Análisis de datos de periódicos digitales

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

CURSO 2015-2016

Luis Felipe de Oliveira Mesa

Director

Antonio Sarasa Cabezuelo

Facultad de Informática

Universidad Complutense de Madrid

Madrid, Junio de 2016

AUTORIZACIÓN PARA LA DIFUSIÓN DEL TRABAJO FIN DE GRADO Y SU DEPÓSITO EN EL REPOSITORIO INSTITUCIONAL E-PRINTS COMPLUTENSE

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en .....de la Facultad de ....., autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TF) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo (opcional):

- ☐ 6 meses
- ☐ 12meses

TÍTULO del TFG: .....
.....

Curso académico: 20..... / 20.....

Nombre del Alumno/s:
.....
.....

Tutor/es del TFG y departamento al que pertenece:
.....
.....
.....

Firma del alumno/s

Firma del tutor/es

Agradecimientos

Me gustaría agradecer a Antonio Sarasa por toda la ayuda prestada durante el desarrollo de la herramienta. Igualmente me gustaría agradecer por la oportunidad de estudiar en España al Consejo Nacional de Desenvolvimiento Científico y Tecnológica, que es un órgano del Ministerio de la Ciencia, Tecnología e Innovación, que promueve la investigación en Brasil. También agradezco a mi familia que siempre me dio el soporte necesario para que yo pueda seguir trabajando en este proyecto.

Abstract

It was developed in this project a web Application which main objective is to offer the user data from text analysis of digital newspaper news. The application allows the user to search for custom subjects and configure the text analysis over the data retrieved. Among these analysis, the sentiment analysis stands out. For this analysis the tool allow the user to upload custom dictionaries of words and weights, which will be used to customize the search.

The tool has been developed using the Python programming language and the web Framework called Django. The storage of the application has been done using a NoSQL MongoDB database.

Keywords: *digital newspaper, data analysis, big data, python*

Resumen

En este proyecto se ha desarrollado una aplicación Web cuya finalidad es ofrecer al usuario datos provenientes del análisis de texto de las noticias que se encuentran en periódicos online. La aplicación permite al usuario realizar búsquedas personalizadas sobre temáticas específicas y configurar algunos tipos de análisis sobre la información recuperada. Entre los análisis que son llevados a cabo destaca el análisis del sentimiento. Para ello se ofrece la posibilidad de que el usuario utilice sus propios diccionarios de pares palabra-valor, utilizados para realizar este tipo de análisis.

Para la codificación de la herramienta, se ha utilizado el lenguaje de programación Python y la framework web Django. El almacenamiento de la información de la aplicación se ha realizado sobre una base de datos NoSQL de tipo MongoDB.

Palabras clave: *periódico online, análisis de datos, big data, python.*

Índice

Índice de Figuras

CAPÍTULO 1: Introducción

Todos los días son creados 2.5 quintillones de bytes de información, tanto que el 90% de los datos en el mundo actualmente ha sido creado en los últimos dos años. El máximo aceptado generalmente es que los datos estructurados representan solamente un 20% de la información generada en una organización. La información textual está incluida en los 80% de los datos no estructurados, teniendo una grande importancia para las compañías. Esta información se expresa de diferentes formas cómo tweets, publicaciones, noticias de periódicos, etc.

Los objetivos de este trabajo han sido:

- Recuperar noticias y su contenido de los periódicos El Mundo y El Pais.
- Ofrecer un motor de búsqueda en el contenido de las noticias.
- Crear una aplicación Web que permita seleccionar noticias de acuerdo a varias temáticas y poder analizar su contenido.
- Proveer acceso a datos estadísticos a respecto de categorías y palabras contenidas en noticias.
- Realizar un análisis de sentimiento de las noticias recuperadas.
- Realizar búsqueda de noticias agrupadas por temas de violencia, enfermedad y economía, y su respectivo grado.

El desarrollo de la aplicación se ha realizado en varias fases.

En la primera fase se realizó la especificación de la aplicación, fase la cual se analizaron las principales funcionalidades de la aplicación a ser desarrollado y qué tecnologías podrían utilizarse. Se fue planteado el desarrollo aplicación Web que permitiera hacer una búsqueda más inteligente que la convencional en el contenido de noticias de los periódicos digitales El Mundo y El País. También fueron investigadas las herramientas antecedentes similares, donde la principal es el Google News. A continuación se realizó el diseño y implementación de la herramienta.

El plan de trabajo de este proyecto es descrito por las actividades de:

- Concepción de la idea de proyecto y de las tecnologías a ser utilizadas;
- Descripción sencilla de los requisitos básicos de la idea originaria de la fase de concepción;
- Investigación acerca de aplicaciones antecedentes y sus funcionalidades;

- Desarrollo de la aplicación, se realizando paralelamente cambios y testes en los requisitos;
- Escritura de la memoria del proyecto;

CHAPTER 1: Introduction

Every day around 2.5 quintillion bytes of data are created, so much that 90% of the data in the world has been created during the last two years. The maximum generally accepted is that the structured data represents only around 20% of the data produced in an organization. The text data is included in the 80% of the non-structured data, holding great value for the organizations. That data are expressed in different forms such as tweets, posts, news, etc.

The goals of this course project has been:

- Retrieve news and their content from El Mundo and El País digital newspaper.
- Offer a search engine for the news content.
- Create a Web application that allows the user to select news according to various subjects and to analyze their content.
- Provide access to statistic data related to categories and included words in news.
- Execute a sentimental analysis in retrieved news.
- Search for news grouped by subjects such as violence, disease and economy.

The application development has been done in several phases. In the first one it was made the application specification, in which phase the main possible functionalities and the available technologies were analyzed. It was planned to develop a Web application that would allow the user to execute a more intelligent search in the news content published by the digital newspapers El Mundo and El Pais. It was also researched the similar antecedent applications, which the main application is Google News. Following this, the design and implementation of the tool started.

- The work plan for this project is described by the following activities:
- Conception of the ideas of project and technologies to use.
- Simple description of basic requirements discovered in the conception phase.
- Research about the antecedent applications and their functionalities.
- Development of the application, doing at the same time tests and changes in the project requirements whenever is needed.
- Writing the project report.

CAPÍTULO 2: Especificación de la aplicación

Casos de Uso

Durante la planificación fueron definidos los casos de uso de la herramienta los cuales son mostrados en el diagrama de casos de uso siguiente.

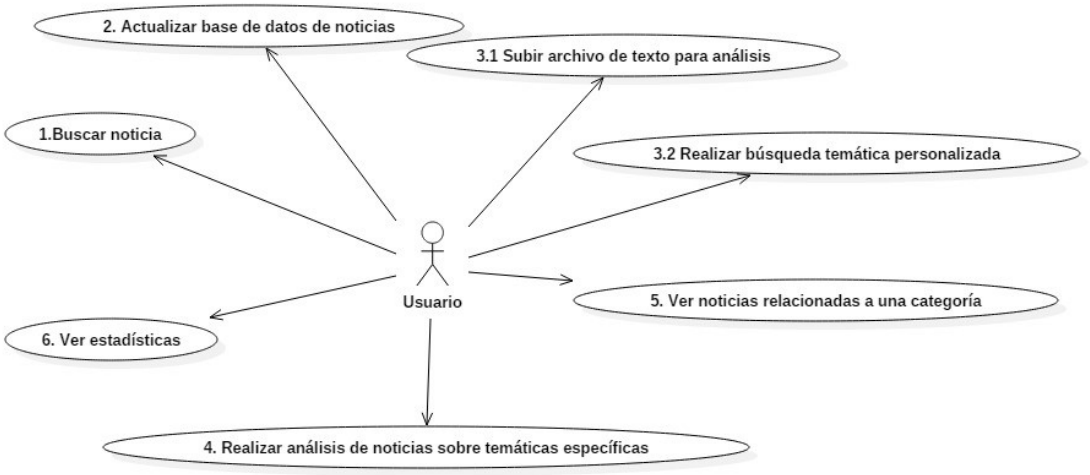


Figura 2.1 Diagrama de casos de uso

Funcionalidad: Buscar noticia

Identificador del caso de uso	CU 1
Nombre	Buscar noticia
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La lista con las noticias que contengan las palabras clave buscadas ha de mostrarse por pantalla.
Descripción	Este caso de uso describe la funcionalidad de realización de búsqueda de noticias relacionadas a palabras clave insertadas en el motor de búsqueda de la herramienta.
Flujo principal	1. El usuario pulsa sobre la caja de texto que se encuentra en el menú lateral. 2. El usuario introduce sus datos de búsqueda y efectúa la búsqueda pulsando en “Buscar”. 3. Se muestra una página web con los resultados de la búsqueda.
Flujo alternativo	

Funcionalidad: Actualizar base de datos de noticias

Identificador del caso de uso	CU 2
Nombre	Actualizar base de datos de noticias
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La base de datos está actualizada con nuevas noticias añadidas.
Descripción	Este caso de uso describe la funcionalidad de actualizar base de datos de noticias de la herramienta. Noticias que ya estaban almacenadas en la herramienta y que han tenido su enlace borrado del periódico digital serán borradas de la base de datos.
Flujo principal	1. El usuario pulsa sobre botón “Alimentar” situado en el menú lateral. 2. El sistema realiza la actualización de la base de datos, no haciendo cualquier cambio en la vista de la página mientras no termina de hacerlo. 3. Se actualiza la página principal de la aplicación con las nuevas noticias añadidas.
Flujo alternativo	

Funcionalidad: Realizar búsqueda personalizada temática de noticias

Identificador del caso de uso	CU 3
Nombre	Subir archivo de texto para análisis
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La herramienta carga los pares palabra valor para permitir la realización del análisis.
Descripción	Este caso de uso describe la funcionalidad de subida de archivo en formato texto por el usuario. El archivo de texto contiene palabras y pesos que van a ser utilizados en el análisis.
Flujo principal	1. El usuario pulsa sobre el botón “Noticias relacionadas” situado en el menú lateral. 2. Se muestra la página web “Noticias relacionadas” en la pantalla. 3. Se muestra una página web con la vista “Noticias relacionadas” 4. El usuario pulsa sobre el botón “Archivo” situado al centro de la página. 5. Se muestra una ventana donde se permite elegir un archivo de texto situado en algún dispositivo de almacenamiento. 6. El usuario pulsa sobre el botón “Ok”. 7. Se muestra un cambio en la caja de texto que está al lado del botón “Subir”, mostrando el nombre del archivo de texto elegido previamente.
Flujo alternativo	

Identificador del caso de uso	CU 3.2
Nombre	Realizar búsqueda temática personalizada
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La lista con las noticias, que tengan una frecuencia mínima de las palabras clave contenidas en el archivo de texto, ha de mostrarse por pantalla.
Descripción	Este caso de uso describe la funcionalidad de realización de análisis con datos personalizados subidos por el usuario.
Flujo principal	8. El usuario pulsa sobre el botón “Upload” situado abajo y a la izquierda del contenedor de descripción de funcionalidad. 9. Se muestra una página web con los resultados de la búsqueda.
Flujo alternativo	1. El usuario pulsa sobre la caja de texto situado debajo del botón “Archivo”, introduce los datos de búsqueda y efectúa la búsqueda pulsando en “Subir”. 2. Se muestra una página web con los resultados de la búsqueda considerando la reputación relacionada a palabra buscada.

Funcionalidad: Realizar análisis sobre temáticas específicas

Identificador del caso de uso	CU 4.1
Nombre	Realizar análisis de noticias sobre temáticas específicas
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La lista con las noticias que tengan una frecuencia mínima de las palabras clave relacionadas a temáticas específicas ha de mostrarse por pantalla.
Descripción	Este caso de uso describe la funcionalidad de realización de análisis sobre temáticas de violencia, enfermedad y economía en noticias de periódicos digitales.
Flujo principal	1. El usuario pulsa sobre el botón “Análisis” situado en el menú lateral. 2. Se muestra la página web “Análisis” en la pantalla. 3. El usuario introduce sus datos de búsqueda en la caja de texto situada debajo de la descripción de la funcionalidad y pulsa el botón “Buscar”. 4. Se muestra una página web con los resultados de la búsqueda, organizados por temática.
Flujo alternativo	

Funcionalidad: Ver noticias relacionadas a una categoría

Identificador del caso de uso	CU 5.1
Nombre	Ver noticias relacionadas a una categoría
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La lista con las noticias que tengan una frecuencia mínima de las palabras clave relacionadas a temáticas específicas ha de mostrarse por pantalla.
Descripción	Este caso de uso describe la funcionalidad de realización de análisis sobre temáticas de violencia, enfermedad y economía en noticias de periódicos digitales.
Flujo principal	1. El usuario pulsa sobre el botón “Politica” situado en el menú lateral. 2. Se muestra la página web “Política” en la pantalla.
Flujo alternativo	

Funcionalidad: Ver estadísticas

Identificador del caso de uso	CU 6.1
Nombre	Ver estadísticas
Actores	Cualquier usuario que acceda al sistema
Precondiciones	Ninguna.
Postcondiciones	La lista con las estadísticas relacionadas a frecuencia de palabras y de noticias agrupadas por categoría.
Descripción	Este caso de uso describe la funcionalidad de acceso a informaciones estadísticas respecto a frecuencia de palabras en el contenido de noticias, en peticiones de búsqueda y la frecuencia de noticias clasificadas en cada una de las categorías definidas por la herramienta.
Flujo principal	3. El usuario pulsa sobre el botón “Estadística” situado en el menú lateral. 4. Se muestra la página web “Estadística” en la pantalla.
Flujo alternativo	

CAPÍTULO 3: Tecnologías utilizadas

Python

Python es un lenguaje dinámico de código abierto, de propósito general, de alto nivel y orientado a objetos. Este lenguaje fue elegido para ser utilizado en el desarrollo de este proyecto debido a sus pocos obstáculos en su entrada, ofreciendo un largo número de librerías de alta calidad, y una grande comunidad que ha ayudado con el crecimiento de la popularidad de este lenguaje. Es un lenguaje fácil de se leer y aprenderse, siendo muy utilizado para ciencia, análisis de datos e ingeniería. La versión de Python utilizada en este proyecto es la 3.5.1 y la IDE utilizada es el PyCharm.

Django

Django es un framework de desarrollo web gratuito y open source de alto nivel, escrito en Python, y que fue elegido para el desarrollo de este proyecto debido a que toma cuenta de una grande parte de los problemas del Desarrollo Web, permitiendo al programador centrarse en escribir la aplicación de manera más rápida y fácil. La versión de Django utilizada en este proyecto es la 1.9.6.

MongoDB

MongoDB es una base de datos orientada a documentos que son almacenados en BSON. (BSON es una representación binaria de JSON). MongoDB es la más popular y favorita base de datos NoSQL, y está siendo utilizado en producción por una grande lista de empresas de los más variados dominios manejando terabytes de datos de manera eficiente. Una de las más importantes diferencias con respecto a bases de datos relacionales es que no es necesario seguir un esquema. Los comandos en MongoDB se pueden realizar en una consola que está construida sobre Javascript, se realizando así las consultas en este lenguaje. Debido a estas características, MongoDB fue elegido para almacenar nuestros datos de aplicación. La versión de MongoDB utilizada en este proyecto es la 3.0.

Pymongo

Pymongo es un paquete Python que contiene herramientas para trabajar con MongoDB y es la manera recomendada para trabajar con MongoDB desde Python. La versión de Pymongo utilizada en este proyecto es la 3.2.2.

Beautiful Soup

Beautiful Soup es un paquete Python para recoger datos de archivos HTML y XML, permitiendo así la realización de Web Scraping, que es una técnica utilizada para extraer información de sitios web y que fue muy utilizada en el desarrollo de esta herramienta. La versión de Beautiful Soup utilizada en este proyecto es la 4.

Urllib.request

Urllib.request es un paquete Python que define funciones y clases que ayudan a abrir URLs, en su mayoría HTTP.

NLTK

NLTK (Natural Language Toolkit) es un paquete Python para el procesamiento de lenguaje natural. NLTK ofrece interfaces fáciles de se utiliza para más de 50 recursos corporales y lexicales cómo WordNet, junto con un conjunto de librerías para clasificación, razonamiento semántico, derivaciones, análisis sintácticos, entre otros.

Pip

Pip es un mantenedor de paquetes para Python, permitiendo realizarse instalaciones y desinstalaciones fácilmente.

CAPÍTULO 4: ARQUITECTURA DE LA APLICACIÓN

La arquitectura de la aplicación es multicapa y sigue el patrón Modelo-Vista-Controlador.

Las Views de Django tiene el rol de Controlador y los templates tienen el rol de View.

Cuando se pulsa algún botón de acceso a alguna funcionalidad se crea una petición, que es enviada para tratamiento por la View responsable que recibe el parámetro de petición y redirecciona a la página requerida. Para mostrar datos almacenados en la base de datos MongoDB, se hace una consulta a MongoDB desde la View. MongoDB va a enviar a la View los datos requeridos en formato JSON, y la View va a tratar esos datos y mostrarlos al usuario en la pantalla.

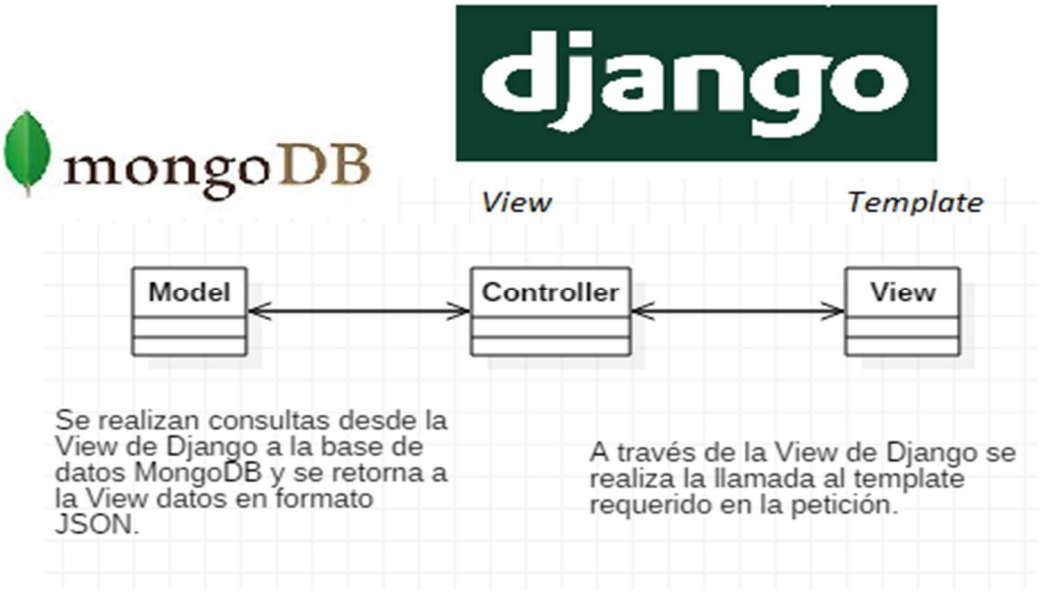


Figura 4.1 Arquitectura de la aplicación

CAPÍTULO 5: MODELO DE DATOS

MongoDB es una base de datos orientada a documentos, que son almacenadas en colecciones.

Las colecciones que son usadas en la herramienta son descritas en este apartado.

noticia

Colección creada para almacenar datos sobre las noticias, que contiene documentos descritos por 5 campos.

Título: Almacena el título de la noticia.

Periódico: Almacena el nombre del periódico.

Url: Almacena el enlace de la noticia.

Contenido: Almacena el contenido en texto de la noticia.

Categoría: Almacena la categoría de la noticia.

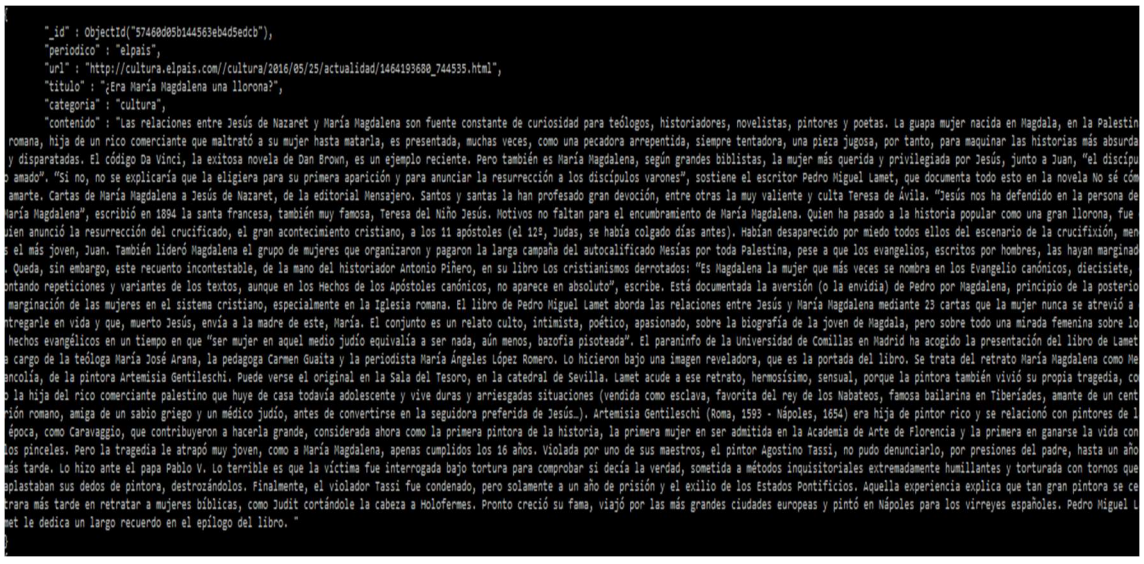


Figura 5.1 Captura de pantalla de un documento de la colección noticia.

estadística_palabras

Colección creada para almacenar datos sobre las palabras que contiene documentos descritos por 2 campos.

Palabra: Almacena una palabra

Contador: Almacena la cantidad de veces que la palabra aparece en el contenido de noticias.

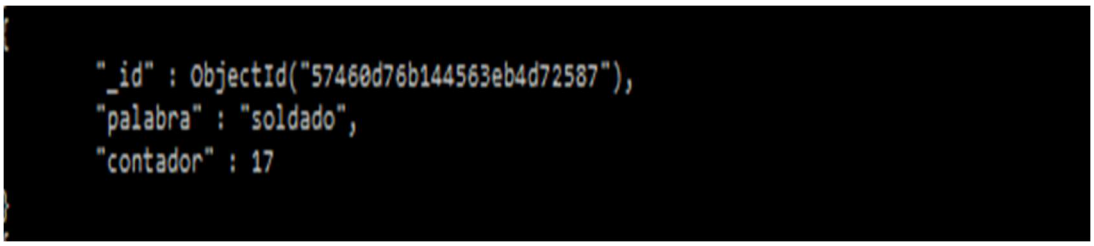


Figura 5.2 Captura de pantalla de un documento de la colección estadistica_palabras

palabras_buscadas

Colección creada para almacenar datos sobre palabras que contiene documentos descritos por 2 campos.

Palabra: Almacena una palabra

Contador: Almacena la cantidad de veces que la palabra fue buscada se utilizando la herramienta de búsqueda de la aplicación.

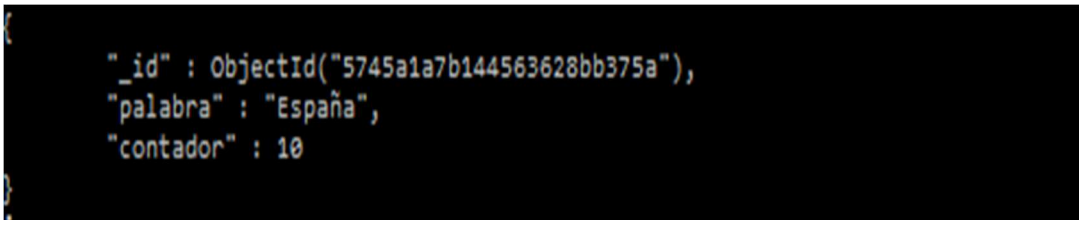


Figura 5.3 Captura de pantalla de un documento de la colección palabras_buscadas

estadistica_categorias

Colección creada para almacenar datos sobre las categorías que clasifican las noticias. Esta colección contiene documentos descritos por 2 campos.

Cateogoria: Almacena una palabra

Contador: Almacena la cantidad de veces que la palabra aparece en el contenido de noticias.

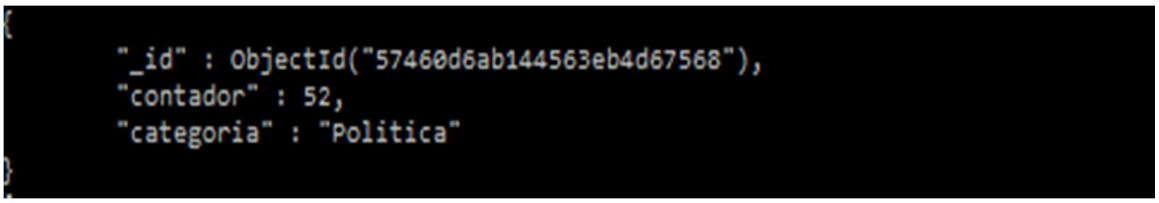


Figura 5.4 Captura de pantalla de un documento de la colección estadistica_categorias

CAPÍTULO 6: DISEÑO E IMPLEMENTACIÓN

El diseño e implementación de la herramienta fueron realizados según la funcionalidad, siendo definidas sus funciones que contienen la lógica en el archivo views.py.

Actualizar base de datos de la aplicación

Esta funcionalidad permite la actualización de la base de datos MongoDB, realizándose el Web Scraping e insertando en la base de datos MongoDB nuevos documentos que representan noticias, además de borrar noticias que no estén disponibles actualmente.

El usuario puede acceder a esta funcionalidad desde cualquier vista de la herramienta en el menú lateral, pulsando sobre el botón “Alimentar” situado en el menú lateral.

En el diseño de esta funcionalidad fue planteado el uso de Web Scraping para realizar la obtención de contenido de noticias de los periódicos digitales El País y El Mundo.

Con el uso de la librería Urllib.request se abre el enlace de cada categoría de los periódicos. Una vez obtenida la página, la función de Web Scraping realiza un bucle *for* por toda la página, almacenando todos los enlaces de noticias en una lista. Estos enlaces son encontrados debido al uso de la librería Beautiful Soup, que permite encontrar elementos HTML en una página de una manera sencilla.

```
def web_scrapper_elmundo():
    list_enlace = [['internacional','http://www.elmundo.es/internacional.html'],
                   ['politica','http://www.elmundo.es/opinion.html?cid=MENUHOM24801&s_kw=opinion'],
                   ['economia', 'http://www.elmundo.es/economia.html?cid=MENUHOM24801&s_kw=economia'],
                   ['ciencia','http://www.elmundo.es/ciencia.html?cid=MENUHOM24801&s_kw=ciencia'],
                   ['tecnologia', 'http://www.elmundo.es/tecnologia.html?cid=MENUHOM24801&s_kw=tecnologia'],
                   ['cultura', 'http://www.elmundo.es/cultura.html?cid=MENUHOM24801&s_kw=cultura'],
                   ['estilo','http://www.elmundo.es/sociedad.html?cid=MENUHOM24801&s_kw=sociedad'],
                   ['deporte', 'http://www.elmundo.es/deportes.html?cid=MENUHOM24801&s_kw=deportes']]
    client = MongoClient('localhost', 27017)
    db = client.test
    enlaceList = []
    count = 0
    for enlace_item in list_enlace:
        html = urllib.request.urlopen(enlace_item[1]).read()
        sopa = BeautifulSoup(html, "html.parser")
        contenedor_div = sopa.find("div", class_='contenedor')
        elem = contenedor_div.find_all("div", class_="elementos-texto")
        for elements in elem:
                header = elements.find("header")
                h1 = header.find("h1")
                enlace = h1.find("a")
                try:
                    content = urllib.request.urlopen(enlace['href']).read()
                    sopa = BeautifulSoup(content, "html.parser")
                    contentText = sopa.find("div", itemprop="articleBody")
                    phrases = contentText.find_all("p")
                    text=""
                    for phrase in phrases:
                        text = text + phrase.text + " "
                    text = text.replace("\n","").replace("\t","")
                    title = enlace.text
                    title = title.replace("\n","").replace("\t","")
                    if(len(text)>5) and (len(title)>5):
                        new = News(enlace.text, text, enlace['href'], enlace_item[0])
                        if new.url not in enlaceList:
```

Figura 6.1 Parte del código de la función responsable por realizar el Web Scraping de la página El Mundo

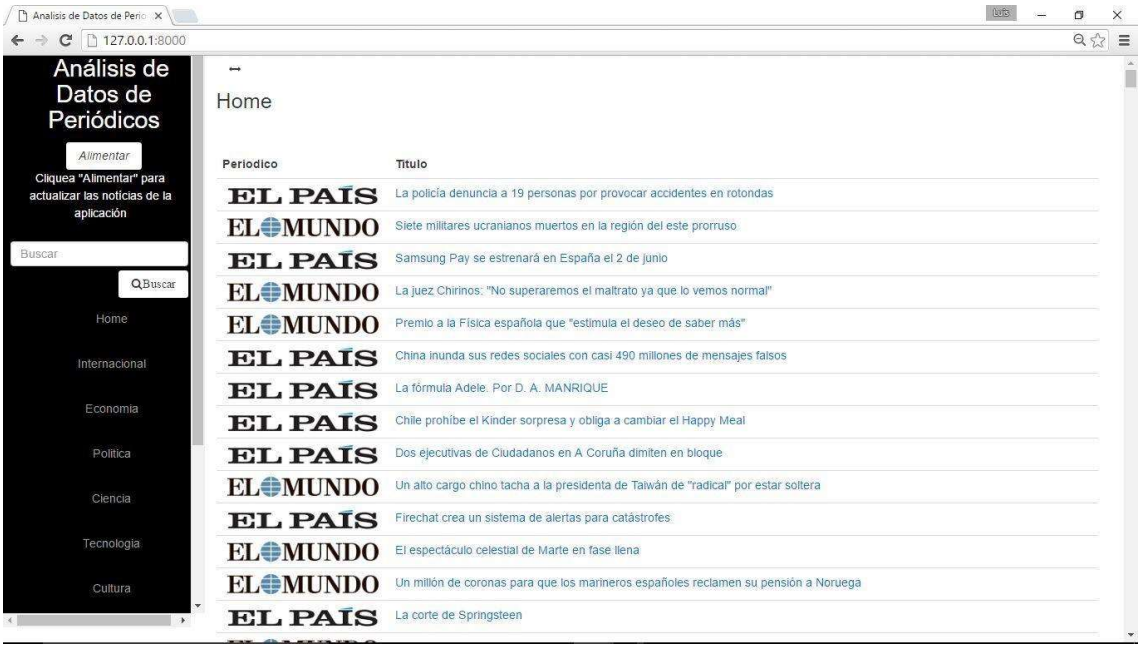


Figura 6.2 Vista de la página inicial

Buscar Noticias

El sistema te permite realizar búsqueda de noticias que contengan las palabras clave buscadas y que estén almacenadas en la base de datos MongoDB. Se retorna al usuario una vista conteniendo los resultados de la búsqueda. Se accede a esta funcionalidad.

En el diseño de esta funcionalidad fue planteado el uso de la librería pymongo para realizar la consulta a la base de datos MongoDB y retornar los resultados a la View. El usuario puede acceder a esta funcionalidad desde cualquier vista de la herramienta en el menú lateral, pulsando sobre el botón “Buscar” después de introducir datos de búsqueda en la caja de texto a su lado.

Para permitir que la aplicación haga una búsqueda de noticias que contengan todas las palabras buscadas fue utilizada la función “text” de MongoDB, que permite hacer una búsqueda textual en campos indexados con un índice de texto.

```
def search_list_view(request):
    message = ""
    query = request.GET.get('q', '')
    list_q = query.split()
    client = MongoClient('localhost', 27017)
    db = client.test
    new_q = ""
    for word in list_q:
        word = '\"' + word + '\"'
        new_q = new_q + word + ' '
    new_q = new_q[:-1]
    query_list = db.noticias.find({'$text':{'$search': new_q}})
    words = db.palabras_buscadas.find()
    words_list = []

    for word in words:
        words_list.append(word['palabra'])
    if query in words_list:
        word = db.palabras_buscadas.find_one({'palabra':request.GET.get('q', '')})
        db.palabras_buscadas.update({"_id": word["_id"]}, {'$inc':{'contador':1}})
    else:
        db.palabras_buscadas.insert_one({'palabra':request.GET.get('q', ''), 'contador':1})
    query_list = list(db.noticias.find({'$text':{'$search': new_q}}))
    return render_to_response('app/search_list_view.html', { 'search_result': query_list, 'message':message, 'word': query})
```

Figura 6.3 Parte del código de la función responsable por realizar la búsqueda textual

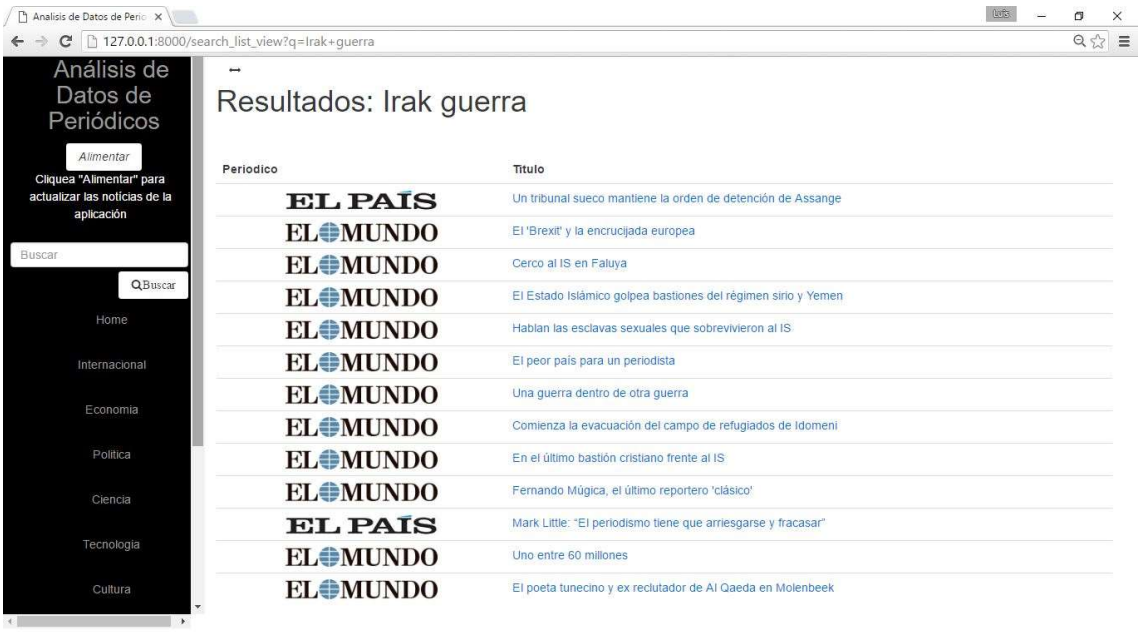


Figura 6.4 Vista de la página que ha de ser mostrada al ejecutarse la búsqueda

Análisis temáticos de noticias

Esta funcionalidad permite la búsqueda de noticias relacionadas a temáticas específicas y la palabra introducida en la búsqueda. Se retorna al usuario una vista conteniendo los resultados de la búsqueda.

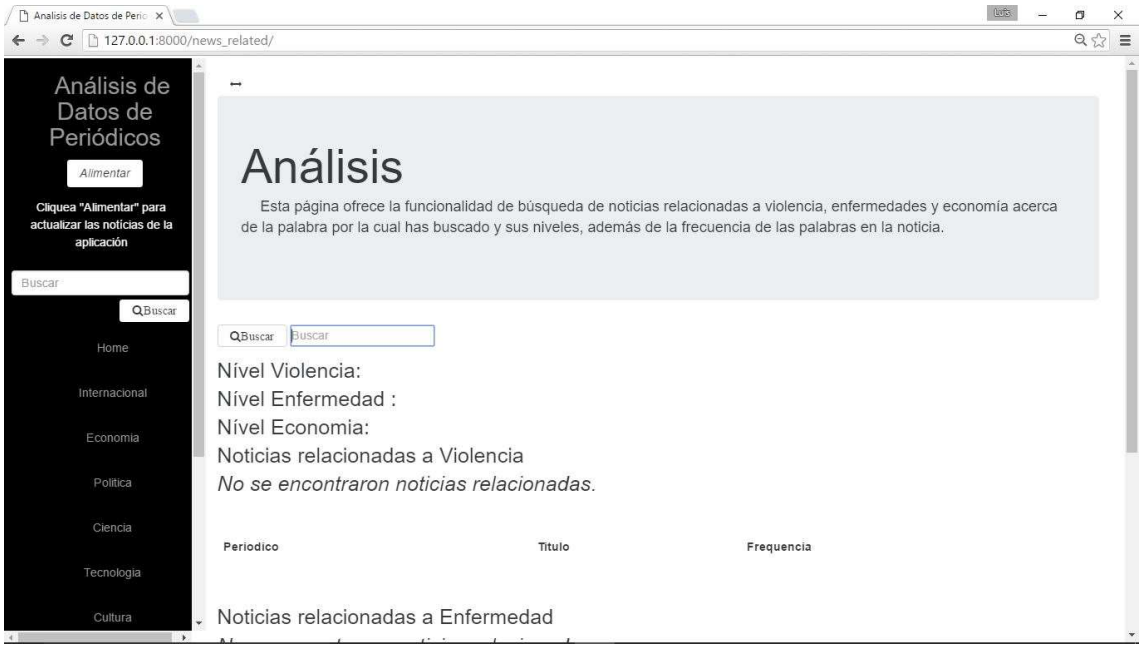
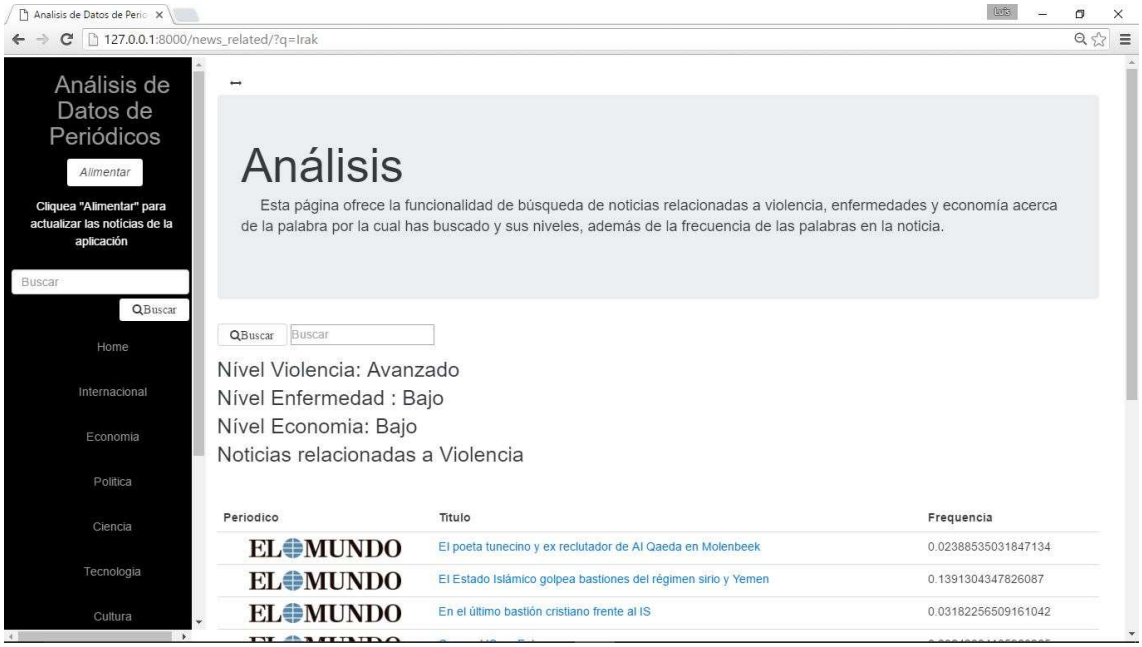


Figura 6.5 Vista inicial de la página “Análisis”



6.6 Vista de la página que se muestra al ejecutar la búsqueda por la palabra “Irak”

En el diseño de esta funcionalidad fue planteado el uso de la librería pymongo para realizar la consulta a la base de datos MongoDB y retornar los resultados a la View, además del uso de diccionarios de palabras y sus respectivos pesos para uso en el análisis.

Las tres temáticas definidas son Violencia, Enfermedad y Economía. Para la realización del análisis, se calcula el peso de la palabra en un texto, cálculo lo cual es definido por la multiplicación de la cantidad de veces que la palabra aparece en el texto y el respectivo peso definido en el diccionario, se realizando después la división por 100, obteniendo una frecuencia. Fue definida una frecuencia mínima de 0.02 cómo restricción a los resultados, permitiendo así que la aplicación muestre al usuário solamente resultados que tengan un mínimo grado de importancia.

```
if request.GET.get('q', '')!= "":
    for new in news:
        try:
            query_word = request.GET.get('q', '')
            if query_word in new['contenido']:
                content = new['contenido'].split()
                count_words = len(content)
                for word, weight in violence.items():
                    for wrd in content:
                        if wrd.startswith(word):
                            count_v = count_v + weight
                for word, weight in enfermedad.items():
                    for wrd in content:
                        if wrd.startswith(word):
                            count_e = count_e + weight
                for word, weight in economia.items():
                    for wrd in content:
                        if wrd.startswith(word):
                            count_eco = count_eco + weight
                if count_v/count_words > freq:
                    violence_news.append(News_related(new['titulo'], new['url'], count_v/count_words, new['periodi
                if count_e/count_words > freq_e:
                    enfermedad_news.append(News_related(new['titulo'], new['url'], count_e/count_words, new['perio
                if count_eco/count_words > freq_eco:
                    economia_news.append(News_related(new['titulo'], new['url'], count_eco/count_words, new['perio
                count_v = 0
                count_e = 0
                count_eco = 0
            else:
                pass
        except:
```

Figura 6.7 Parte del código utilizado para el cálculo de la frecuencia de palabras

Análisis personalizado temático de noticias

Esta funcionalidad permite al usuario subir un archivo de texto conteniendo palabras y pesos y ejecuta una función de la herramienta que realiza la búsqueda basada en los datos contenidos en el archivo de texto subido por el usuario y en la palabra clave insertada en el campo de texto de búsqueda.

Figura 6.8 Vista inicial de la página “Noticias relacionadas”

Para la realización de análisis personalizado de noticias la herramienta utiliza el mismo método que la funcionalidad análisis temático de noticias, se realizando la multiplicación de la cantidad de veces que la palabra aparece en el texto y el respectivo peso definido en el diccionario, se realizando después la suma del peso acumulado de todas las palabras y siguiente la división por 100, obteniendo una frecuencia. Se permite añadir peso negativo a las palabras, ofreciendo la posibilidad de filtrar noticias por temática de interés de usuario y excluir de los resultados noticias relacionadas a palabras que no son del interés del usuario. Fue definida una frecuencia mínima de 0.02 cómo restricción a los resultados.

Figura 6.9 Vista de la página “Noticias relacionadas” al cargar un archivo de texto y introducir una palabra en la caja de texto

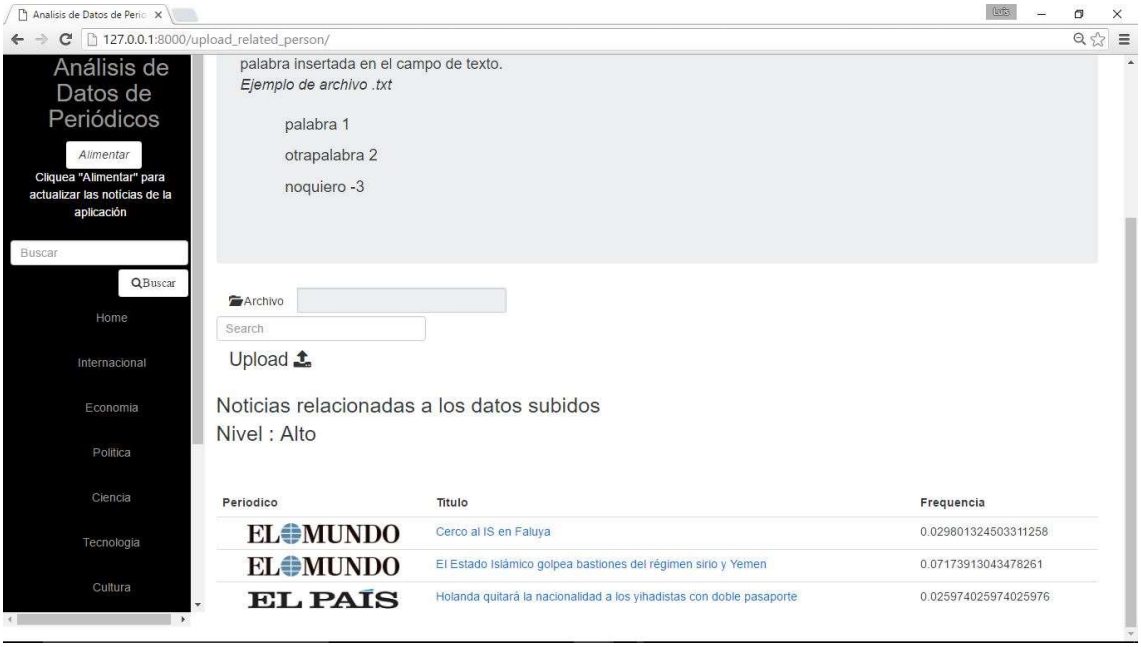


Figura 6.10 Vista de la página “Noticias relacionadas” al ejecutarse la búsqueda con archivo de texto elegido y palabra introducida

Si el usuario no elige a un archivo de texto, la herramienta realiza una búsqueda de noticias relacionadas a la reputación de la palabra introducida, según el diccionario de palabras definido para este fin.

Visualizar datos estadísticos

La aplicación permite al usuario visualizar a una vista conteniendo:

- Las diez palabras más buscadas en la aplicación en la primera columna y su respectiva cantidad en la segunda columna.
- Las categorías de noticias definidas en la aplicación en la primera columna y su respectiva cantidad de noticias relacionadas en la segunda columna.
- Las 10 palabras más populares en el contenido de noticias almacenadas en la primera columna y su respectiva cantidad en la segunda columna.

Figura 6.12 Vista de la página “Estadísticas”

CAPÍTULO 7: Conclusiones y trabajo futuro

Con la finalización del desarrollo del proyecto se han alcanzado los objetivos planteados en la planificación del proyecto.

La herramienta ejecuta de manera eficiente sus funcionalidades. Entre los puntos fuertes de la aplicación vale resaltar la funcionalidad de busca por temática específica y la búsqueda personalizada temática, que permite al usuario una mayor interacción con la plataforma, una vez que no solamente se retornan noticias relacionadas a los datos introducidos por el usuario, sino también os permite añadir sus propios datos personalizados, se tornando una herramienta potente y flexible. Cómo puntos débiles se encuentra la falta de una funcionalidad de descarga de contenidos, visto que la aplicación ofrece mucha información en pantalla, pero no permite su descarga.

Para la mejoría de la herramienta, se podría implementar una central de descargas, donde permitiría la descarga de noticias, estadísticas y resultados del análisis por temática específica y personalizado. Se podría mejorar el número de periódicos disponibles para análisis, visto que es limitado al El Mundo y El País, y permitir el análisis basado en noticias en destacado del periódico, agregando así un valor mayor al análisis una vez que noticias en destacado pueden tener un impacto diferenciado en los resultados del análisis. Todas las mejoras pensadas tienen como objetivo el desarrollo de una herramienta más completa y con una capacidad de análisis mayor.

CHAPTER 8: Conclusion and future work

The Project development is finished and the objectives which were set up in the project plan were achieved.

The tool executes in an efficient way its functionalities. Considering the strengths of the tool, it's worth to highlight the custom search for specific subjects and the totally customized subject search, which allows the user to highly interact with the platform, since that not only it retrieves the news related to the user's input data, but it allows the user to add its own custom data, turning the tool a powerful and flexible tool. As weak points it's important to say about the lack of download functionalities, since the tool doesn't allow the user to store the high amount of data from the application in a simple way such as a text file download.

For the tool improvement it would be interesting to add a download center, where the user could download the news content, statistic data and custom and specific subject results. Also it would be important to add more digital newspaper for analysis, since the tool is restricted to El Pais and El Mundo newspapers data, and to allow the analysis based on highlights news, incorporating more value to the analysis since the highlight news could have a different impact in the analysis results. All the improvements ideas described here have the development of a more complete tool and a better analysis capacity as their goals.

Bibliografía

Django documentation https://docs.djangoproject.com/en/1.9/

PyMongo documentation https://api.mongodb.com/python/current/index.html

Beautiful Soup documentation https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Urllib.request documentation https://docs.python.org/3/library/urllib.request.html

MongoDB documentation https://docs.mongodb.com/manual/

Natural Language Toolkit documentation http://www.nltk.org/

The Django Book http://www.djangobook.com/en/2.0/index.html

Web Scraping with Python, Release Date: 06/10/2015, Author: Ryan Mitchell

MongoDB: The Definitive Guide, 2nd Edition, Release date: May 2013, Author: Kristina Chodorow

ANEXO I: GUÍA DE INSTALACIÓN

Para la configuración del entorno de ejecución de la aplicación es necesario descargar algunas aplicaciones y librerías de Python. En este anexo será descrito este proceso.

Instalación de Python, MongoDB y paquetes Python

Python 3.5.1

La versión a ser descargada de Python es la 3.5.1 que puede ser descargada desde el enlace https://www.python.org/ftp/python/3.5.1/python-3.5.1.exe.

Figura A.1 Pantalla de instalación de Python

Se debe marcar la opción “Add Python 3.5 to PATH” y “Install launcher for all users (recommended)” y pulsar en “Install Now”, empezando así a instalarse Python.

Para la instalación de paquetes para Python se utiliza pip, un mantenedor de paquetes que es instalado junto a Python automáticamente.

Paquetes requeridos para instalarse y su respectivo código pip que tiene que ser introducido en la línea de comando.

Django: pip install django

Pymongo: pip install pymongo

Beautiful Soup: pip install beautifulsoup4

NumPy: pip install numpy

NLTK: Hay que descargar desde el enlace https://pypi.python.org/pypi/nltk la versión correspondiente al sistema de ejecución. Después de instalado, hay que descargar los conjuntos de palabras de nltk. Para eso, se abre el interpretador Python y se introduce nltk.download()

MongoDB: Hay que descargar desde el enlace https://www.mongodb.com/download-center la versión correspondiente al sistema de ejecución (32bits o 64 bits). Después de instalado, hay que crear una carpeta llamada "data" y dentro de esta carpeta crear una carpeta llamada "db".

Ejecución de la aplicación

Para ejecutar la aplicación se requiere que se ejecute el servidor de MongoDB (archivo mongod que se sitúa en la carpeta "MongoDB\Server\3.0\bin").

Después de tener ejecutando el servidor de MongoDB se debe abrir el Símbolo de Sistema y navegar hasta la carpeta del principal del proyecto, donde haya el archivo "manage.py" y introducir el comando "python manage.py runserver" y pulsar "Enter".

```
C:\Users\luisdeolpy\Documents\Python Scripts\AnalisisPDigitales>python manage.py runserver
```

Figura A.2 Ejecución de la aplicación

Ha de mostrarse en la pantalla de Simbolo de Sistema el mensaje descrito a seguir, con el enlace de acceso a la aplicación.

```
System check identified no issues (0 silenced).
June 13, 2016 - 20:36:02
Django version 1.9.6, using settings 'DjangoWebProject1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura A.3 Mensaje confirmando que la aplicación está en ejecución

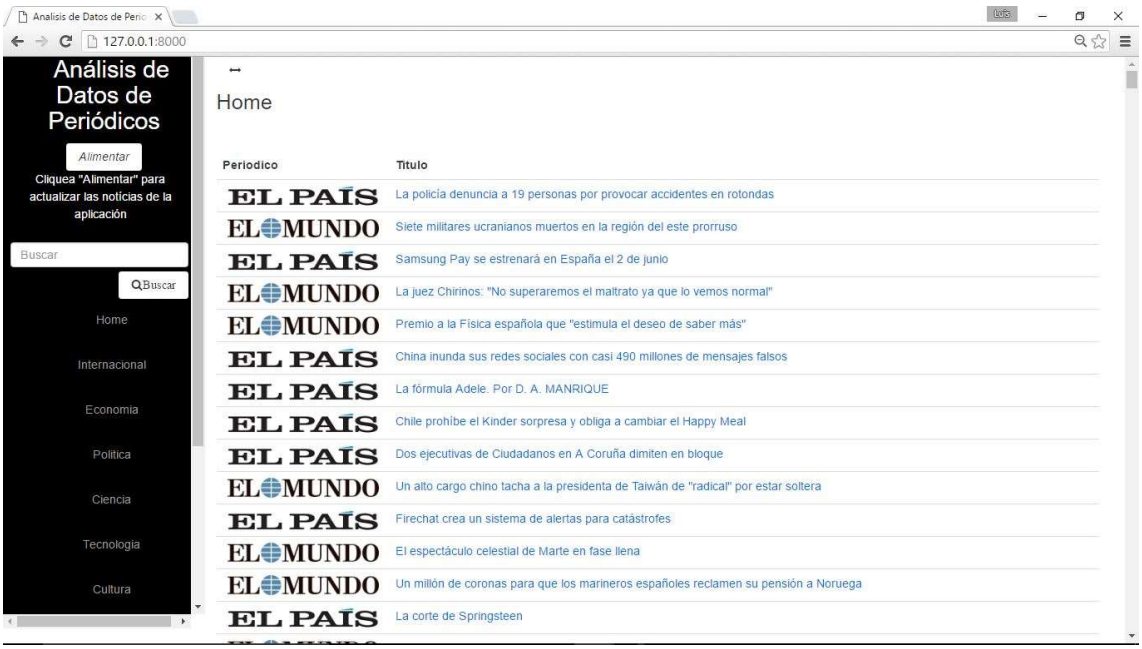


Figura A.4 Página inicial de la aplicación

ANEXO II: GUÍA DE USO

La página inicial de la aplicación ha de mostrarse al acceder a su enlace.

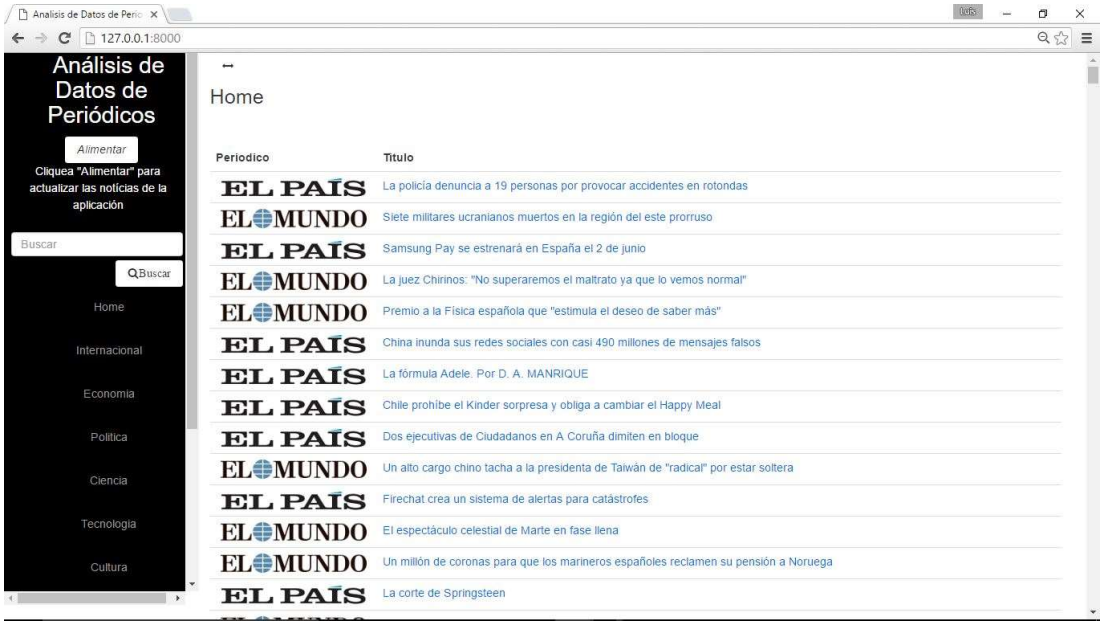


Figura U.1 Vista inicial de la aplicación

Desde esta página inicial se puede acceder a las páginas respectivas a cada una de las categorías disponibles “Internacional”, “Economía”, “Política”, “Ciencia”, Tecnología”, “Cultura”, “Deporte” y “Sociedad” al pulsar sobre el botón sobre el respectivo nombre. Cada una de ellas tiene una vista muy parecida, teniendo cómo principales diferencias el encabezado (que corresponde a la categoría) y el contenido organizado en una tabla con los títulos de las noticias relacionadas categoría y el periódico de publicación.

El menú lateral permite ocultarse al pulsar en el botón <-> que está sobre el encabezado de cada una de las páginas.

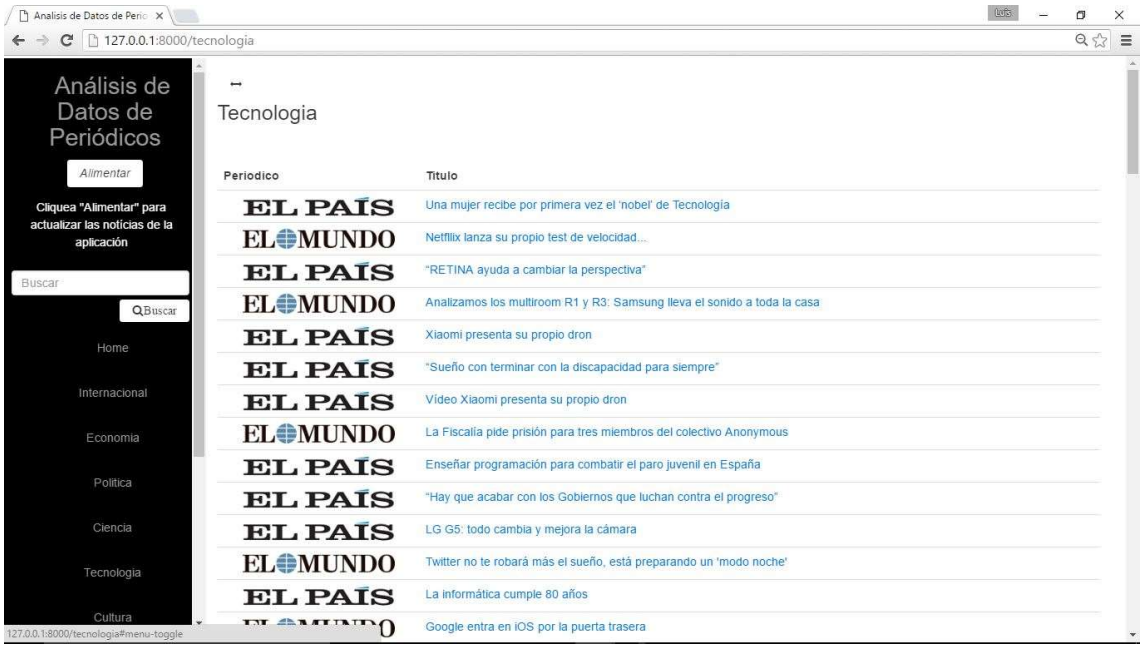


Figura U.2 Vista de la página "Tecnología"

También se puede acceder a la página Estadística, donde ha de mostrarse los datos estadísticos de la aplicación. Se muestra en esa página 3 tablas que contienen datos respecto a frecuencia de palabras en el contenido de noticias y introducidas en el motor de búsqueda, además de la frecuencia de noticias agrupadas por categoría definida en la aplicación.

Figura U.3 Vista inicial de la página "Estadística"

Al pulsar sobre el botón “Análisis” en el menú lateral, ha de mostrarse la página “Análisis”, donde hay una descripción de la funcionalidad.

Figura U.4 Vista inicial de la página “Análisis”

Se permite introducir una palabra en la caja de texto que está al lado del botón “Buscar”. Al pulsar sobre el botón “Buscar”, la aplicación te retornará las noticias de temática de violencia, enfermedad y economía relacionadas a la palabra introducida.

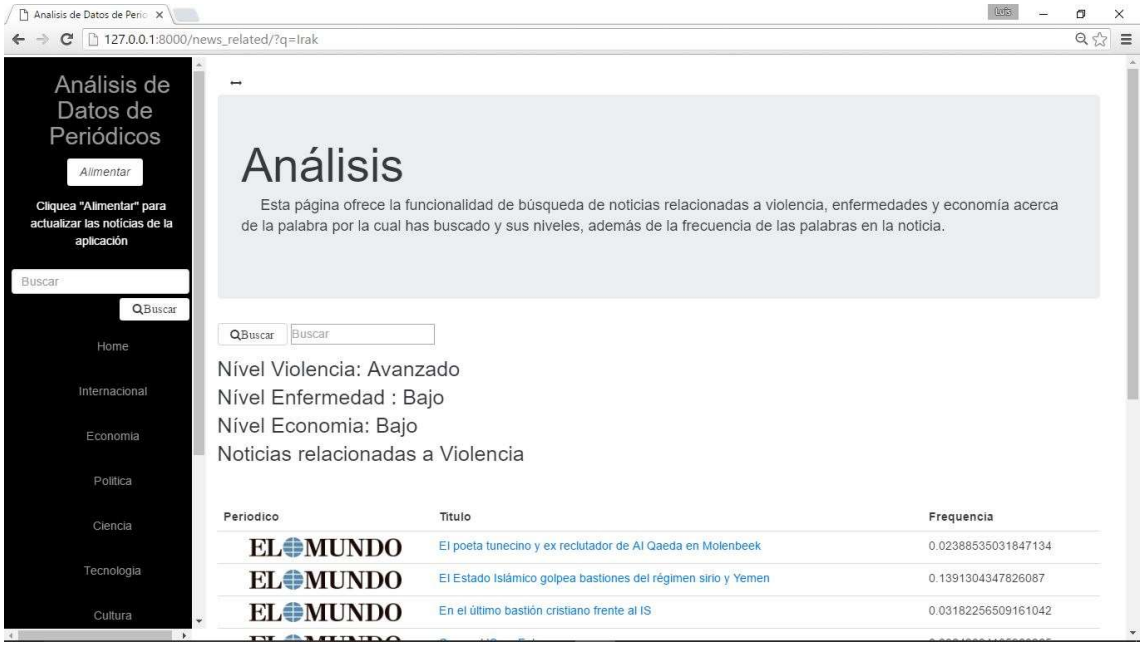


Figura U.5 Vista retornada al buscar por la palabra “Irak” en la página “Análisis”

Al pulsar sobre el botón “Noticias relacionadas” en el menú lateral, ha de mostrarse la página “Noticias relacionadas” en la pantalla, donde hay una descripción de la funcionalidad y una lista de noticias relacionadas a reputación positiva.

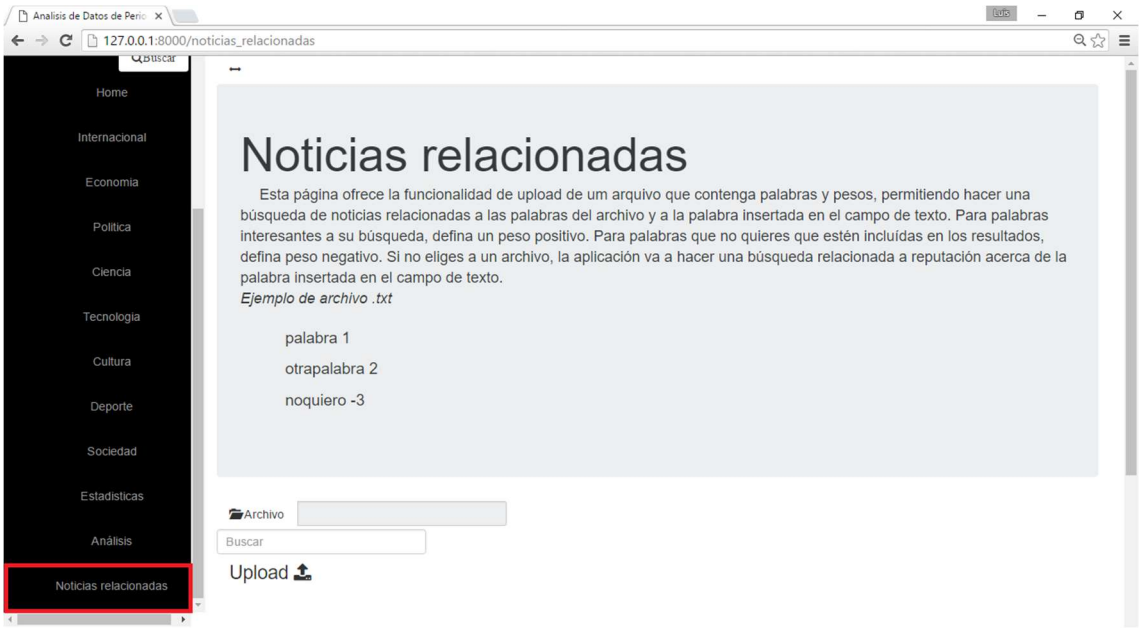


Figura U.6 Vista inicial de la página “Noticias relacionadas”

Se permite introducir un texto en la caja de texto que está sobre el botón “Upload”. También se permite añadir un archivo de texto que contenga pares de palabra y su respectivo peso a fin de realizar una búsqueda personalizada por el usuario. Se definen pesos positivos o negativos para cada palabra. Si se atribuye peso negativo a una palabra, la herramienta disminuirá la importancia de noticias relacionadas a esta palabra, permitiendo una búsqueda flexible.

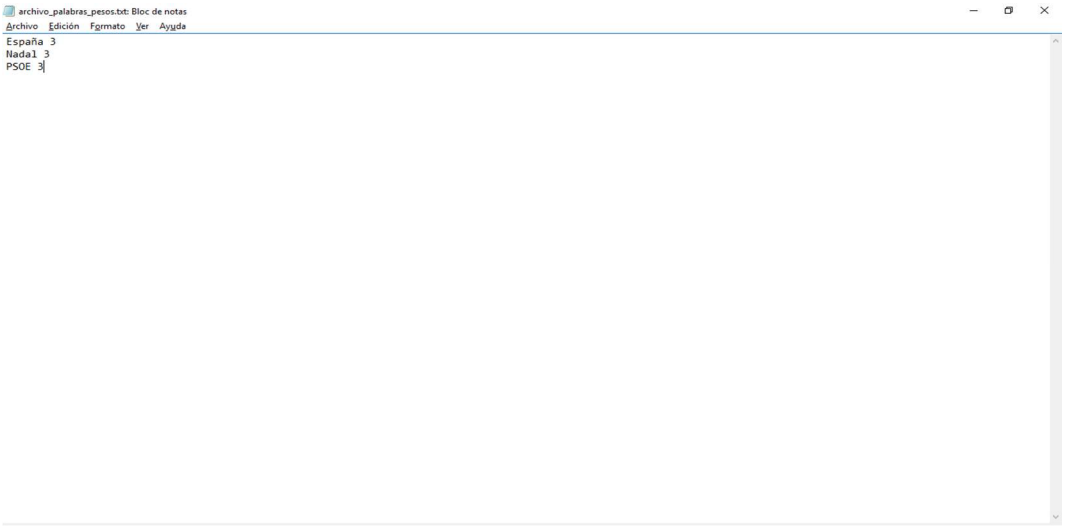
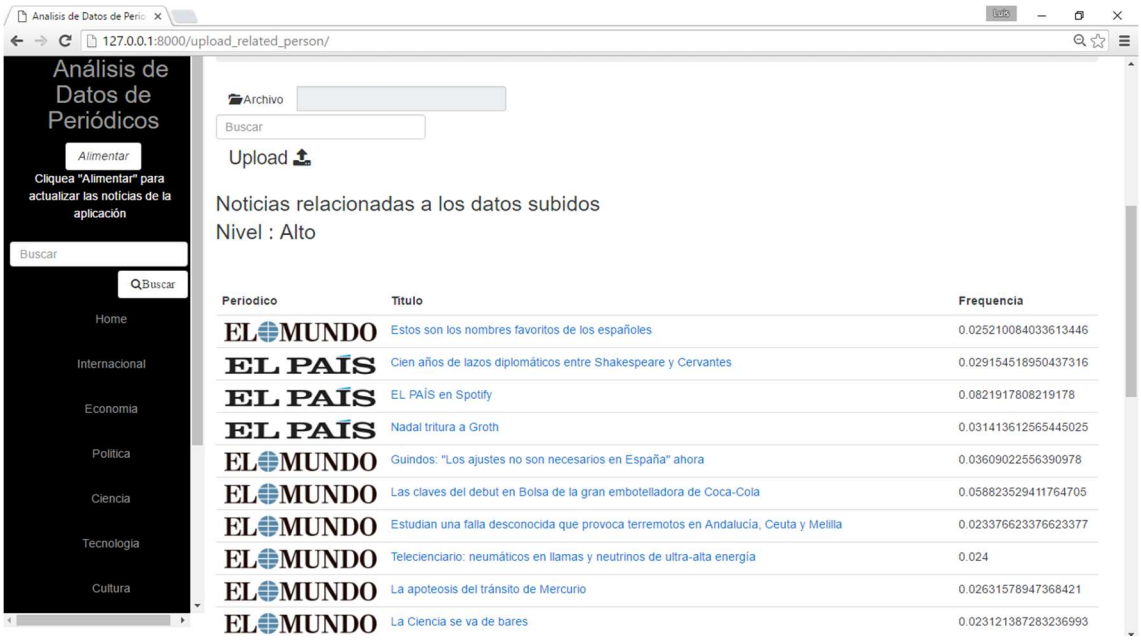


Figura U.7 Archivo de texto en formato aceptado por la aplicación

Al pulsar sobre el botón “Upload”, se realizará una búsqueda de noticias que estén relacionadas a estas palabras y pesos, y se retornará al usuario una tabla conteniendo el título de las noticias, junto a su periódico y frecuencia calculado. Cuanto mayor es la frecuencia, más relacionada es la noticia a los datos introducidos por el usuario. La frecuencia mínima definida por la aplicación es de 0.02.



U.8 Vista de resultados tras pulsar sobre el botón “Upload” y utilizando el archivo de texto de la figura U.7

Si no subes un archivo de texto, se realizará una búsqueda de noticias relacionadas a la reputación de la palabra introducida en la caja de texto.