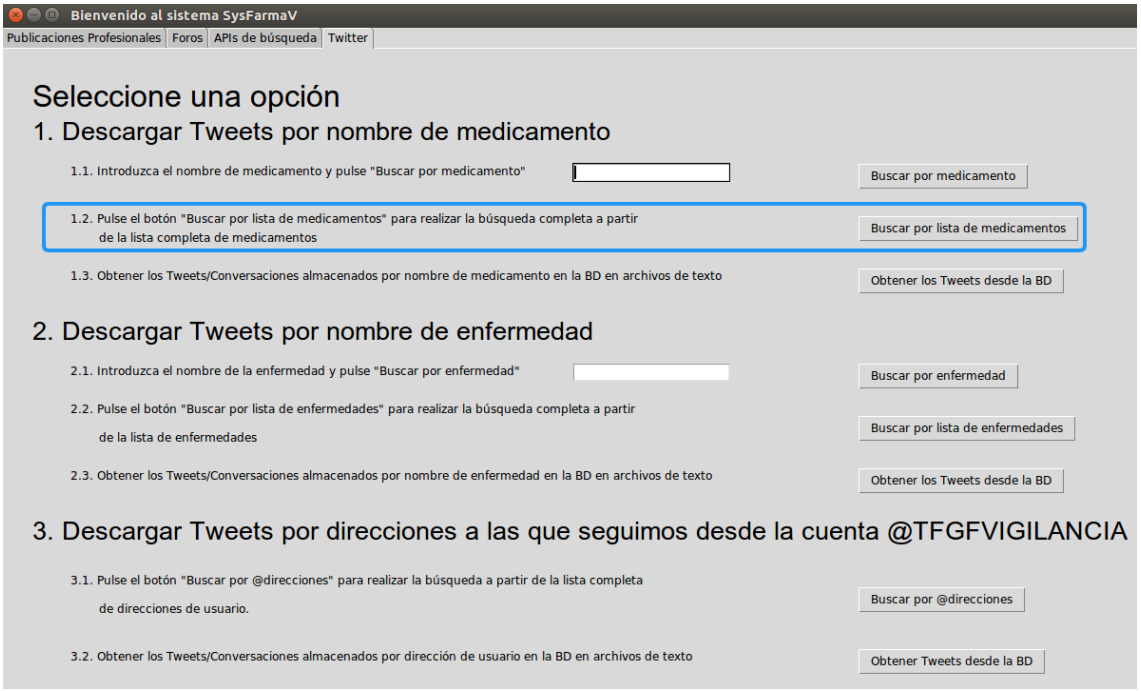


FIGURA A.68: Interfaz gráfica: Descargar textos de Twitter a partir de una lista de medicamentos

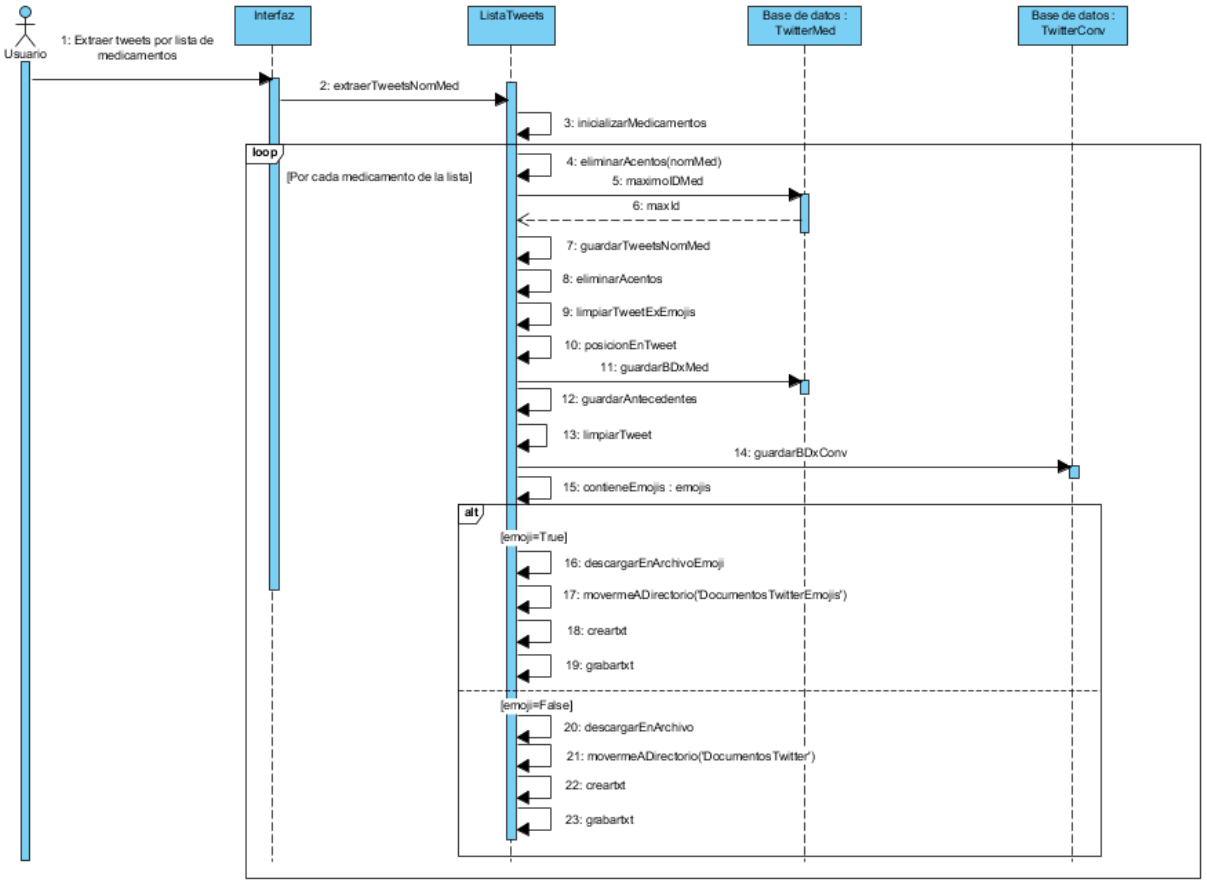


FIGURA A.69: Diagrama de secuencia: Descargar textos de Twitter a partir de una lista de medicamentos

Descargar textos de Twitter por nombre de medicamento desde la base de datos

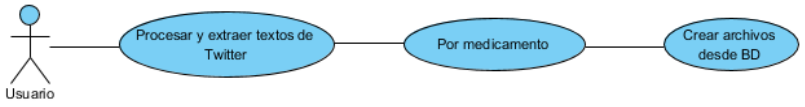


FIGURA A.70: Caso de uso extendido: Descargar textos de Twitter por nombre de medicamento desde la base de datos

Convertir en documentos de texto los tuits de medicamentos de la base de datos	
Descripción	Convierte en documentos de texto los tuits de medicamentos descargados en las tablas TwitterMed y TwitterConv.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla TwitterMed contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-TwitterMed y los registros de la tabla TwitterMed y TwitterConv se pasan a documentos de texto.

TABLA A.24: Descargar textos de Twitter por nombre de medicamento desde la base de datos

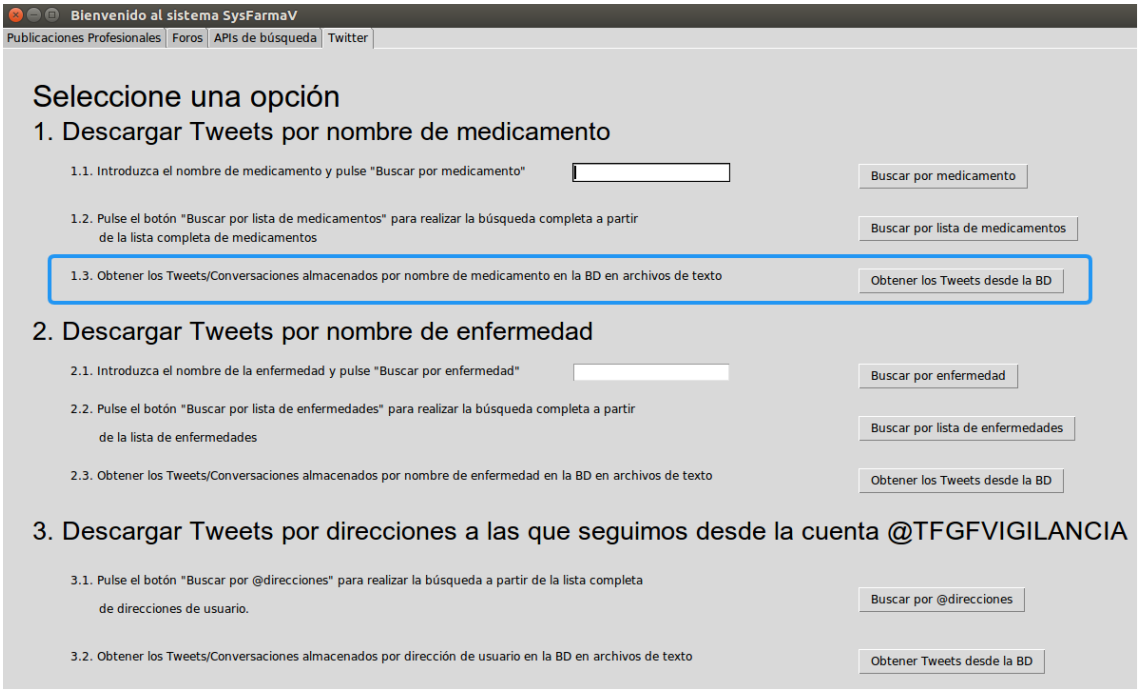


FIGURA A.71: Interfaz gráfica: Descargar textos de Twitter por nombre de medicamento desde la base de datos

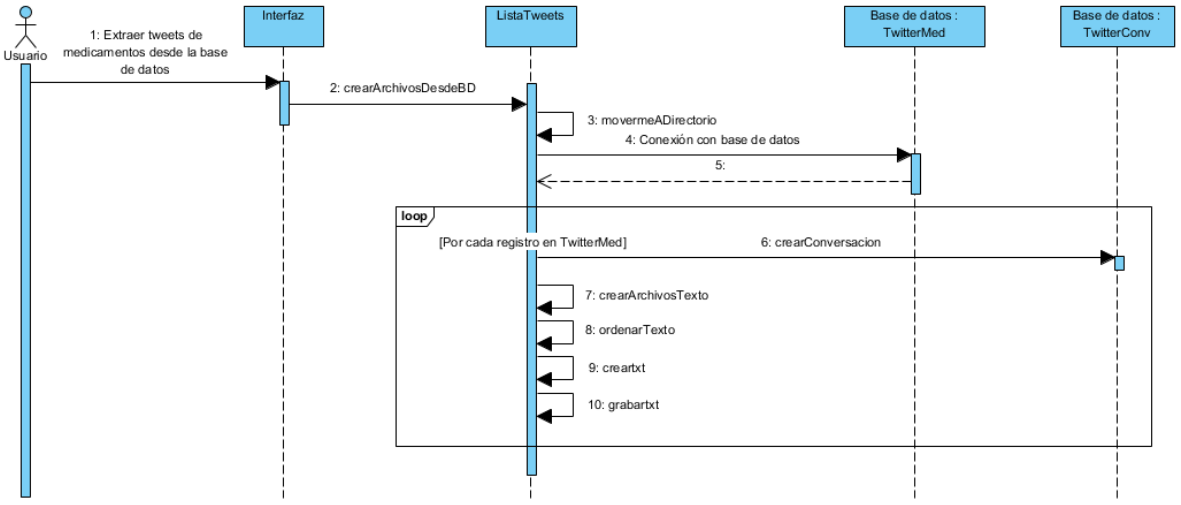


FIGURA A.72: Diagrama de secuencia: Descargar textos de Twitter por nombre de medicamento desde la base de datos

Descargar textos de Twitter a partir del nombre de una enfermedad

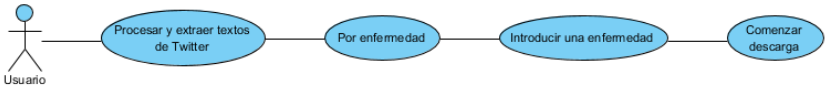


FIGURA A.73: Caso de uso extendido: Descargar textos de Twitter a partir del nombre de una enfermedad

Procesar, descargar y almacenar los tuits publicados que contengan el nombre de la enfermedad	
Descripción	Procesa, descarga y almacena los tuits publicados que contengan la enfermedad que el usuario haya introducido.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto y una API de Twitter con una clave de autentificación para desarrolladores.
Poscondiciones	Las tablas TwitterEnf y TwitterConv de la base de datos se han actualizado y se han descargado los tuits.

TABLA A.25: Descargar textos de Twitter a partir del nombre de una enfermedad

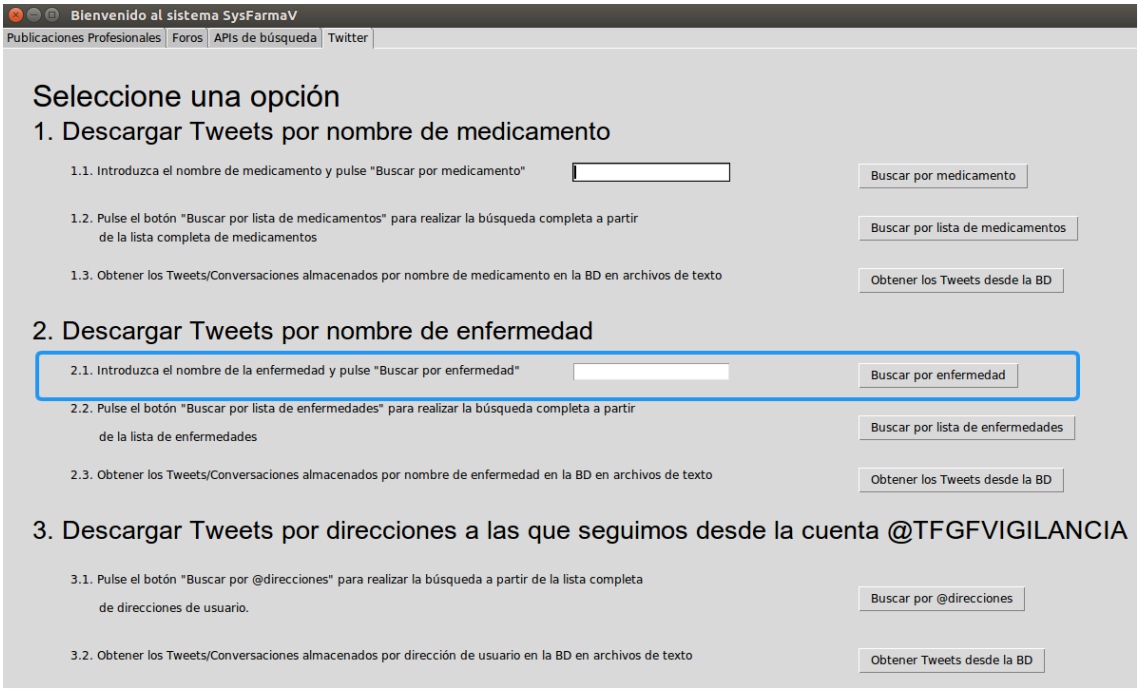


FIGURA A.74: Interfaz gráfica: Descargar textos de Twitter a partir del nombre de una enfermedad

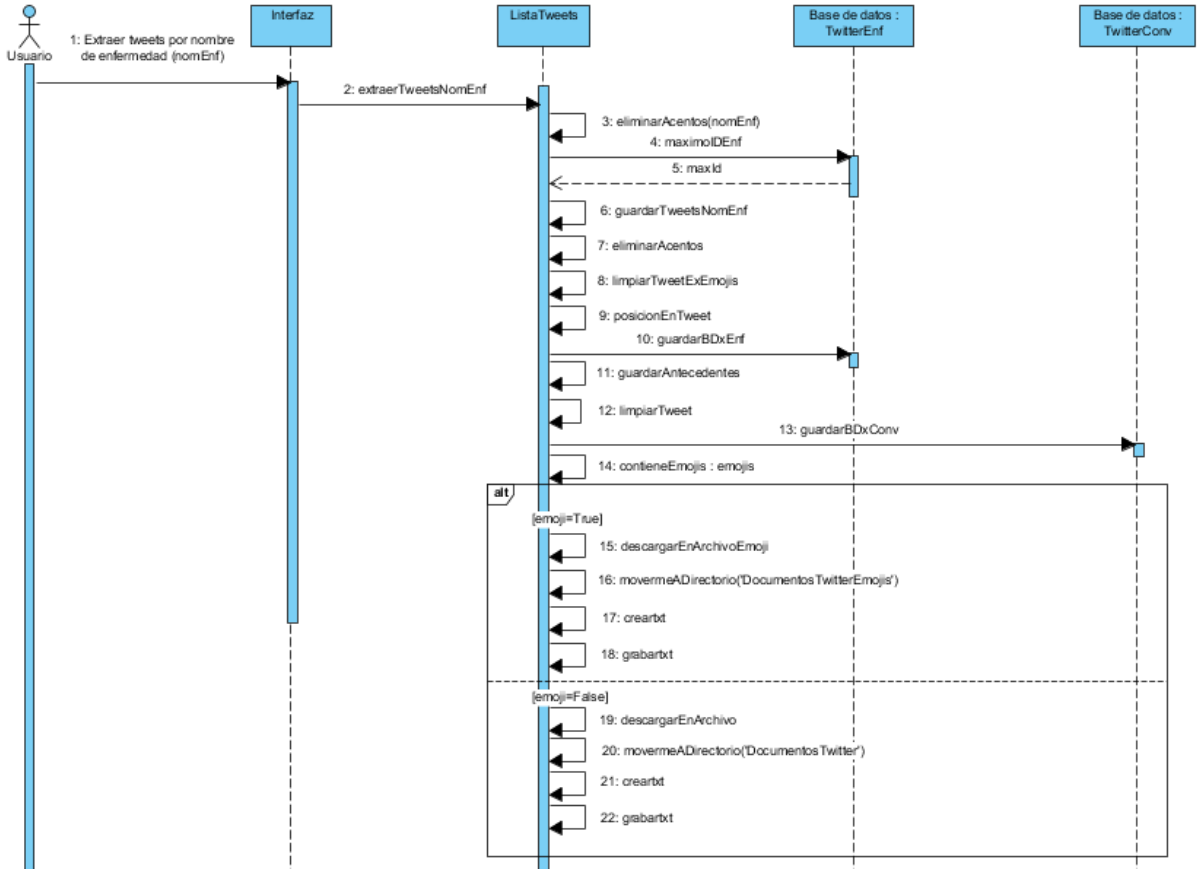


FIGURA A.75: Diagrama de secuencia: Descargar textos de Twitter a partir del nombre de una enfermedad

Descargar textos de Twitter a partir de una lista de enfermedades

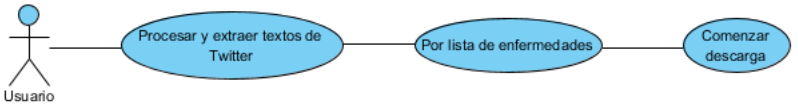


FIGURA A.76: Caso de uso extendido: Descargar textos de Twitter a partir de una lista de enfermedades

Procesar, descargar y almacenar tuits que contengan alguna enfermedad de la lista Enfermedades.txt	
Descripción	Procesa, descarga y almacena los tuits publicados que contengan las enfermedades que aparecen en el archivo Enfermedades.txt
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, el archivo Enfermedades.txt y una API de Twitter con una clave de autentificación para desarrolladores.
Poscondiciones	Las tablas TwitterEnf y TwitterConv de la base de datos se han actualizado y se han descargado los tuits.

Tabla A.26: Descargar textos de Twitter a partir de una lista de enfermedades

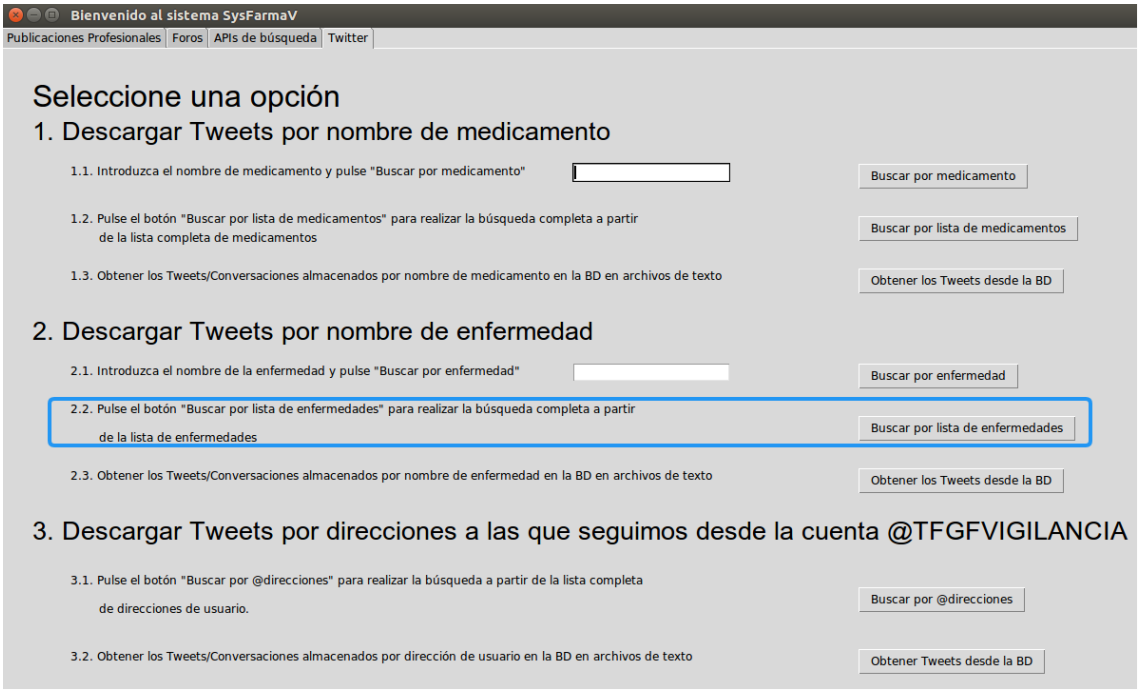


Figura A.77: Interfaz gráfica: Descargar textos de Twitter a partir de una lista de enfermedades

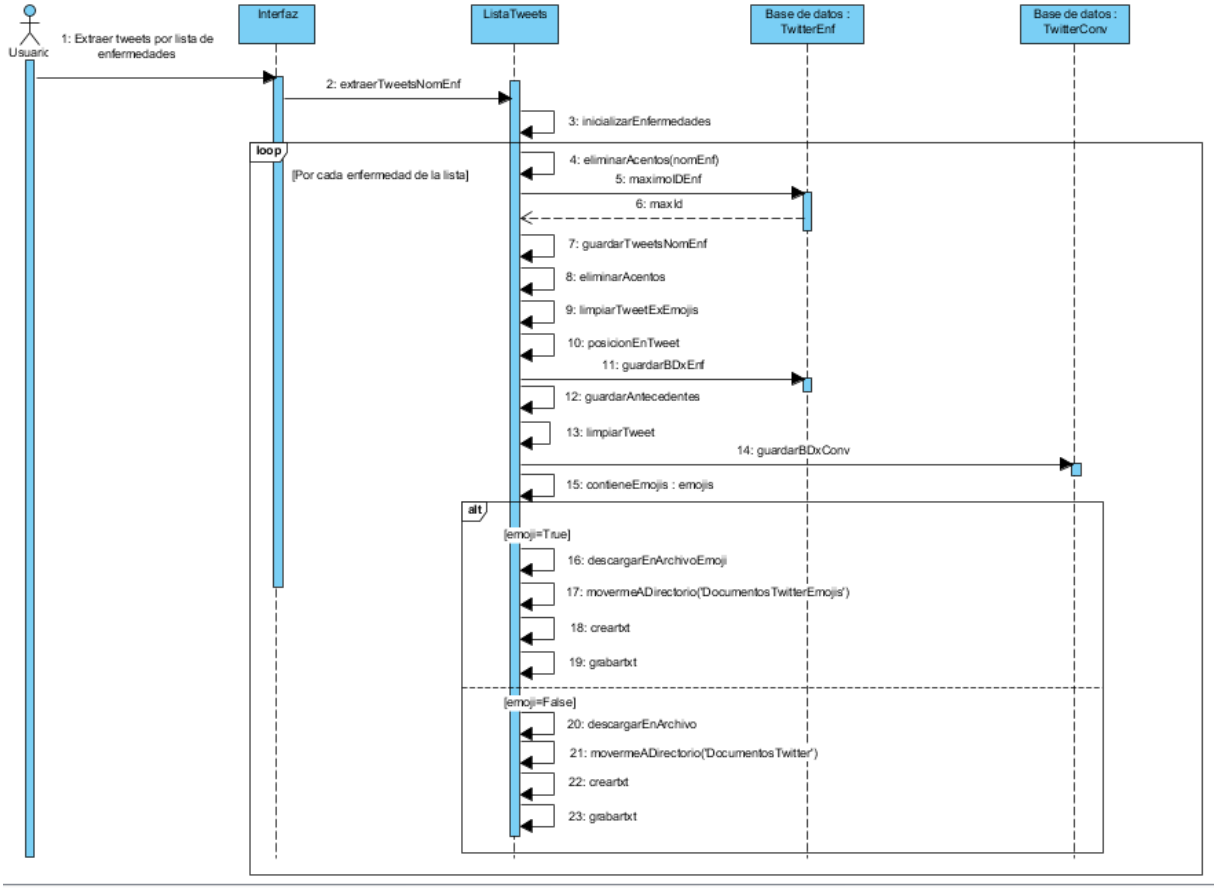


FIGURA A.78: Diagrama de secuencia: Descargar textos de Twitter a partir de una lista de enfermedades

Descargar textos de Twitter por nombre de enfermedad desde la base de datos

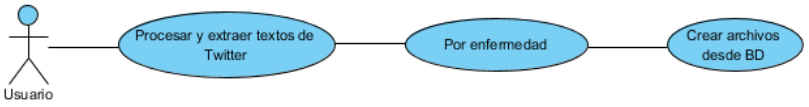


FIGURA A.79: Caso de uso extendido: Descargar textos de Twitter por nombre de enfermedad desde la base de datos

Convertir en documentos de texto los tuits de enfermedades de la base de datos	
Descripción	Convierte en documentos de texto los tuits de enfermedades descargados en las tablas TwitterEnf y TwitterConv.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla TwitterEnf contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-TwitterEnf y los registros de la tabla TwitterEnf y TwitterConv se pasan a documentos de texto.

TABLA A.27: Descargar textos de Twitter por nombre de enfermedad desde la base de datos

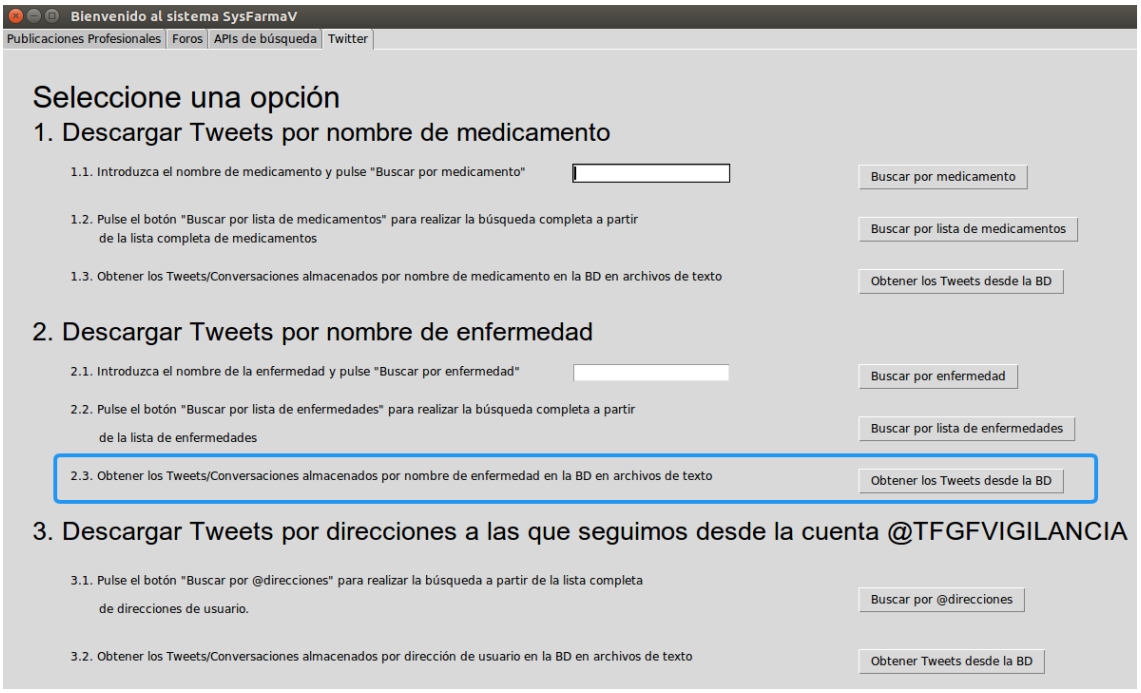


FIGURA A.80: Interfaz gráfica: Descargar textos de Twitter por nombre de enfermedad desde la base de datos

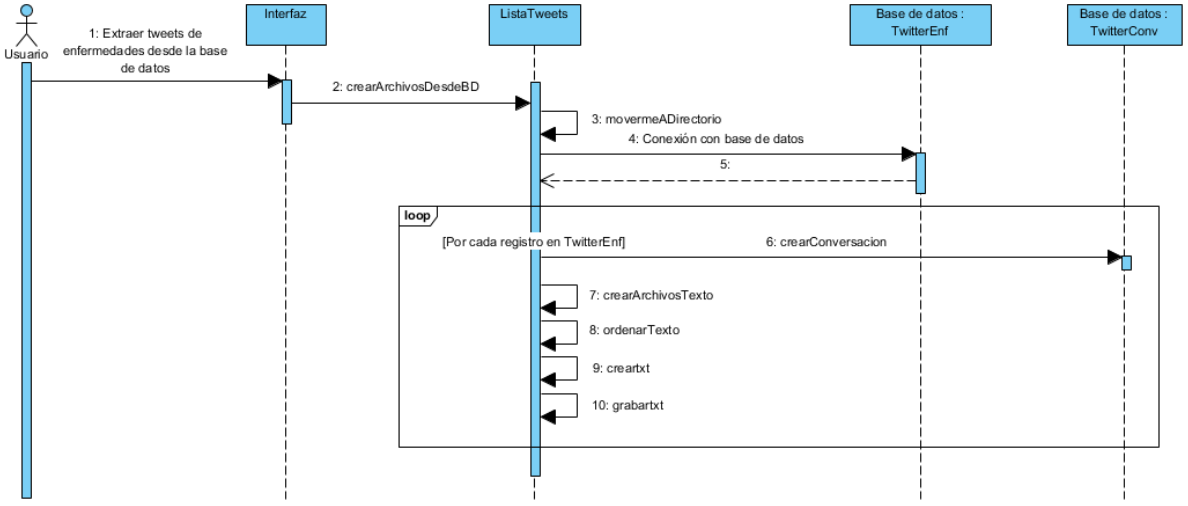


FIGURA A.81: Diagrama de secuencia: Descargar textos de Twitter por nombre de enfermedad desde la base de datos

Descargar textos de Twitter por dirección de usuarios seguidos

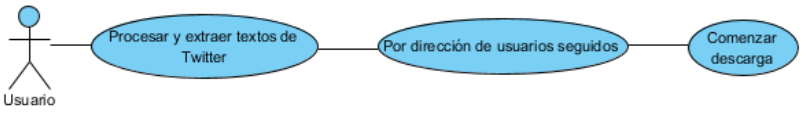


FIGURA A.82: Caso de uso extendido: Descargar textos de Twitter por dirección de usuarios seguidos

Procesar, descargar y almacenar tuits publicados por usuarios seguidos por @TFGFVIGILANCIA	
Descripción	Procesa, descarga y almacena los tuits publicados por las cuentas que sigue la cuenta @TFGFVIGILANCIA.
Actores	Usuario
Precondiciones	Existen la base de datos BDProyecto, la cuenta de Twitter @TFGFVIGILANCIA y una API de Twitter con una clave de autentificación para desarrolladores.
Poscondiciones	Las tablas TwitterDire y TwitterConv de la base de datos se han actualizado y se han descargado los tuits.

TABLA A.28: Descargar textos de Twitter por dirección de usuarios seguidos

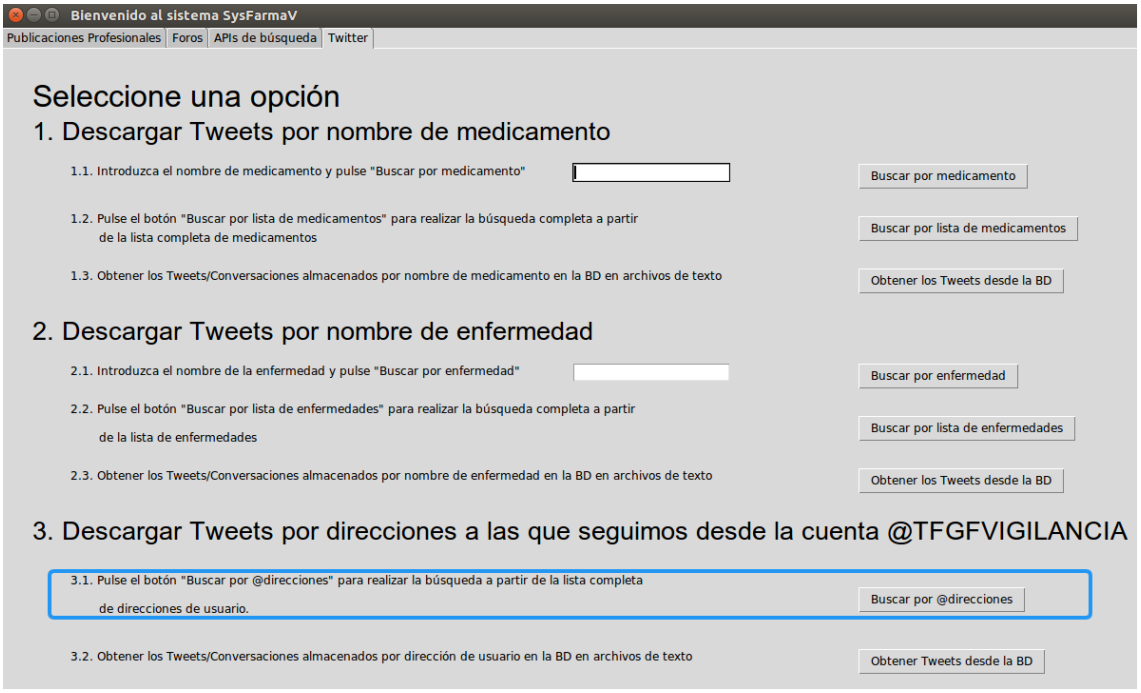


FIGURA A.83: Interfaz gráfica: Descargar textos de Twitter por dirección de usuarios seguidos

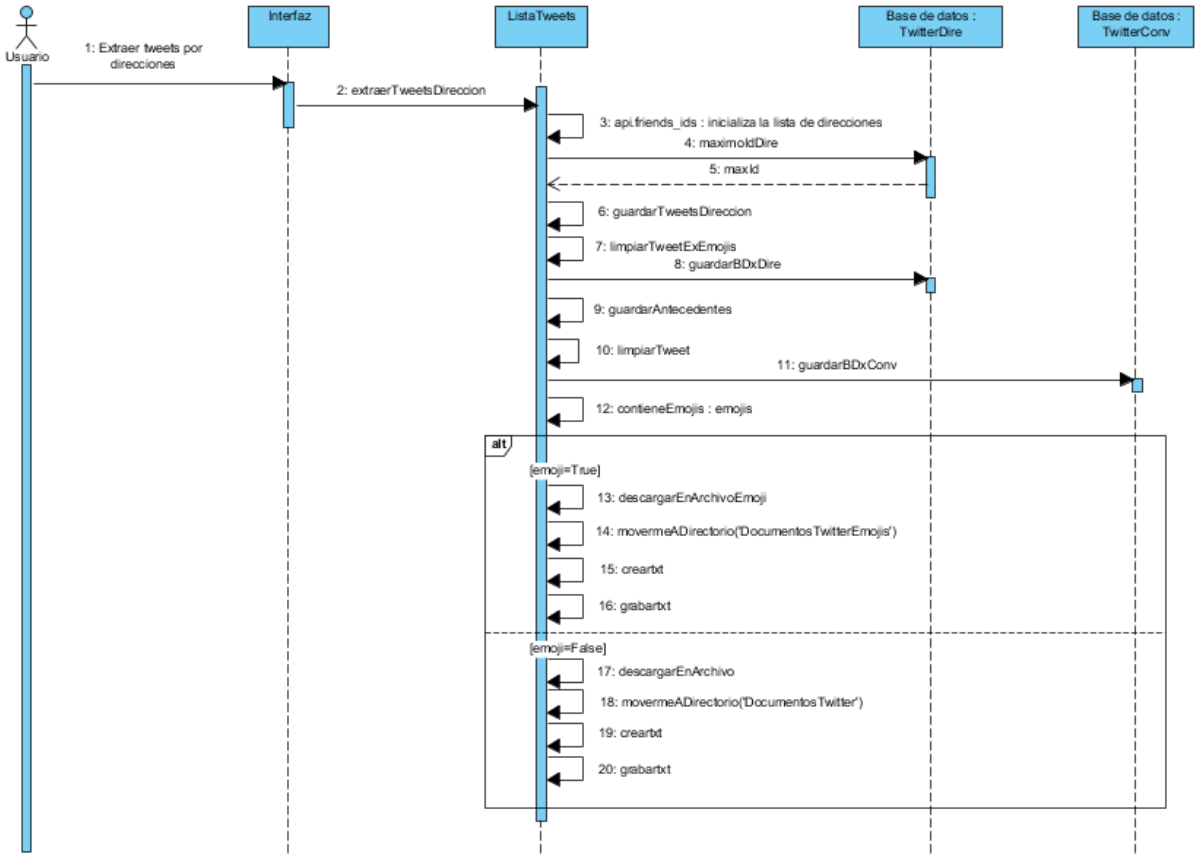


FIGURA A.84: Diagrama de secuencia: Descargar textos de Twitter por dirección de usuarios seguidos

Descargar textos de Twitter por dirección desde la base de datos

FIGURA A.85: Caso de uso extendido: Descargar textos de Twitter por dirección desde la base de datos

Convertir en documentos de texto los tuits descargados por dirección de usuarios	
Descripción	Convierte en documentos de texto los tuits almacenados en las tablas Twitter-Dire y TwitterConv.
Actores	Usuario
Precondiciones	Existe la base de datos BDProyecto y la tabla TwitterDire contiene al menos un registro.
Poscondiciones	Se crea el directorio ArchivosDB-Twitter y los registros de las tablas Twitter-Dire y TwitterConv se pasan a documentos de texto.

TABLA A.29: Descargar textos de Twitter por dirección desde la base de datos

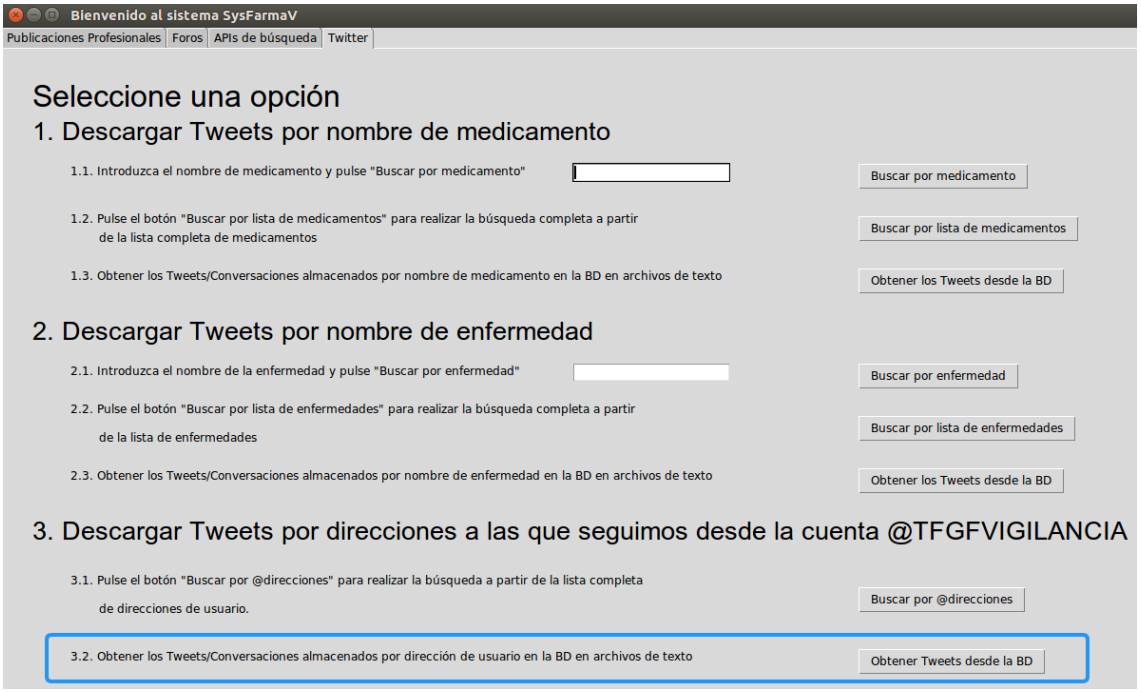


FIGURA A.86: Interfaz gráfica: Descargar textos de Twitter por dirección desde la base de datos

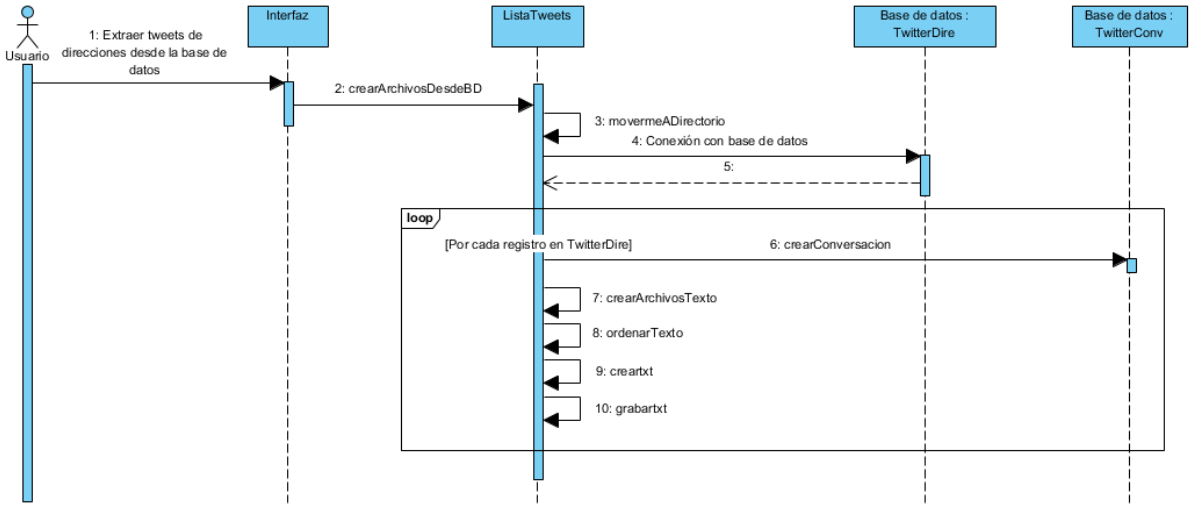


FIGURA A.87: Diagrama de secuencia: Descargar textos de Twitter por dirección desde la base de datos

Apéndice B

Anexo II. Manual de usuario

Menú

El menú hará posible acceder a las diferentes ventanas que componen la aplicación. Con el fin de una navegación más cómoda será visible en todo momento desde cualquier ventana. Dentro de cada una de estas ventanas se encontrarán las funcionalidades que lleva a cabo la herramienta.

Figura B.1: Menú de la aplicación

Publicaciones Profesionales

La ventana «Publicaciones Profesionales» contiene las funcionalidades relacionadas con páginas webs médicas predeterminadas (AEMPS y SEFH) y páginas webs genéricas.

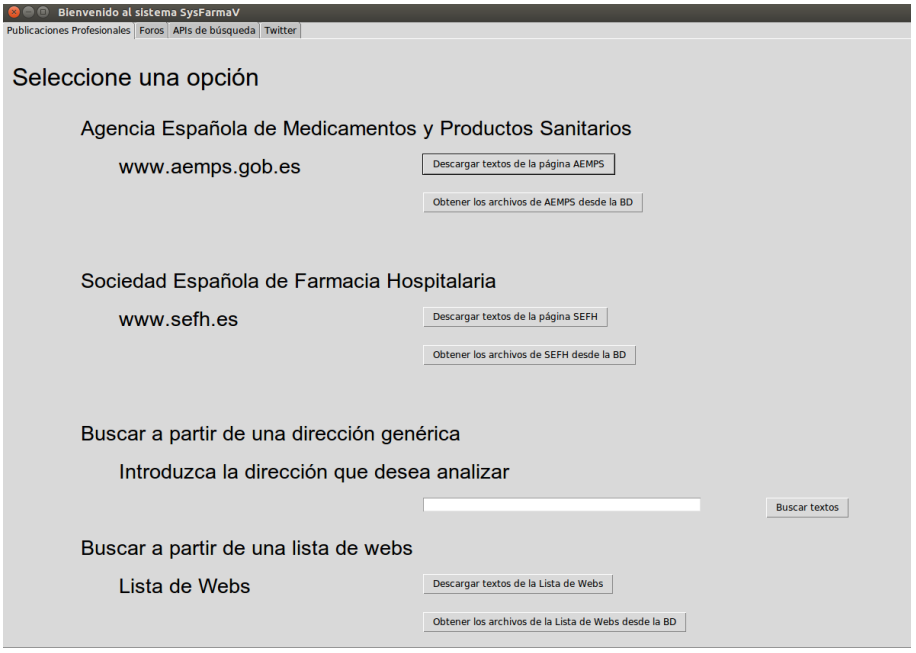


Figura B.2: Interfaz «Publicaciones Profesionales»

1. **Descargar textos AEMPS**

   Para descargar información de la página de AEMPS se deberá pulsar el botón "Descargar textos de la página AEMPS". La aplicación descargará el contenido de la página web www.aemps.gob.es que considere relevante. Todo texto correctamente procesado se descargará en la carpeta DocumentosAEMPS y se almacenará en la tabla ArchivosAEMPS de la base de datos.

2. **Descargar textos AEMPS desde la base de datos**

   Para descargar información de la página de AEMPS desde la base de datos se deberá pulsar el botón "Obtener los archivos de AEMPS desde la BD". Los registros de la tabla ArchivosAEMPS se convertirán a documentos de texto en ArchivosBD-ArchivosAEMPS.

3. **Descargar textos SEFH**

   Para descargar información de la página de SEFH se deberá pulsar el botón "Descargar textos de la página SEFH". La aplicación descargará el contenido de la página web www.sefh.es/boletin-sefh.php que considere relevante. Todo texto correctamente procesado se descargará en la carpeta DocumentosSEFH y se almacenará en la tabla ArchivosSEFH de la base de datos.

4. **Descargar textos SEFH desde la base de datos**

   Para descargar información de la página de SEFH desde la base de datos se deberá pulsar el botón "Obtener los archivos de SEFH desde la BD". Los registros de la tabla ArchivosSEFH se convertirán a documentos de texto en ArchivosBD-ArchivosSEFH.

5. **Descargar textos de una web genérica**

   Para descargar información de una web genérica se deberá escribir la dirección web de la misma y pulsar el botón "Buscar textos". La aplicación descargará el contenido de la página web que sea considerada médica. Todo texto correctamente procesado se descargará en la carpeta HTML y se almacenará en la tabla ArchivosHTML de la base de datos.

6. **Descargar textos de una lista de webs**

   I. Para comenzar, se editará el documento de texto ListaWebs.txt del directorio Proyecto/ArchivosMedicos/ incluyendo aquellas páginas de las que se quiera extraer la información. Cada una de las direcciones web deberá de ir en una única línea del documento.

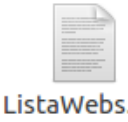


FIGURA B.3: Archivo ListaWebs.txt

   II. Ya definidas las páginas webs, se deberá pulsar el botón "Descargar textos de la Lista de Webs" para comenzar la descarga secuencial y obtención de textos médicos. Los documentos se alojarán en HTML y se almacenarán en la tabla ArchivosHTML de la base de datos.

7. **Descargar textos de webs genéricas desde la base de datos**

   Para descargar información de webs genéricas desde la base de datos se deberá pulsar el botón "Obtener los archivos de la Lista de Webs desde la BD". Los registros de la tabla ArchivosHTML se convertirán a documentos de texto en ArchivosBD-ArchivosHTML.

Foros

La ventana «Foros» contiene las funcionalidades relacionadas con foros y blogs médicos predeterminados (HospitalClinic y DMedicina) y foros genéricos.

Figura B.4: Interfaz «Foros»

1. **Descargar textos HospitalClinic**

   Para descargar información del blog HospitalClinic se deberá pulsar el botón "Descargar textos de la página HospitalClinic". La aplicación descargará el contenido de blog.hospitalclinic.org/es/ que considere relevante. Todo texto correctamente procesado se descargará en la carpeta DocumentosHospitalClinic y se almacenará en la tabla ArchivosHospitalClinic de la base de datos.

2. **Descargar textos HospitalClinic desde la base de datos**

   Para descargar información de la página de HospitalClinic desde la base de datos se deberá pulsar el botón "Obtener los archivos de HospitalClinic desde la BD". Los registros de la tabla ArchivosHospitalClinic se convertirán a documentos de texto en ArchivosBD-ArchivosHospitalClinic.

3. **Descargar textos DMedicina**

   Para descargar información del foro DMedicina se deberá pulsar el botón "Descargar textos de la página DMedicina". La aplicación descargará el contenido de www.dmedicina.com que considere relevante. Todo texto correctamente procesado se descargará en la carpeta DocumentosDMedicina y se almacenará en la tabla ArchivosDMedicina de la base de datos.

4. **Descargar textos DMedicina desde la base de datos**

   Para descargar información de la página de DMedicina desde la base de datos se deberá pulsar el botón "Obtener los archivos de DMedicina desde la BD". Los registros de la tabla ArchivosDMedicina se convertirán a documentos de texto en ArchivosBD-ArchivosDMedicina.

5. **Descargar textos de una lista de foros**

   I. Para comenzar, se editará el documento de texto ListaForos.txt del directorio Proyecto/ArchivosMedicos/ incluyendo aquellos foros de los que se quiera extraer la información. Cada una de las direcciones web deberá de ir en una única línea del documento.

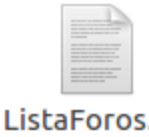


FIGURA B.5: Archivo ListaForos.txt

II. Ya definidos las direcciones de los foros, se deberá pulsar el botón "Descargar textos de la Lista de Foros" para comenzar la descarga secuencial y obtención de textos médicos. Los documentos se alojarán en HTML y se almacenarán en la tabla ArchivosForoHTML de la base de datos.

6. **Descargar textos de foros desde la base de datos**

   Para descargar información de foros desde la base de datos se deberá pulsar el botón "Obtener los archivos de la Lista de Foros desde la BD". Los registros de la tabla ArchivosForoHTML se convertirán a documentos de texto en ArchivosBD-ArchivosForoHTML.

APIs de búsqueda

La ventana «APIs de búsqueda» contiene las funcionalidades relacionadas con los dos motores de búsqueda (Google y Bing). Las cuatro acciones que existen para cada motor de búsqueda se llevan a cabo de la misma forma, por lo que se dará una explicación de cada una de ellas valiendo tanto para Google como Bing.

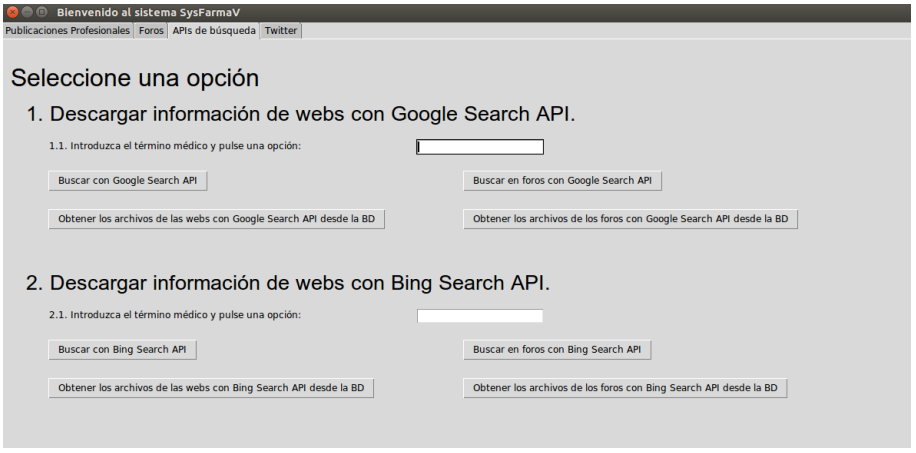


FIGURA B.6: Interfaz «APIs de búsqueda»

1. **Descargar textos de webs con un motor de búsqueda**

   I. Se colocará el término médico que se desea buscar en la entrada de texto.

   II. Una vez definido el término a buscar se deberá pulsar "Buscar con Google Search API" o "Buscar con Bing Search API" para comenzar el proceso. El motor de búsqueda devolverá las direcciones webs de donde se extraerá la información. Estos textos se alojarán en API y se almacenarán en la tabla ArchivosAPI.

2. **Descargar textos de foros con un motor de búsqueda**

   I. Se colocará el término médico que se desea buscar en la entrada de texto.

II. Una vez definido el término a buscar se deberá pulsar “Buscar en foros con Google Search API” o “Buscar en foros con Bing Search API” para comenzar el proceso. El motor de búsqueda devolverá las direcciones de los foros de donde se extraerá la información. Estos textos se alojarán en API y se almacenarán en la tabla ArchivosForoAPI.

3. **Descargar textos de webs de un motor de búsqueda desde la base de datos**

   Para descargar información de webs de un motor de búsqueda desde la base de datos se deberá pulsar el botón “Obtener los archivos de las webs con Google Search API desde la BD” u “Obtener los archivos de las webs con Bing Search API desde la BD”.

   I. Si la opción es la de GoogleSearch, se convertirán en textos aquellos registros de la tabla ArchivosAPI que se hayan obtenido con este motor de búsqueda en ArchivosBD-ArchivosAPI.

   II. Si la opción es la de Bing, se convertirán en textos aquellos registros de la tabla ArchivosAPI que se hayan obtenido con este motor de búsqueda en ArchivosBD-ArchivosAPI.

4. **Descargar textos de foros de un motor de búsqueda desde la base de datos**

   Para descargar información de foros de un motor de búsqueda desde la base de datos se deberá pulsar el botón “Obtener los archivos de los foros con Google Search API desde la BD” u “Obtener los archivos de los foros con Bing Search API desde la BD”.

   I. Si la opción es la de GoogleSearch, se convertirán en textos aquellos registros de la tabla ArchivosForoAPI que se hayan obtenido con este motor de búsqueda en ArchivosBD-ArchivosForoAPI.

   II. Si la opción es la de Bing, se convertirán en textos aquellos registros de la tabla ArchivosForoAPI que se hayan obtenido con este motor de búsqueda en ArchivosBD-ArchivosForoAPI.

Twitter

La ventana «Twitter» contiene las funcionalidades relacionadas con la red social. En esta ventana se podrán recuperar tuits que traten de temas médicos de usuarios de Twitter.

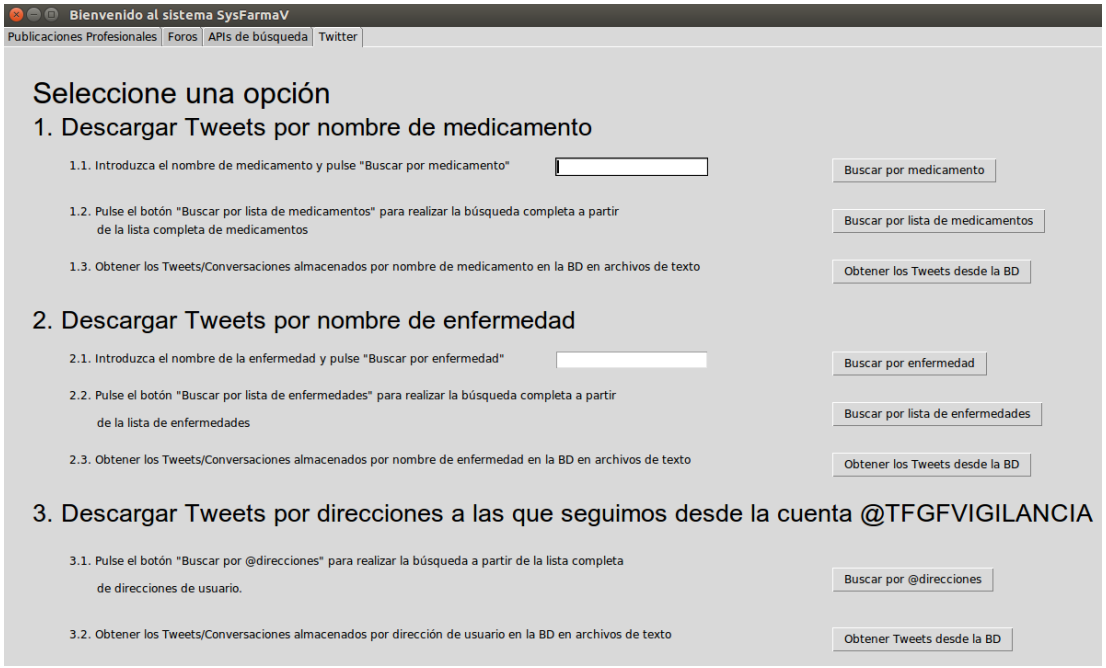


Figura B.7: Interfaz «Twitter»

1. **Descargar tuits por nombre de medicamento/enfermedad**

   I. Se colocará el término médico que se desea buscar en la entrada de texto.

   II. Una vez definido el término a buscar se deberá pulsar "Buscar por medicamento" o "Buscar por enfermedad" para comenzar el proceso. Se hará una recopilación de tuits que contengan ese término, los textos se alojarán en DocumentosTwitter o DocumentosTwitterEmoji en caso de que los tuits contengan algún *emoji* y se almacenarán en la tabla TwitterMed/TwitterEnf de la base de datos.

2. **Descargar tuits por lista de medicamentos/enfermedades**

   Para descargar tuits de una lista de medicamentos/enfermedades se deberá pulsar el botón "Buscar por lista de medicamentos" o "Buscar por lista de enfermedades". Se hará una recopilación de tuits que contenga algún término de la lista, los textos se alojarán en DocumentosTwitter o DocumentosTwitterEmoji en caso de que los tuits contengan algún *emoji* y se almacenarán en la tabla TwitterMed/TwitterEnf de la base de datos.

3. **Descargar tuits de medicamentos/enfermedades desde la base de datos**

   Para descargar los tuits almacenados de medicamentos/enfermedades en la base de datos se deberá pulsar el botón "Obtener los Tweets desde la BD". Dependiendo el apartado (medicamento o enfermedad) en el que se pulse el botón se convertirán a documentos de texto los registros de la tabla TwitterMed/TwitterEnf y TwitterConv en la carpeta ArchivosBD-TwitterMed/ArchivosBD-TwitterEnf.

4. **Descargar tuits por direcciones**

   Para descargar los tuits de los usuarios que sigue la cuenta @TFGFVIGILANCIA se deberá pulsar el botón "Buscar por @direcciones". Se hará una recopilación de tuits publicados por las cuentas relacionadas con la medicina que siga la cuenta. Estos tuits se alojarán en DocumentosTwitterX-Direccion o DocumentosTwitterXDireccionEmojis en caso de que los tuits contengan algún *emoji* y se almacenarán en la tabla TwitterDire de la base de datos.

5. **Descargar tuits por direcciones desde la base de datos**

   Para descargar los tuits almacenados de usuarios seguidos por @TFGFVIGILANCIA se deberá pulsar el botón "Obtener Tweets desde la BD". Se convertirán a documentos de texto los registros de la tabla TwitterDire y TwitterConv en la carpeta ArchivosBD-TwitterDire.

Apéndice C

Anexo III. Manual de instalación

1. Descomprimir el proyecto.
    - *unzip Proyecto2019JonAnderHierro.zip*
2. Comprobar versión de Python.
    - *python -V*
    - *sudo apt-get install python-tk*
    - *sudo apt-get install python-pip python-dev build-essential*
3. Instalar librerías genéricas de Python.
    - *sudo -H pip install –upgrade pip*
    - *sudo pip install BeautifulSoup*
    - *sudo pip install bs4*
    - *sudo pip install html2text*
    - *sudo pip install pdfminer*
    - *sudo pip install pypdf*
    - *sudo pip install tweepy*
    - *sudo pip install ftfy*
    - *sudo apt-get install unicode-data*
    - *sudo pip install nltk*
    - *sudo pip install sklearn*
    - *sudo pip install pickle-mixin*
    - *sudo pip install numpy*
    - *sudo pip install stripogram*
    - *sudo pip install simplejson*
    - *sudo pip install joblib*
    - *sudo pip install selenium*
4. Instalar librerías de Google.
    - *sudo pip install google*

- *sudo pip install google-api-python-client*

5. En caso de que la *subscription key* de Bing haya caducado:

   - Seguir las siguientes instrucciones de [26] para crear una cuenta gratuita en Azure y generar una *subscription key* en el recurso *Cognitive Services*.
   - Generar la *subscription key*.

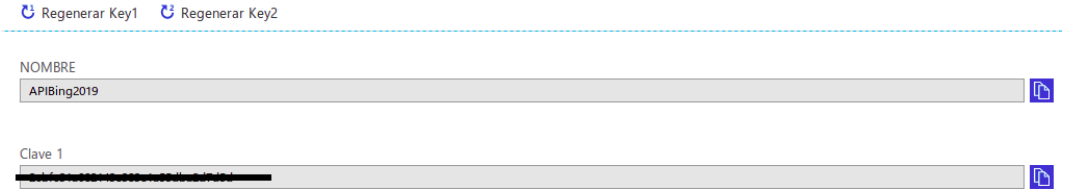


FIGURA C.1: Generación de la *subscription key*

   - Colocar la subscription key generada en el archivo Proyecto/Bing.py

FIGURA C.2: Lugar de la *subscription key* en el programa

6. Instalar librerías de Bing.

   - *virtualenv mytestenv*
   - *cd mytestenv*
   - *source bin/activate*
   - *python -m pip install azure-cognitiveservices-search-websearch*
   - *sudo pip install azure-cognitiveservices-search-websearch*

7. Dar permiso de ejecución a geckodriver para la navegación.

   - *sudo chmod +x /path/Proyecto/geckodriver/geckodriver*

8. Acceder al proyecto y ejecutar.

   - *cd Proyecto2019JonAnderHierro/*
   - *cd Proyecto/*
   - *sudo python SysFarmaV.py*

Bibliografía

[1] Raúl Villegas Beltrán. Ciclo de validación de una aplicación informática. *Universitat oberta de Catalunya*, 2013.

[2] Kwang Leng Goh, Ravi Kumar, Ashutosh Singh, and Rajendra Dash. PyBot: An Algorithm for Web Crawling. 2011.

[3] Alma Sainz-Maza Cañive. Sistema de búsqueda, descarga y procesamiento masivo de textos relacionados con la farmacovigilancia a partir de páginas web, foros y redes sociales. *Universidad del País Vasco*, 2016.

[4] Centro de Investigaciones Sociológicas. Tres problemas principales. 2019.

[5] Genbeta. Eclipse IDE. 2014.

[6] Enrique Oriol. PyDev – Eclipse como IDE de Python o Django. *Blog Enrique Oriol*, 2015.

[7] Rubén Velasco. DB Browser for SQLite, la forma más fácil de crear y editar bases de datos SQLite. *RedesZone*, 2018.

[8] Eugenia Bahit. Capítulo 10. Un paseo por los módulos de la librería estándar. *Uniwebsidad*, 2013.

[9] Michelle Núñez Galindo and Diana Dueñas Chávez. Creación y edición de documentos con LaTeX. *Universidad Nacional Autónoma de México*, 2017.

[10] Roberto Canales Mora. Modelado UML con Visual Paradigm. *Universidad Nacional Autónoma de México*, 2004.

[11] Juan Pablo Fernandez. Nuestra Misión - Software de Planificación de Proyectos que permite a los miembros del equipo contribuir en la planificación del proyecto. *PlanHammer Blog*, 2008.

[12] OBS-EDU. GanttProject: Análisis del Software. *Online Business School*, 2014.

[13] Wash Llu Graduateschua Th, Brian Pinkerton, Edward Lazowska, and David Notkin. WebCrawler: Finding What People Want. pages 24–43, 2001.

[14] Michael J. Paul, Abeed Sarker, John Brownstein, Azadeh Nikfarjam, Matthew Scotch, Karen L. Smith, and Graciela Gonzalez. Social media mining for public health monitoring and surveillance. pages 468–479, 2016.

[15] https://es.overleaf.com/learn.

[16] https://es.stackoverflow.com/.

[17] Robert Leaman, Laura Wojtulewicz, Ryan Sullivan, Annie Skariah, Jian Yang, and Graciela Gonzalez. Towards internet-age pharmacovigilance: Extracting adverse drug reactions from user posts to health-related social networks. *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 117–125, 2010.

[18] Atul Nakhasi, Ralph J Passarella, Sarah G Bell, Michael J Paul, Mark Dredze, and Peter Pronovost. Malpractice and Malcontent: Analyzing Medical Complaints in Twitter. *AAAI Tech Rep*, pages 84–85, 01 2012.

[19] Oliver A. Mcbryan. GENVL and WWWW: Tools for taming the Web. *Computer Networks and ISDN Systems*, 1994.

[20] PythonHosted. *Module google* , 2013.

[21] Aaron Hill. *Quickstart: Use the Bing Web Search SDK for Python*, 2019.

[22] Wikimedia. Wikimedia Downloads.

[23] Consumo y Bienestar Social Ministerio de Sanidad. ¿Qué es SNOMED CT? 2014.

[24] Johanna Orellana Alvear. Árboles de decisión y Random Forest. *Nederlands tijdschrift voor geneeskunde*, page 1121, 2018.

[25] Jurgen Claassen. Gold standard, not golden standard. *Universidad de Cuenca*, page 1121, 2006.

[26] Aaron Hill. Quickstart: Use the Bing Web Search SDK for Python. 2019.