E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Grado en Ingeniería Informática

Trabajo Fin de Grado

Iñigo Apesteguía Sanz

Carlos López Molina, Raúl Orduna Urrutia

Pamplona, 26/06/2014

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Resumen

Este proyecto trata de la construcción de un módulo de búsqueda de noticias mediante la búsqueda semántica de palabras clave. El módulo forma parte la aplicación DNow, desarrollada por Diario de Navarra, para la generación de hilos de noticias y su posterior consumo en smartphones. El módulo debe encargarse de buscar noticias en la una base de noticias nutrida de las noticias publicadas en la www.diariodenavarra.es . El alumno ha sido encargado del proceso completo de análisis de requisitos, diseño del módulo y posterior implementación y prueba del mismo.

Keywords

TextAlytics, Cyclone, búsqueda semántica, Periodismo, manejo archivos Json.

Indice

Introducción

1.1 Situación actual

Nos encontramos en una época en la que las constantes mejoras tecnológicas y sociales han abierto un gran abanico de nuevas opciones con las que llegar a la información rápidamente. Una gran influencia en el ello ha sido la expansión de internet. Por ejemplo, con las redes sociales se llegan a conocer los sucesos casi al instante de que ocurran.

Como sabemos las empresas no son ajenas a toda esta expansión y hoy en día estan integradas en todas o la mayoria de las redes sociales: twitter, Facebook, flikcr… Otro campo aprovechado por las es la expansión mundial que hay de los smartphones, a través de las aplicaciones móviles.

Esta expansión ha llegado a tal punto que ahora muchas empresas tienen su propia aplicación móvil. Ésto provoca que las empresas intenten ofertar aplicaciones innovadoras. Las empresas dedicadas a la información no se encuentran ajenas a todo esto y buscan que sus ususarios tengan accesible la información lo mas rápido posible.

Diario de Navarra ha apostado por crear una aplicación móvil que permita el acceso inmediato a una serie de noticias relevantes. La aplicación tendrá una interfaz gráfica para que el usuario tenga una lectura cómoda y que además le permita seleccionar sobre que noticia quiere recibir notificaciones si se actualiza el hilo de la misma. Este proyecto no sólo busca la comodidad del usuario sino quetambien busca desarrollar una interfaz gráfica para que el periodista cree el hilo de la noticia con comodidad.

La interfaz del periodista contará con tres columnas:una columna para la creación del hilo; otra columna en la que podrá acceder a posibles comentarios relacionados en las redes sociales (twitter, Facebook…); y otra columna el periodista encontrará una lista de noticias del Diario de Navarra que esten relacionadas semánticamente.

El trabajo desarrollado en este proyecto se centra en la búsqueda semántica de noticias relacionadas. Utilizando un servidor asíncrono codificado en Python permitimos al periodista buscar en una lista de noticias, que pueden ser actualizadas siempre que quiera el periodista.

1.2 Objetivos

El objetivo principal del Diario de Navarra es la creación de una aplicación móvil (DNow) que permita al periodista el seguimiento a tiempo real de una serie de noticias apoyado, en todo momento, por dos buscadores. Un buscador semántico en la base de noticias publicadas por el Diario de Navarra y un buscador o rastreador de redes sociales que le permitiría al usuario interaccionar con el periodista. Esta interacción estará en todo momento moderada por el periodista, es decir, no todos los posibles comentarios de los usuarios van a aparecer en el hilo de la noticia.

En cuanto a los objetivos propios de este proyecto podemos encontrar:

1. Creación de servidor asíncrono que responda las peticiones de búsqueda que puedan realizar los periodistas durante la creación y edición del hilo de la noticia.
2. Construcción de módulo que nos permita acceder a la lista de noticias publicadas en diariodenavarra.es para que se puedan añadir a la base de noticias sobre la que va a trabajar el buscador de noticias.
3. Construcción de un módulo que nos permita realizar una búsqueda semantica, de un texto definido por el periodista, en la lista de noticias guardadas en el servidor.
4. Construcción o uso de un sistema que nos permita estandarizar la presentación de las noticias en la interfaz del periodista.
5. Creación de un módulo que utilice la herramienta TextAlytics para la extracción de entidades, relaciones… de un texto. Este módulo también deberá formatear la información en un archivo Json que será utilizado por el buscador de noticias.
6. Creación de una interfaz intuitiva para el periodista que le permita actualizar la lista de noticias sobre las que trabaja el buscador y realizar búsquedas con facilidad
7. Utilización de un ranking a la hora de mostrar las noticias relacionadas dependiendo de la cercanía a la búsqueda relacionada.

1.3 Mejoras y cambios

En este apartado se van a comentar las nuevas funciones que se quieren añadir con el proyecto:

- Se introducirá un buscador semántico para la búsqueda de noticias que podrá ser utilizado en otras secciones de la empresa.
- Se añadirá un rastreador de redes sociales que podría ser una herramienta útil para ver el impacto de una noticia.
- Con la aplicación móvil se intentará acercar mas la información proporcionada al lector.

1.4 Definición de términos

- **Proyecto DNow**: proyecto original del Laboratorio de Diario de Navarra que consta de cuatro módulos generales. Una aplicación móvil, un buscador semántico, un rastreador de redes sociales y un módulo creador de la noticia.
- **Buscador en redes sociales**: Módulo que realiza la búsquedas de los términos deseados en diferentes redes sociales a elegir entre: Facebook, YouTube, Twitter o Flickr. Se encuentra fuera del alcance del proyecto aquí descrito.
- **Buscador semántico**: Módulo desarrollado en este proyecto. Realiza una serie de búsquedas semánticas sobre la base de datos de noticias publicadas en la página web de Diario de Navarra.
- **Aplicación móvil**: Aplicación desarrollada para presentar a usuarios finales los resultados de ambos buscadores.
- **Palabras clave/Keywords**: Palabras con las que se va a realizar la búsqueda en la base de noticias.
- **TextAlytics**: Conjunto de herramientas que permiten la extracción de diversa información de las noticias mediante llamadas a su API.
- **Informático**: Empleado de Diario de Navarra con conocimientos de informática suficientes que se encarga de arrancar y apagar el sistema.
- **Periodista/Cliente**: Empleado de Diario de Navarra que se encarga de la creación las noticias que van a ser cubiertas diariamente para su consumo en la aplicación móvil.
- **Python**: Lenguaje de programación con el que se ha trabajado en el proyecto a la hora de la creación del servidor y del buscador de noticias.

- **Twisted**: Framework de red codificado en Python que facilita la programación dirigida por eventos.
- **Cyclone**: Framework web para Python utilizado para la creación del servidor.
- **HTML**: Lenguaje de etiquetado que es utilizado para realizar páginas web.
- **API**: Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software.
- **Tarjetas HTML**: Concepto que refleja la presentación estandarizada que se va a utilizar para las noticias.
- **Embedly**: Conjunto de herramientas que permiten la creación de tarjetas HTML para las noticias.

2 Requisitos y arquitectura

2.1 Requisitos de desarrollo

En este apartado vamos a dedicarlo a enumerar los resquisitos de desarrollo. Desde el Diario de Navarra se nos requirieron algunas características:

- Buscador semántico: la búsqueda realizada en la base de noticias publicadas en la página web debía ser una búsqueda semántica no sintáctica.
- Visualización en página web: el buscador debe mostrarse en formato de página web accesible para la mayoría de los navegadores más utilizados.
- Servidor fácil ensamblado: el servidor en el que se monte debe tener un fácil ensamblado con el resto de módulos que componen el proyecto DNow.

2.2 Requisitos de arquitectura lógica

Los requisitos de arquitectura lógica impuestos por el Diario de Navarra son sobre el lenguaje de programación que debíamos utilizar, sobre el formato sobre en el que se debían presentar las noticias y sobre el formato del archivo o archivos donde se iba a guardar la información.

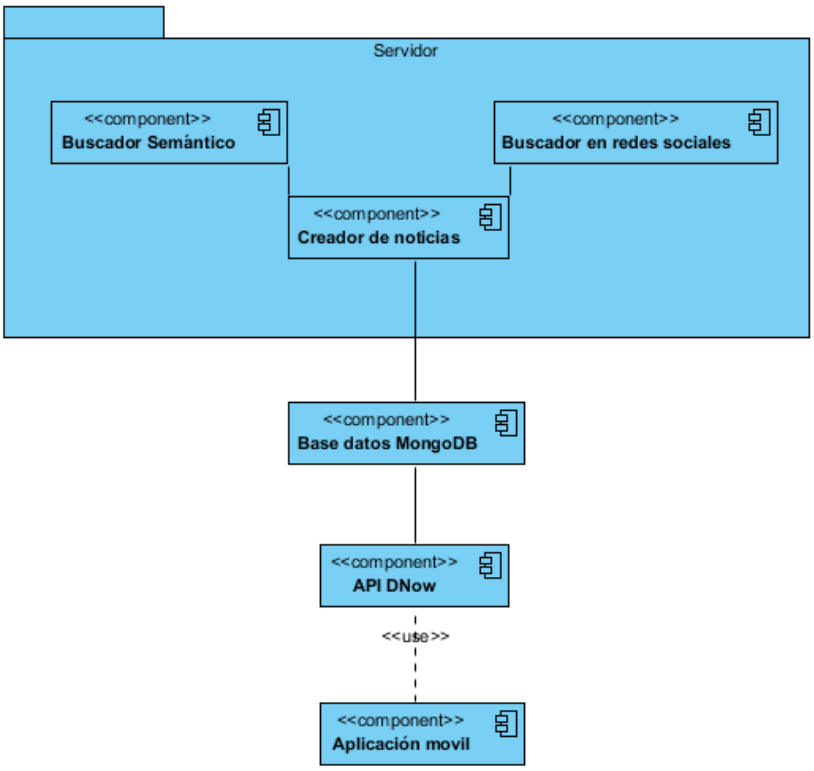

Ilustración 1 Diagrama de componentes general

En primer lugar vamos a explicar la elección de Python como lenguaje de programación. La elección de dicho lenguaje de programación fue una sugerencia realizada por un asociado al Diario de Navarra. El motivo de mayor peso para esta elección ha sido la unificación en un mismo lenguaje de todos los módulos de la aplicación general. Python nos permite que la creación de un servidor web en el que se puedan acoplar todos los módulos se realice sin mucha dificultad. Otro motivo que nos ayudó a elegir Python se trata de que la herramienta utilizada, TextAlytics, está codificada en éste lenguaje por lo que el uso de ella no reviste ningún problema adicional a la hora de la compatibilidad con el buscador.

La segunda eleccion que se hizo fue el uso de archivos en formato Json a la hora de almacenar los datos. En un principio, el motivo principal fue la intención de que la información de toda la aplicación fuese guardada en MongoDB, que se trata de un gestor de bases de datos orientadas a documentos. Al final, por plazos, no se ha conseguido la integración de todos los módulos de la aplicación general lo que ha imposibilitado la unión de la información obtenida por cada módulo en MongoDB. Una vez nos dimos cuenta que la integración de todos los módulos era imposible en el tiempo disponible continuamos utilizando Json para una posible futura integración. Otro motivo fue que la respuesta obtenida de la herramienta de text-Alytics así como la respuesta obtenida al acceder a la información de las nuevas noticias vienen en formato Json. Por último, otro motivo de peso fue que trabajar con Python utilizando archivos Json se trata de una tarea simple además de rápida.

El formato de presentación de las noticias que se decidió usar es el proporcionado por Embedly. El motivo principal fue que ya habían trabajado antes con la herramienta además de que el formateo es casi igual para las redes sociales que se utilizan en otro módulo del proyecto DNow.

2.3 Requisitos de la interfaz gráfica

Los requisitos de interfaz gráfica que se propone a continuación es en referencia al back-end, es decir, la interfaz que van a tener los periodistas.

Ilustración 2 Diseño interfaz periodista proyecto DNow

En esta interfaz se pueden diferenciar tres módulos claramente: el módulo de la izquierda en el que se tratarán las interacciones de los usuarios a través de las redes sociales; el módulo de la derecha constituido por un buscador semántico que trabaje sobre las noticias publicadas en www.diariodenavarra.es ; y por último, el módulo central en el que el periodista irá creando el hilo de la noticia pudiendo arrastrar con facilidad, con drag and drop, tanto interacciones de los usuarios como noticias relacionadas con el tema que se está tratando en la noticia o con algún tema que le parezca una aportación interesante para la noticia.

2.4 Requisitos de instalación de software

En este apartado vamos de mostrar el software necesario para la correcta creación de la herramienta presentada.

- **Linux/Ubuntu**: el sistema operativo necesario para la creación de la herramienta de búsqueda necesita que el sistema en el que se vaya a poner a funcionar sea Linux, preferiblemente Ubuntu.

- **Python 2.7**: el sistema en el que se trabaje deberá contener un programa para la ejecución de Python en su versión 2.7 para que todos los módulos utilizados dentro del buscador se puedan descargar y utilizar sin problema alguno.
- Los paquetes a descargar son los siguientes:
    - **Paquete Json**: tanto el sistema de búsqueda como la base de noticias trabajan con archivos Json, por lo que es necesaria la instalación del paquete para su manejo.
    - **Paquete Cyclone**: es un paquete necesario para poder tener un servidor en el que escuchar las peticiones o búsquedas del periodista.
    - **Demo API TextAlytics**: esta descarga es necesaria para utilizar la API porque contieneun archivo que, con alguna modificación, se utiliza a la hora de obtención de información semántica.
    - **Paquete requests**: se utiliza para la obtención de la respuesta sobre las nuevas noticias que hayan sido publicadas. Se utiliza en la actualización de la base de noticias.
    - **Paquete numpy**: paquete necesario para trabajar con matrices algunos resultados que se dan en el desarrollo.
    - **Paquete datetime**: a la hora de la actualización automática se necesitan módulos que trabajen con fechas.
    - **Paquete time**: es un paquete necesario para trabajar con el formato de fechas que maneja la base de noticias.
    - **Módulo twisted.internet.reactor**: este módulo del paquete twisted es la usada para crear la red del servidor utilizado.
    - **Módulo twisted.python.log**: este módulo del paquete twisted es utilizado para escribir diferente información en el servidor.
- **Un navegador web:** un navegador en el que se pueda acceder a la dirección del servidor en el que se encuentra el buscador.
- **Servidor apache en Ubuntu:** se necesita un servidor apache en el que alojar el servidor del buscador.

2.5 Requisitos de arquitectura física

Los requisitos de arquitectura física definidos por el Diario de Navarra para el módulo desarrollado en este proyecto son los siguientes:

- Debe existir un cliente que pueda utilizar el buscador a través de un navegador web sin necesidad de estar en la misma red que el servidor.
- El buscador se debe de nutrir de las noticias que se encuentran alojadas en el servidor donde se guardan las noticias publicadas en www.diariodenavarra.es.
- El servidor principal de este módulo será el encargado de responder a las peticiones que le realice el cliente así como de actualizar la lista de noticias que utiliza en la búsqueda obteniendo las nuevas en el servidor externo.

3 Análisis funcional

3.1 Diagrama de casos de uso

El diagrama de casos de uso que aparece a continuación podemos ver que aparecen cuatro actores: Informático, Servidor Root, Periodista y Servidor DDN. El Informático se encarga de arrancar el servidor, el Periodista es el cliente que realiza las diferentes llamadas al Servidor Root a través de los casos de uso que se aprecian; y, por último, el Servidor DDN se utiliza en el caso de uso Actualizar para obtener las nuevas noticias que se hayan publicado desde la última actualización.

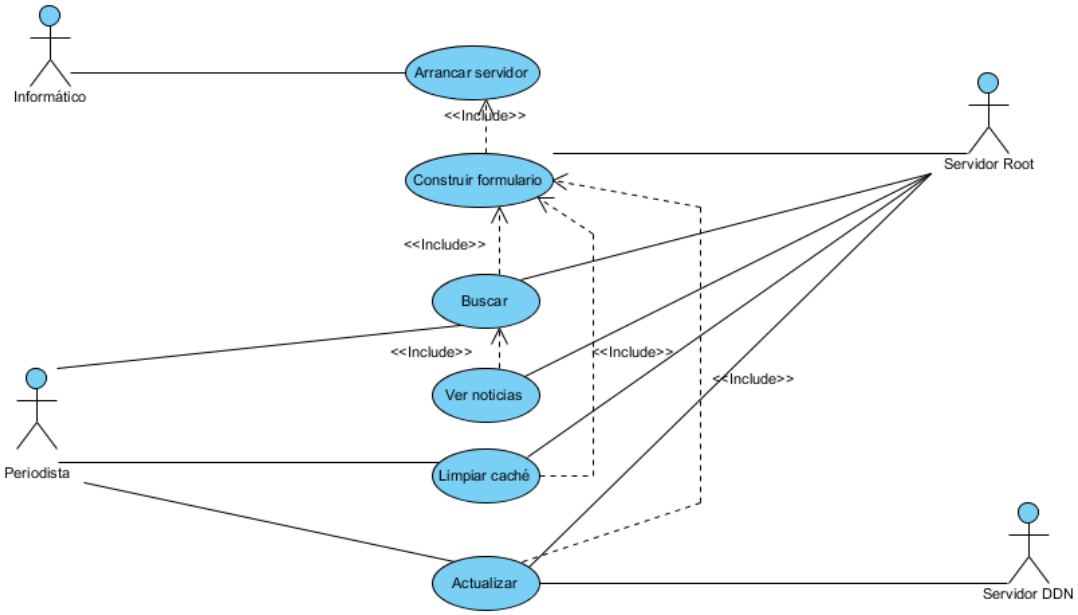

Ilustración 3 Diagrama de casos de uso

3.2 Casos de uso

1- Caso de uso Arrancar servidor

- Actores: Informático
- Descripción**:** El informático debe arrancar el servidor para que puedan servirse las diferentes búsquedas de los periodistas.
- Precondición: El sistema debe estar apagado.
- Flujo Básico**:**
    - El informático arranca el servidor
    - El servidor arranca su servicio.
    - El servidor escribe en la terminal su estado.

- El servidor se mantiene atento a escuchar peticiones.

- Flujo Alternativo:
- Postcondición: El servidor se queda iniciado y escuchando las posibles peticiones de los periodistas.
- Condiciones de error:
- Criterio de aceptación: El servidor se encuentra disponible para responder a todas las peticiones que le hagan los periodistas
- Diagrama de actividad:

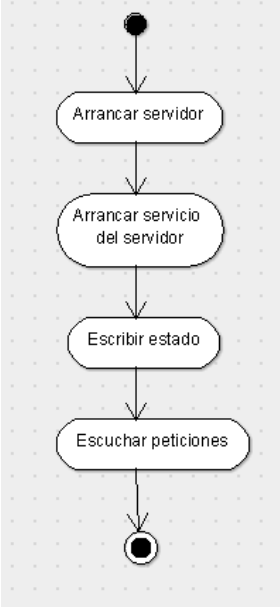

Ilustración 4 Diagrama de actividad Arrancar sistema

2- Caso de uso Construir formulario

- Actores: Servidor root
- Descripción**:** El periodista de Diario de Navarra se dispone a crear una nueva noticia y quiere incluir alguna noticia antigua para aportar nueva información por lo que requiere realizar una busqueda o una actualización y necesita un formulario.
- Precondición: Con el servidor ya iniciado, el periodista necesita realizar una peticion GET al servidor.
- Flujo Básico**:**
  - El servidor recibe peticion GET.
  - El servidor accede al directorio de plantillas y coge la correspondiente.
  - La plantilla es rellenada por el servidor.

- El servidor devuelve la plantilla rellenada.

- Flujo Alternativo:
- Postcondición: El periodista dispone de un formulario web en el que poder realizar búsquedas de noticias y actualizar la lista de noticias en las que buscar.
- Condiciones de error:
- Criterio de aceptación: El periodista obtiene un formulario en el navegador web que le permite actualizar la lista de noticias y realizar búsquedas sobre esas noticias.
- Diagrama de actividad:

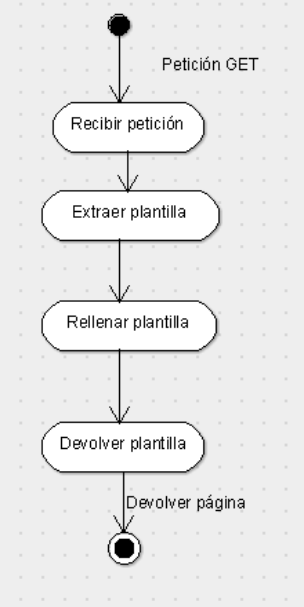

Ilustración 5 Diagrama de actividad Construir formulario

3- Caso de uso Actualizar

- Actores: Periodista, Servidor de Noticias
- Descripción**:** El periodista quiere tener actualizadas todas las noticias para tener la búsqueda más completa posible.
- Precondición: El servidor se encuentra inciado, el periodista tiene cargado el formulario de búsqueda
- Flujo Básico**:**
    - El periodista realiza una petición POST con el parámetro “actualizar“, al pulsar el botón actualizar en el formulario de actualización.
    - El sistema accede al archivo de fecha donde se encuentra la fecha de la última actualización.

- El sistema realiza una petición al servidor de noticias con unas restricciones de fechas.
- El sistema filtra las noticias que llegan y actualiza la lista de noticias sobre las que trabajar.
- El periodista recibe un mensaje que confirma la actualización.

- Flujo Alternativo:
- Postcondición: La lista de noticias sobre la que trabaja el buscador de noticias se encuentra actualizada con todas las noticias publicadas hasta el momento de la realización de la petición por parte del periodista.
- Condiciones de error:
- Criterio de aceptación: En el servidor de noticias no se encuentra ninguna noticia nueva que no se encuentre en la lista de noticias con las que se trabaja.
- Diagrama de actividad:

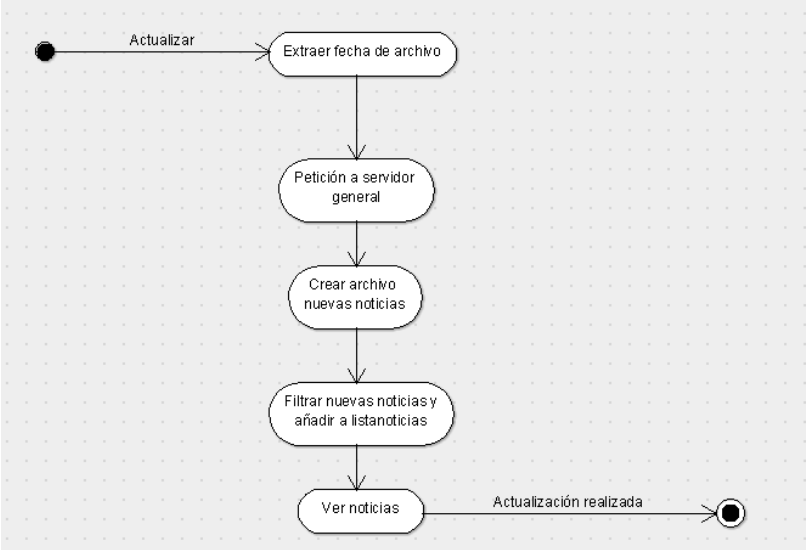

Ilustración 6 Diagrama de actividad Actualizar

4- Caso de uso Buscar

- Actores: Periodista, lista de noticias
- Descripción**:** El periodista encargado de crear la noticia quiere realizar una búsqueda para ver si hay alguna noticia interesante que pueda incluir.
- Precondición: EL sistema debe estar iniciado, el periodista debe tener cargado la pantalla con los formularios de búsqueda y de actualización.
- Flujo Básico**:**
  - El periodista pulsa el botón de buscar

- a) El sistema filtrará el contenido de la búsqueda para ver si esconde información semántica relevante.
- Con los resultados del filtro se buscarán las noticias con mayor coincidencia al contenido buscado.
- El sistema devuelve la lista de las noticias relacionadas ordenadas de mayor a menor coincidencia

- Flujo Alternativo:
  - b) El cuadro de busqueda se encuentra vacio por lo que el sistema devuelve un error.
- Postcondición: El periodista obtiene una lista de noticias relacionadas con el contenido de la búsqueda.
- Condiciones de error: El cuadro de búsqueda se encuentra vacío por lo que no se puede realizar la búsqueda y provoca un error.
- Criterio de aceptación: El periodista ha realizado una búsqueda de un contenido en las noticias sobre las que trabaja el buscador y recibe una lista de noticias relacionadas.
- Diagrama de actividad:

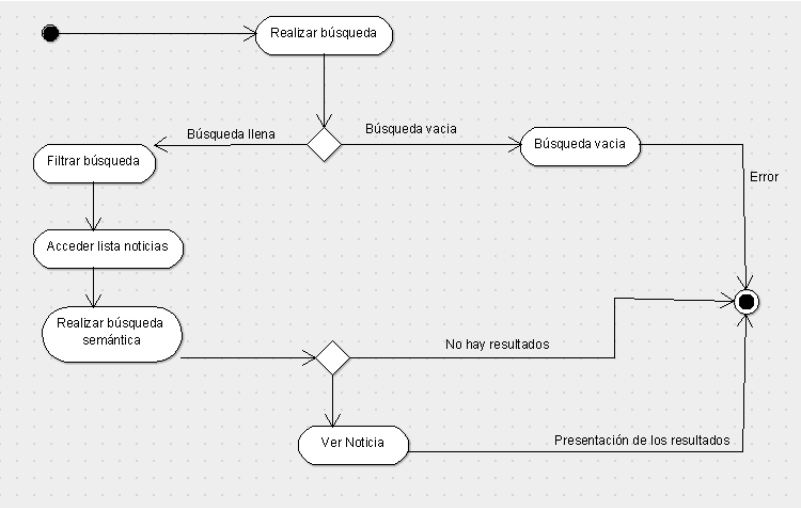

Ilustración 7 Diagrama de actividad Buscar

5- Caso de uso Ver noticias

- Actores: Periodista, lista de noticias
- Descripción: El periodista ha realizado una búsqueda obteniendose así una lista de noticias relacionadas que el sistema debe mostrar con un aspecto agradable para el usuario.

- Precondición: El periodista ha realizado una búsqueda y se ha obtenido una lista de noticias relacionadas.
- Flujo Básico**:**
    - El sistema tiene una lista de noticias resultado de una o varias búsquedas del periodista.
    - El sistema formatea la url de la noticias con Embedly.
    - En el cliente/periodista se formatea la noticia, gracias al uso de la url modificada por el servidor, con Embedly.
- Flujo Alternativo:
- Postcondición: El periodista puede advertir en su pantalla la lista de noticias relacionadas con su búsqueda o búsquedas anteriores, si la lista devuelta no llega a las treintena. Estas noticias se encuentran formateadas usando la herramienta proporcionada por Embedly .
- Condiciones de error:
- Criterio de aceptación: En la pantalla del periodista aparecen un máximo de treinta noticias relacionadas con las últimas búsquedas realizadas ordenadas de mayor a menor relación.
- Diagrama de actividad:

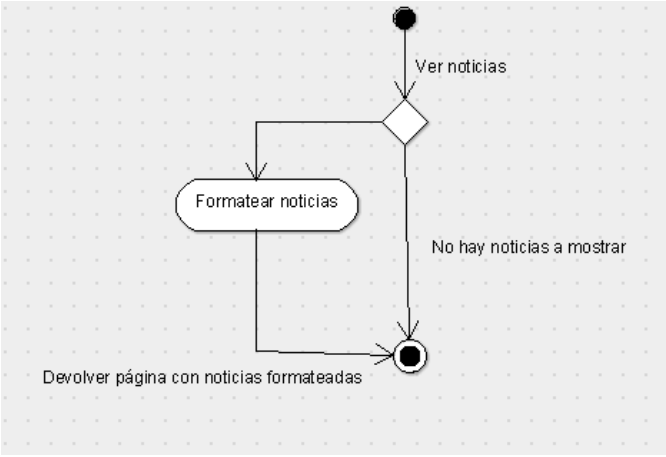

Ilustración 8 Diagrama de actividad Ver noticias

6- Caso de uso Limpiar busquedas anteriores

- Actores: Periodista
- Descripción**:** El periodista quiere limpiar los resultados de las busquedas realizadas anteriormente

- Precondición: El periodista quiere limpiar los resultados de las búsquedas realizadas anteriormente.
- Flujo Básico:
    - El periodista realiza una llamada POST con el parámetro “limpiar”
    - El sistema limpia la “caché” donde se guardaban los resultados de las búsquedas anteriores
    - El sistema devuelve un mensaje en el que se avisa de que se han borrado las búsquedas anteriores.
- Flujo Alternativo:
- Postcondición: El periodista tiene la pantalla con los formularios cargados y la caché de las búsquedas está vacía.
- Condiciones de error:
- Criterio de aceptación: La lista de noticias resultantes de las búsquedas queda vacía y en la pantalla del periodista no aparece ninguna noticia.
- Diagrama de actividad:

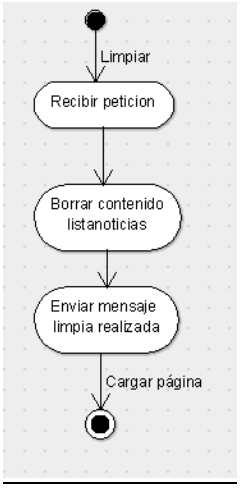

Ilustración 9 Diagrama de actividad Limpiar búsquedas

3.3 Diagrama de actividad

En este diagrama podemos observar el funcionamiento del sistema ante las diferentes peticiones de los clientes. En él vemos los cuatro tipos de posibles peticiones del cliente: la petición de servicio, la petición de búsqueda, la petición de actualización y la petición de limpieza.

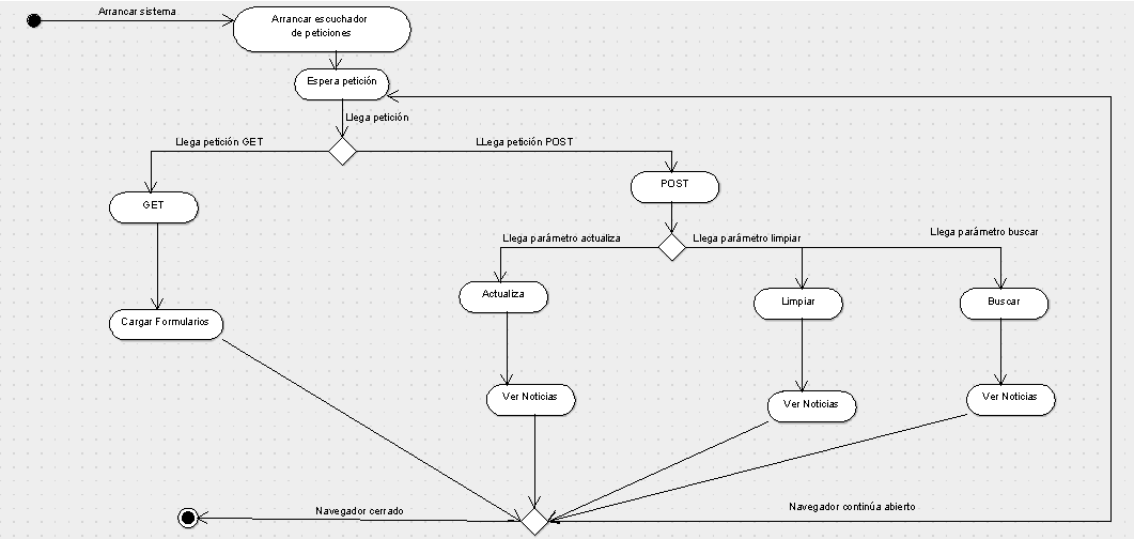

Ilustración 10 Diagrama de actividad general

3.4 Diagramas de secuencia

En el siguiente diagrama de secuencia podemos observar las interacciones o peticiones que puede hacer el Periodista al Servidor: Actualizar, Buscar o Limpiar caché.

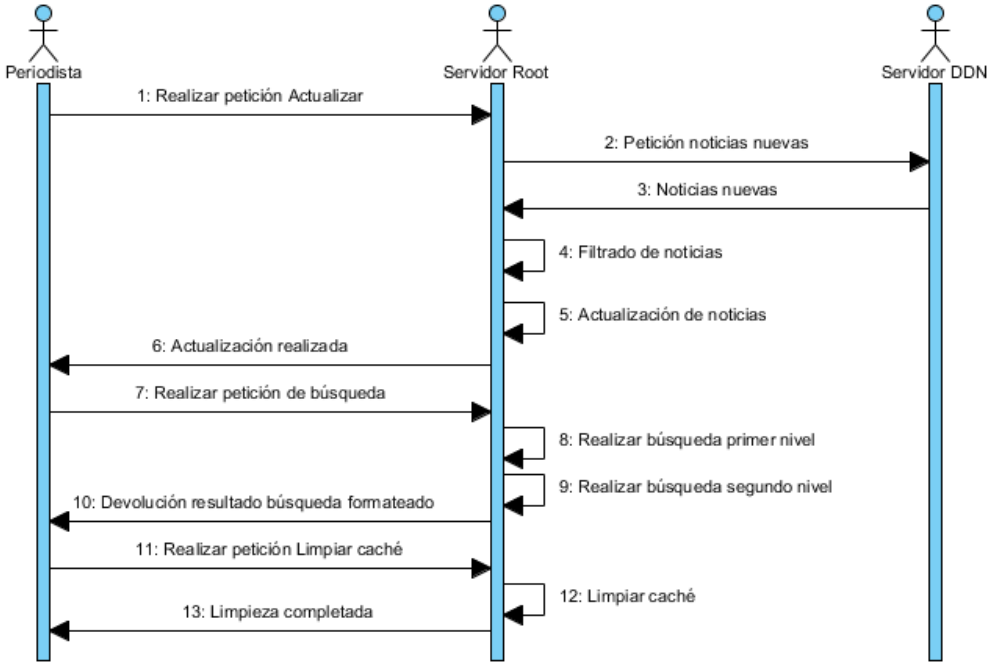

Ilustración 11 Diagrama de secuencia

4. Diseño técnico

4.1 Modelo de componentes

Modelo de componentes particular

En este modelo observamos que nuestro proyecto cuenta con dos componentes claramente diferenciables: Buscador semántico y Cliente.

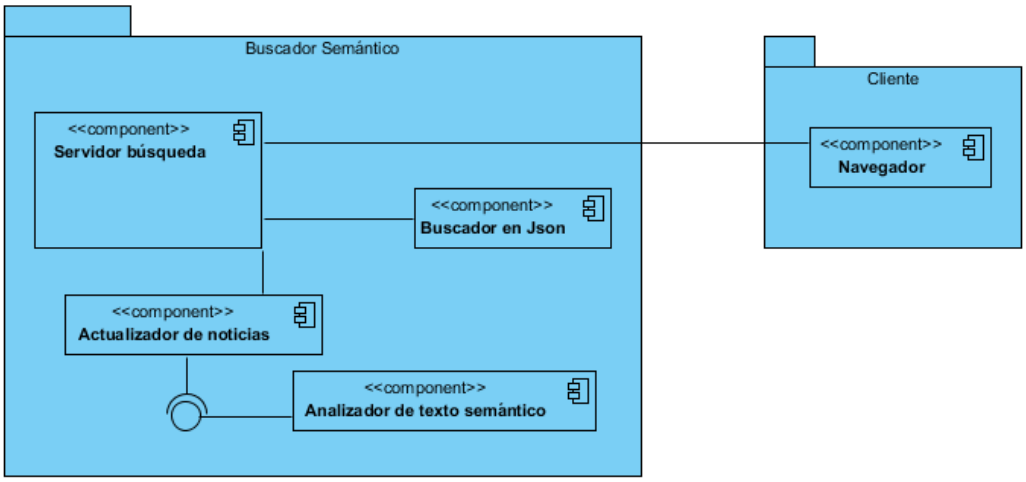

Ilustración 12 Diagrama de componentes particular

4.2 Diseño técnico cliente

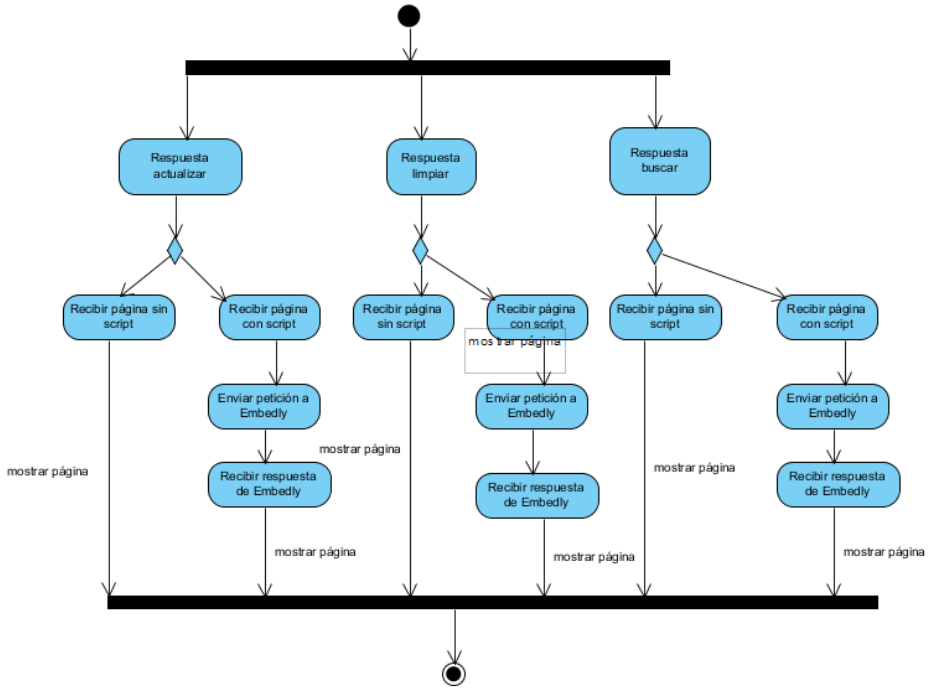

Ilustración 13 Diagrama de actividad de cliente

En este diagrama observamos las posibilidades de actuación del cliente de la aplicación: actualizar, limpiar y buscar, además de sus reacciones a las respuestas recibidas.

4.3 Diagrama de clases

En este diagrama de clases podemos observar que se usan cinco diferentes clases: TextAlytics, Creador, Buscador, MainHandler y requests. TextAlytics es utilizada para la extracción de contenido semántico de las noticias. La clase Creador se encarga de obtener del archivo de nuevas noticias, filtrar y añadir las noticias al archivo de búsqueda que luego utilizará la clase Buscador para realizar las búsquedas necesarias. La clase MainHandler es la utilizada por el servidor para atender las peticiones de los clientes. Por último, la clase requests, la única no implementada en este proyecto, se utiliza para obtener datos de diversas llamadas a páginas web.

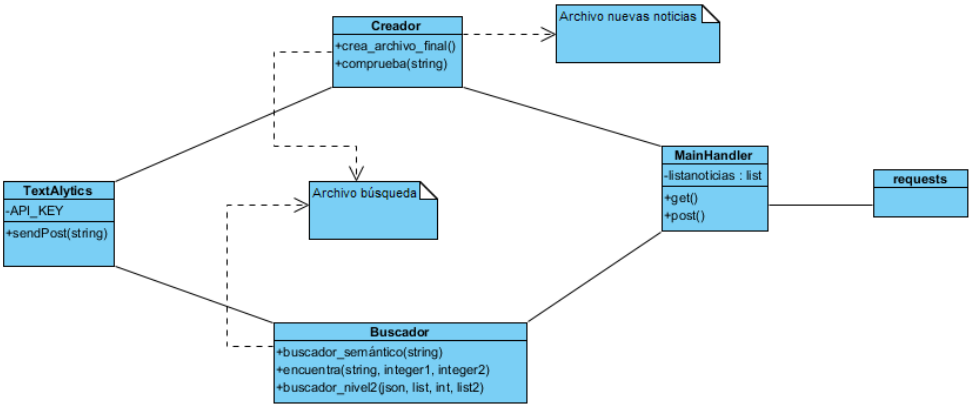

Ilustración 14 Diagrama de clases

4.4 Diseño de interfaz

En este apartado vamos a hablar sobre los requisitos funcionales que tiene el buscador de noticias. Si observamos la página detenidamente observamos que el usuario tiene tres botones que utilizar, véanse botón Actualizar (actualizará las noticias desde la última actualización realizada),botón Buscar (realizará la consulta del contenido deseado por el usuario) y el botón Limpiar búsquedas (limpia del servidor los resultados de las búsquedas anteriores).

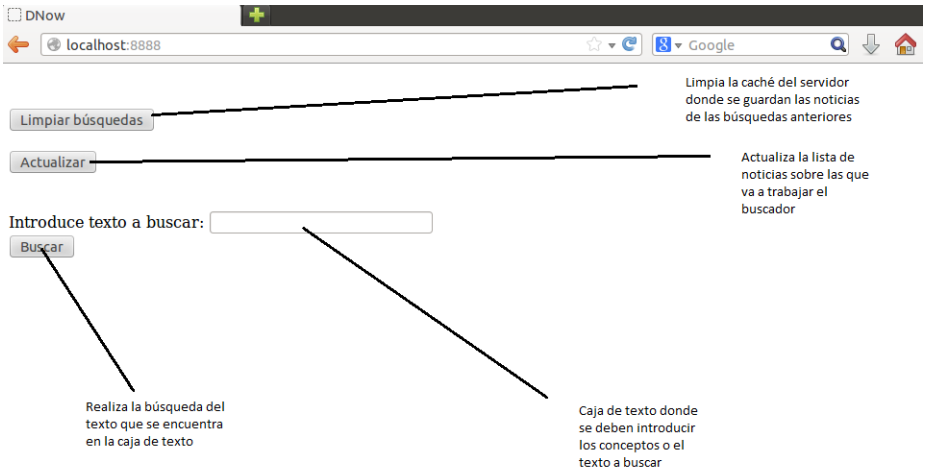

Ilustración 15 Diseño interfaz buscador semántico

Primero vamos a hablar sobre la funcionalidad del botón Actualizar. Al presionar dicho botón el servidor debe acceder al archivo que guarda la fecha de la última actualización en formato timestamp (formato en el que esta la base de noticias de donde se nutre esta herramienta). A continuación, debe transformar la fecha del instante en el que se pulsa el botón a formato timestamp. Una vez que podemos manejar las dos fechas el servidor realiza una llamada al servidor general de noticias con el siguiente formato: www.diariodenavarra.es/movil.php?query=tipo:noticia&fechaTimeStamps%20between%20*fechaUltimaActualizacion*%20AND%20*fechaActual* . Esta llamada devuelve una respuesta en formato Json con las noticias publicadas entre la fecha guardada en el archivo ultima_actualizacion y la fecha actual. Con la respuesta el servidor filtra las noticias en busca de información útil para la búsqueda semántica utilizando la herramienta de TextAlytics. Una vez filtradas las noticias son añadidas a la lista de noticias con la que trabajará el servidor en la búsqueda. También se actualiza la fecha almacenada en el archivo ultima_actualizacion para intentar evitar que las noticias se guarden repetidas. Una vez realizadas todas estas operaciones en la página del usuario aparecería un aviso anunciándole que las noticias han sido actualizadas.

En segundo lugar vamos a explicar la funcionalidad que aporta el botón Buscar. Si se presiona éste botón sin haber rellenado la caja de texto el servidor devuelve un error. Si ya se había realizado una búsqueda con anterioridad la respuesta será la misma lista de noticias pero con el error, es decir, si se le da por error no se debería realizar otra vez la búsqueda sino que la lista de noticias se mantendría inmutable. Por otro lado si hay un contenido dentro de la caja de texto el servidor se encargaría de filtrar el texto buscado con la herramienta

TextAlytics. Una vez filtrado, el servidor comienza la búsqueda de coincidencias de los filtros de búsqueda deseados por el usuario. Si la búsqueda no obtiene resultado sucederá lo mismo que en el caso que no hubiese ningún criterio de búsqueda, aparecería un error manteniéndose la pantalla igual a excepción del mensaje de error que aparecería. En cambio si la búsqueda obtiene resultados se realizará una segunda búsqueda utilizando la información almacenada sobre las noticias coincidentes. Ésta segunda búsqueda tendrá un peso menor en la ordenación de las noticias. Una vez finalizada la segunda búsqueda las noticias son ordenadas y añadidas a una "pila" de noticias relacionadas. El orden de adicción se realiza en función del número de coincidencias que se obtienen al buscar las entidades resultantes de utilizar la herramienta TextAlytics sobre el contenido de la búsqueda. La presentación de los resultados mostrará las noticias relacionadas con un máximo de 30 noticias. Si una búsqueda no obtiene 30 noticias relacionadas se mostraran las noticias de la búsqueda anterior, en el caso de que se hubiese producido alguna búsqueda con anterioridad.

Por último, vamos a explicar la funcionalidad que aporta el botón Limpiar búsquedas. Éste boton proporciona la opción al periodista de limpiar los resultados de las búsquedas que ha realizado con anterioridad. Resulta útil para facilitar la visualización de las nuevas búsquedas. Al pulsar el botón llega una petición al servidor para que borre los resultados. Después de realizar dicha operación se devuelve un mensaje al periodista, que podrá observar que ya no le aparece ninguna noticia como resultado de ninguna búsqueda anterior.

4.5 Protocolos de comunicación

En este apartado vamos a describir el protocolo de comunicación utilizado en la conexión entre el servidor y el periodista. Se trata del protocolo TCP. Dicho protocolo ha sido elegido porque garantiza la entrega de los datos en su destino sin errores y en el orden de transmisión. Al usar TCP, en el hipotético caso de que el servidor se saturase, nos permitiría poder abrir otro servidor en un puerto diferente para atender a todos los periodistas.

4.6 Formatos de intercambio

El formato de intercambio utilizado ha sido HTML. HTML es un lenguaje de marcado para la elaboración de páginas web muy extendido. Se trata de un lenguaje que basa su desarrollo en la referenciación, lo que nos permite la ejecución de elementos externos sin necesidad de que esten ubicados mediante texto. Gracias a ello podemos utilizar los scripts de Embedly que nos permiten crear las etiquetas necesarias para la correcta presentación de cada noticia.

4.7 Diagrama de despliegue

Diagrama de despliegue general

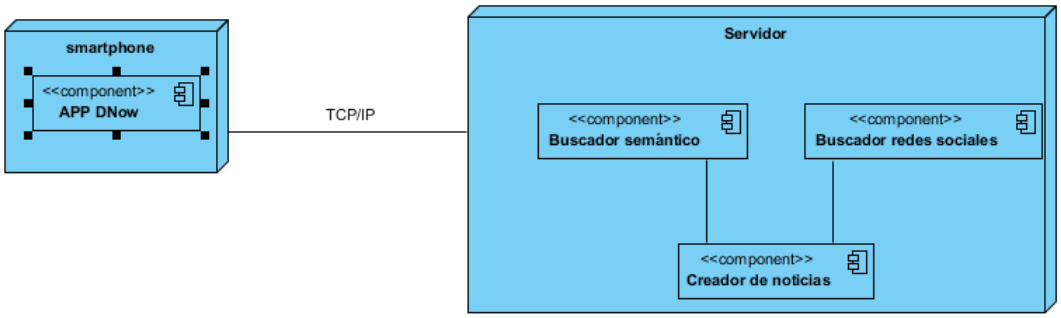

Ilustración 16 Diagrama de despliegue general

Diagrama de despliegue particular

En este diagrama podemos ver que el módulo desarrollado en este proyecto se puede dividir en tres componentes (Cliente, Servidor Proyecto, Servidor MongoDB) físicamente sin que se pierda ninguna funcionalidad. El Cliente se conectará mediante una conexión TCP-IP al Servidor Proyecto que es la parte más importante del proyecto y donde más trabajo se ha realizado. Éste, a su vez, estará conectado con el servidor MongoDB mediante Ethernet.

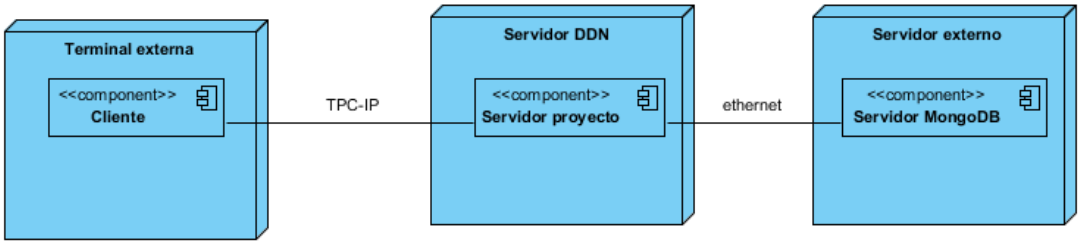

Ilustración 17 Diagrama de despliegue particular

5 Implementación y pruebas

5.1 Problemas encontrados y decisiones tomadas

En este apartado vamos a describir los problemas encontrados durante el desarrollo:

- **Conocimiento de lenguaje Python**: uno de los primeros problemas era el escaso conocimiento previo en la programación en el lenguaje Python. Éste no fue un gran problema, ya que, se trata de un lenguaje que no tiene grandes problemas para ser entendido.
- **Instalación de Ubuntu**: el primer problema que se nos planteó fue la instalación de Ubuntu en el ordenador de trabajo. El problema fue que dicho ordenador se encuentra con el sistema operativo de Windows 8, el cual debido a su nueva configuración de arranque hace casi imposible la compatibilidad de ambos sistemas operativos. La solución que decidimos tomar es la instalación de una máquina virtual que nos permitiese trabajar con Ubuntu sin problemas.
- **Formato de noticias en la interfaz gráfica**: el formato de presentación al periodista debía ser agradable a la vez de claro y homogéneo para todas las noticias que se tienen guardadas. El problema es que la base de noticias con las que se trabaja son diferentes, por ejemplo, algunas tienen imagen… Para solucionar este problema se echa mano de la funcionalidad que nos da Embedly que proporciona un formateo homogéneo a todas las noticias.
- **Formato de texto de las noticias**: el formato en el que se obtienen las nuevas noticias es Json, aunque en alguno de los campos podíamos encontrar dobles comillas que a la hora de extraer los datos nos podían llevar a obtener fallos en el archivo Json final. Este problema se decidió solucionar con la creación de una función que suprimía las dobles comillas en el texto.
- **Embedly:** la decisión de formatear las noticias con Embedly nos solucionó el problema del homogeneo formateo pero derivó en otro problema: el uso de Embedly. Embedly se trata de una empresa que provee métodos sencillos y cómodos para incrustar contenido de todo tipo (videos, noticias, imágenes, urls..) en las páginas web. Aun tratándose de unos métodos sencillos, el problema que nos apareció fue cómo añadir la línea de javascript a cada noticia para que se mostrase correctamente. Al final, el problema se solucionó añadiendo, a cada noticia resultante de una búsqueda, la línea de javascript. Con ello se formateará cada noticia por separado previniendo que si el formateo de una noticia fallase el resto de noticias no se vieran afectadas.

- **Tecnología del servidor:** uno de los mayores problemas, sino el mayor, que nos hemos encontrado en la realización de este proyecto ha sido la elección del servidor que se iba a utilizar. A la hora de decidir que tipo de servidor se iba a utilizar se tomó una decisión en conjunto con los desarrolladores de los otros módulos de DNow con la intención de acoplarlos más adelante. Con ello se tomó la decisión de que debía ser un servidor asíncrono, para poder atender diferentes peticiones además de permitir realizar algunas actividades en segundo plano. En un primer lugar nos decidimos por un servidor de Python básico para desarrollar el proyecto pero pasado un tiempo vimos que no cubría nuestras necesidades. Después de investigar otros servidores codificados en Python nos decidimos por la utilización de Cyclone un servidor que cumplía nuestras necesidades completamente.
- **Selección de herramienta semántica:** al comienzo del proyecto se debía decidir que herramienta se iba a utilizar para la extracción de los conceptos en las noticias. Esta decisión iba a marcar el ajuste que iba a hacer la búsqueda semántica. Después de buscar nos decidimos en un principio por la herramienta proporcionada por OpenCalais, que funciona muy bien en inglés y francés pero que en castellano todavía siguen en desarrollo. Al ver que su funcionamiento en castellano no era del todo bueno seguimos buscando y finalmente utilizamos la herramienta TextAlytics. Ésta herramienta esta desarrollada perfectamente en castellano y aporta mucha más información, lo que nos permite ajustar más la búsqueda semántica. El único problema de la elección de esta aplicación es que al utilizar la version gratuita tiene un límite de llamadas mensual.

5.2 Pruebas realizadas

Pruebas de carga con ApacheBench:

Las pruebas de carga del servidor se han realizado con ApacheBench herramienta de la fundación Apache. Al tratarse de una herramienta gratuita provoca que no sea necesaria ninguna inversión y en Ubuntu se puede instalar con una simple instrucción:

```
sudo apt-get install apache2-utils
```

Una vez instalado hemos realizado pruebas con diferentes cargas al servidor Cyclone que hemos utilizado utilizando siguiente la instrucción obteniendo muy buenos resultados:

```
ab -n num_llamadas -c 100 -g grafica.data http://localhost:8888/
```

1000 llamadas al servidor:

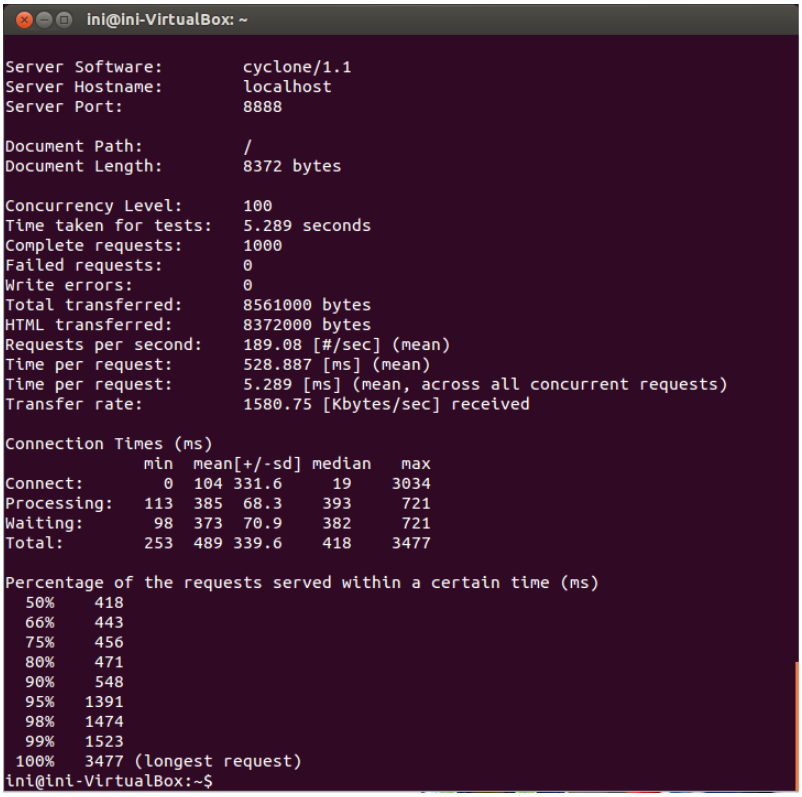

Ilustración 18 Prueba carga 1000 peticiones

5000 llamadas al servidor:

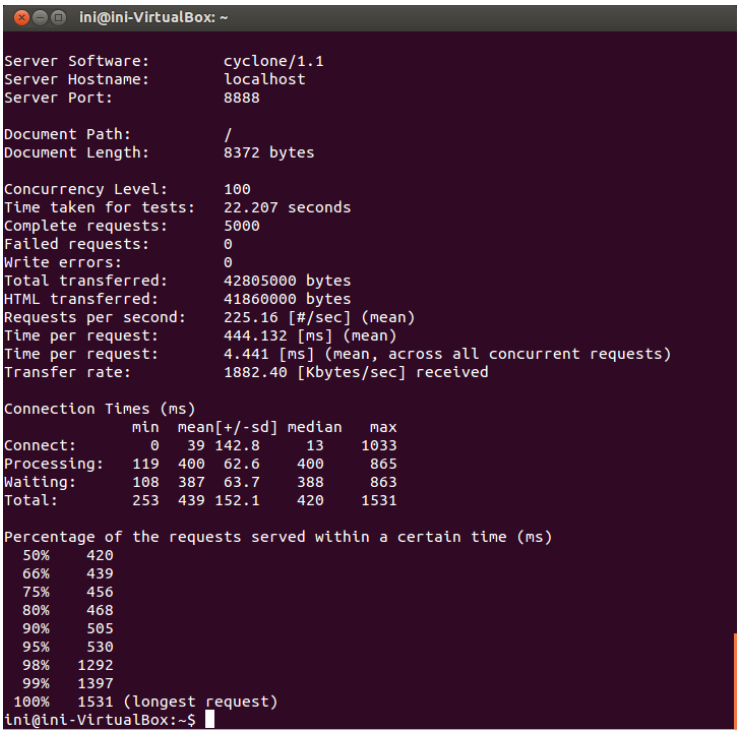

Ilustración 19 Prueba carga 5000 peticiones

10000 llamadas al servidor:

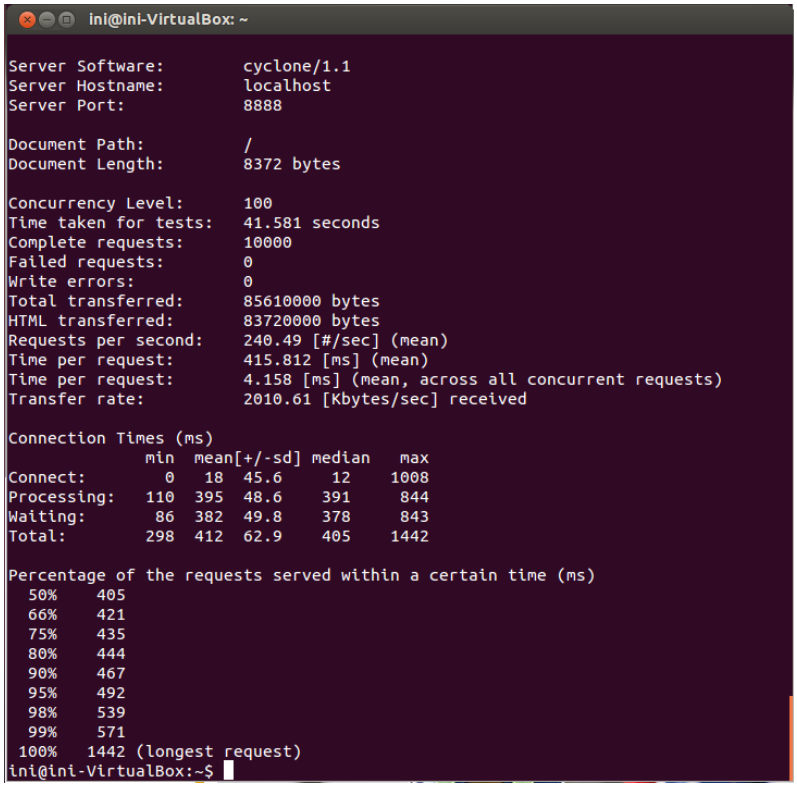

Ilustración 20 Prueba carga 10000 peticiones

20000 llamadas al servidor:

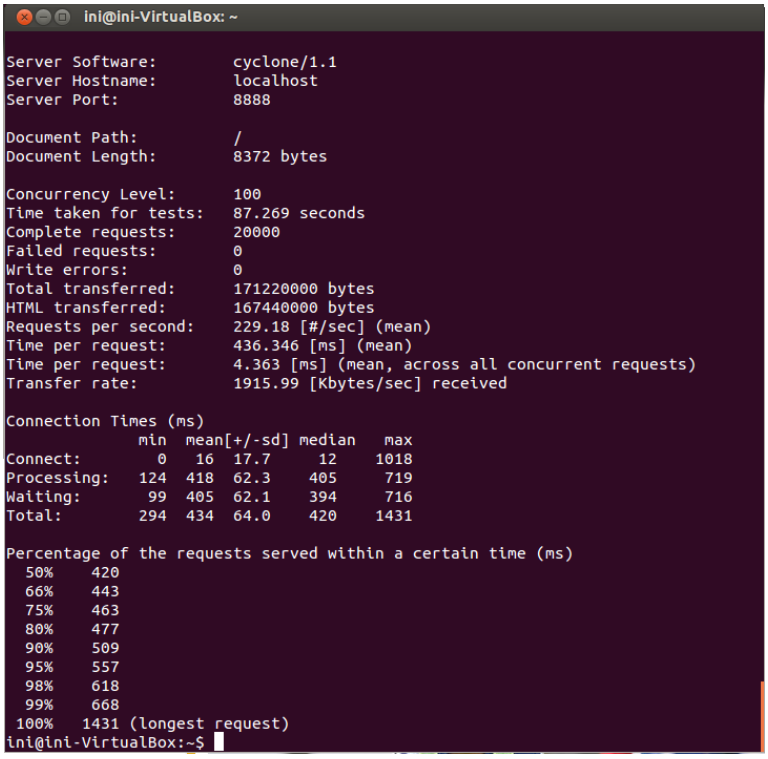

Ilustración 21 Prueba carga 20000 peticiones

50000 llamadas al servidor:

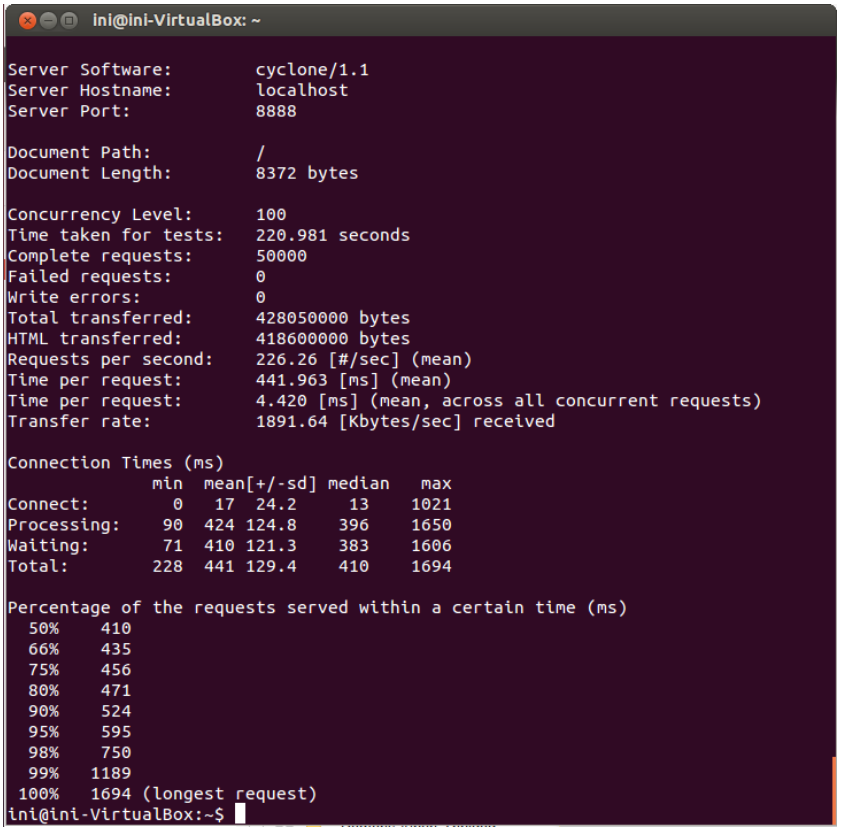

Ilustración 22 Prueba carga 50000 peticiones

Si observamos bien las capturas vemos que el tiempo que se tarda en realizar la prueba no aumenta proporcionalmente al número de llamadas, sino que el tiempo necesitado va aumentando con mas rapidez. Si continuamos observando podemos obervar que el tiempo medio de procesamiento y de espera se mantienen en todas las pruebas alrededor de los 400 milisegundos. Otro aspecto que llama la atención es que en la primera prueba el tiempo medio por petición es de 5289 ms mientras que en el resto se encuentran todos alrededor de los 4200 ms. Por último vemos que el servidor soporta hasta 50000 llamadas proporcionando servicio sin interrupción. Con estos resultados podemos afirmar que el servicio proporcionado al periodista que se encargue de la redacción de la noticia podrá responder sin problemas sus peticiones de búsqueda.

6 Guía de instalación y operación

6.1 Guía de instalación

a. Instalación necesaria para uso del servidor

Para la correcta funcionalidad del servidor es necesario trabajar en plataforma Linux, Ubuntu si es posible, además de la instalación de una serie de paquetes que se exponen a continuación:

- **Paquete Json**: el paquete Json viene instalado de serie en Ubuntu por lo que no es necesaria su instalación para el uso del buscador.
- **Paquete Cyclone**: para la instalación del paquete Cycloneademás de la instalación del paquete twisted, descrito más abajo, se deben realizar los siguientes pasos:
  - Primero debemos descargarnos el paquete desde la dirección http://pypi.python.org/pypi/cyclone
  - Seguidamente debemos escribir en la línea de comandos las siguientes instrucciones, sustituyendo $VERSION por la versión del paquete que nos hayamos descargado:
    - `tar zxvf cyclone-$VERSION.tar.gz`
    - `cd cyclone-$VERSION`
    - `sudo python setup.py install`
- **Paquete requests**: si se quiere instalar el paquete requests es necesaria la instalación previa del paquete pip. Una vez se tenga instalado el paquete pip se debe lanzar la siguiente instrucción desde la línea de comandos:
  - `pip install requests`
- **Paquete pip**: la instalación del paquete pip es necesaria por el uso del mismo en la instalación de alguno de los paquetes. La instalación de pip se puede realizar siguiendo los siguientes pasos:
  - Si escribimos las siguientes instrucciones en la línea de comandos de nuestro Ubuntu instalaremos el paquete pip:
    - `sudo apt-get install python-pip python-dev build-essential`
    - `sudo pip install --upgrade pip`
- **Paquete numpy**: para la instalación de este paquete es necesaria la instalación previa del paquete pip. Si ya tenemos descargado el paquete pip la única instrucción necesaria para instalar el paquete es la siguiente:
  - `sudo pip install numpy`

- **Paquete datetime**: se trata de un paquete instalado por defecto dentro de Ubuntu por lo que no es necesaria ninguna operación para su uso.
- **Paquete time**: se trata de un paquete instalado por defecto dentro de Ubuntu por lo que no es necesaria ninguna operación para su uso.
- **Paquete twisted.internet.reactor y Paquete twisted.python.log**: para la instalación de estos dos paquetes es necesaria únicamente la instalación del paquete twisted, ya que se trata de dos módulos integrados en ese paquete. Para la instalación del paquete twisted es necesario:
  - Descargar el paquete que aparece en esta dirección: http://packages.ubuntu.com/precise/python-twisted
  - Una vez descargado se debe extraer su contenido en nuestro directorio de trabajo.
  - Para finalizar la instalación dentro de la línea de comandos debemos posicionarnos en la carpeta donde están los archivos extraídos y lanzar la siguiente instrucción:
    - `sudo python setup.py install`

b. Instalación necesaria para la utilización del buscador

Para utilizar el buscador únicamente es necesario tener instalado un navegador web.

6.2 Manual de usuario

a. Manual de usuario para arrancar el servidor

- Paso 0: antes de intentar iniciar el operario debe asegurarse de tener todos los paquetes necesarios para el correcto funcionamiento del servidor.
- Paso 1: en la terminal debemos colocarnos en el directorio de trabajo donde tengamos los archivos guardados.
- Paso 2: si ya estamos situados en el directorio debemos ejecutar la siguiente instrucción:

```
python server.py
```

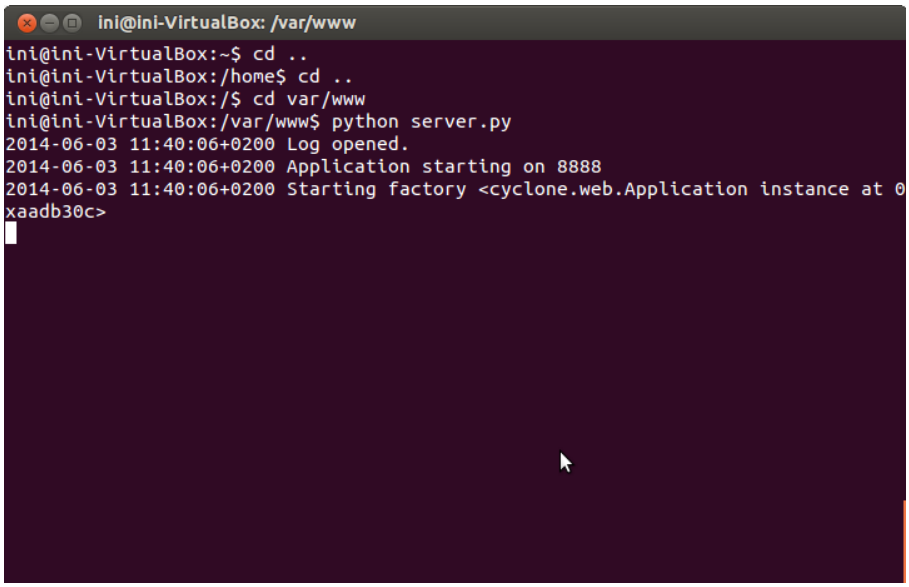

Ilustración 23 Estado servidor arrancado

b. Manual de usuario para utilizar el buscador

- Paso 1: Acceder a la dirección http://localhost:8888/

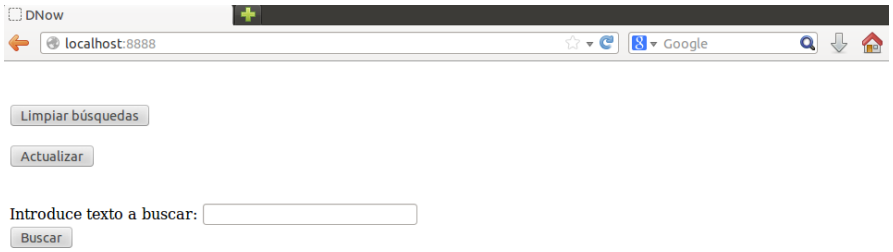

Ilustración 24 Pantalla inicial cliente

- Paso 2: Actualizar la lista de noticias

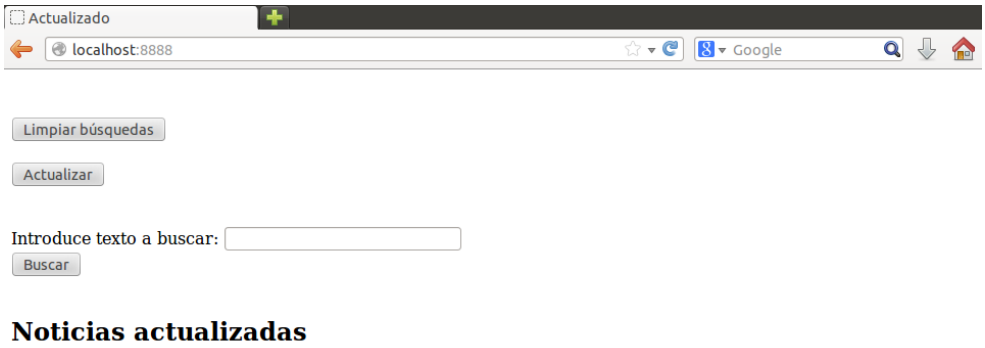

Ilustración 25 Pantalla noticias actualizadas

- Paso 3: Realizar una búsqueda de la temática deseada en la página

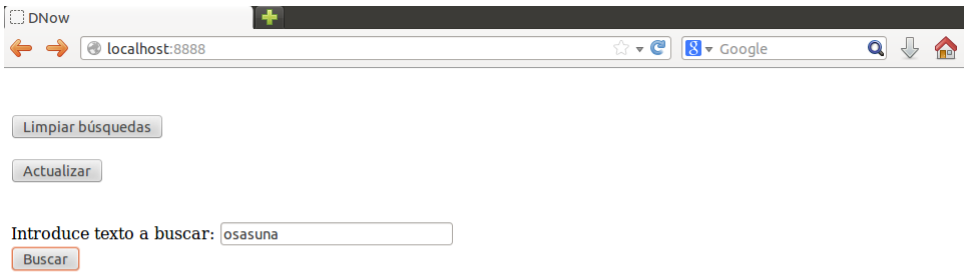

Ilustración 26 Pantalla introducción texto búsqueda

- Paso 4: En la pantalla aparecerán un máximo de 30 noticias. Se puede realizar otra búsqueda y en el caso de que los resultados no lleguen a 30 noticias el buscador guarda una lista con todas las noticias resultantes de las anteriores búsquedas y rellenará la lista con los resultados de las búsquedas más recientes.

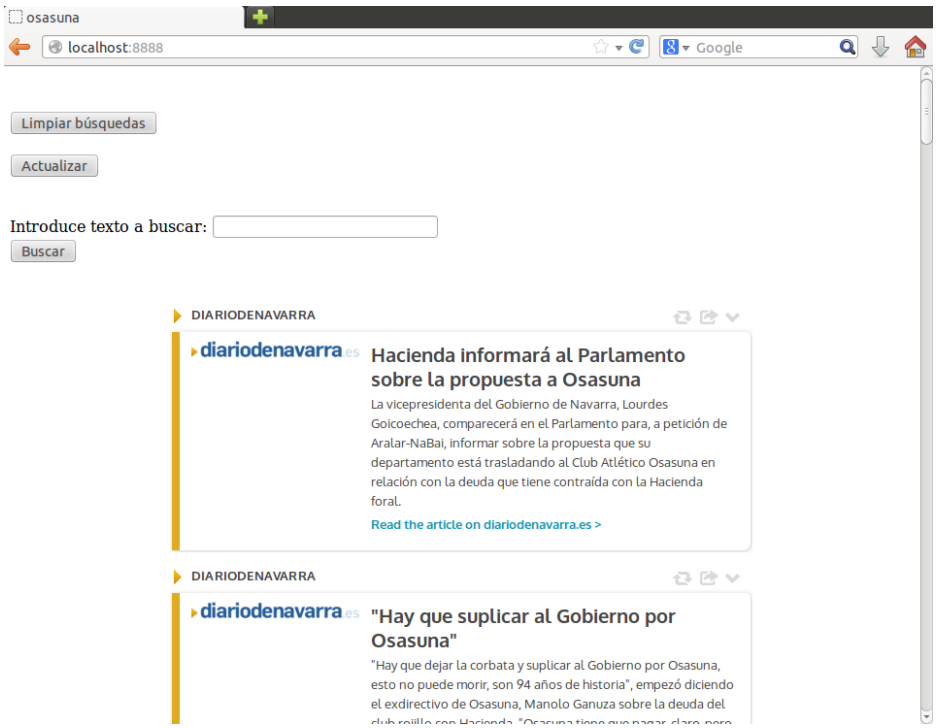

Ilustración 27 Resultado búsqueda

- Paso 5: Limpiar caché de búsquedas anteriores.

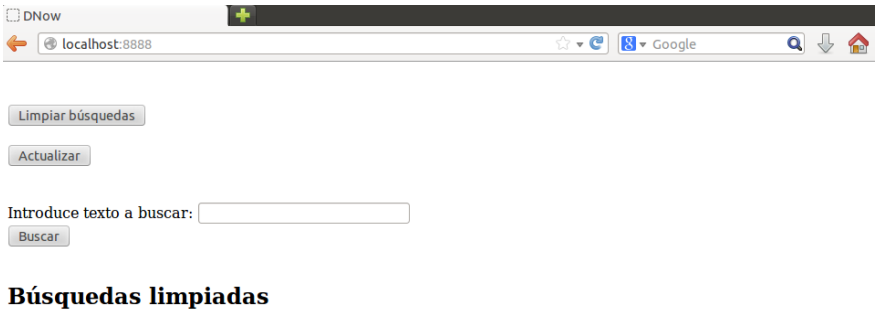

Ilustración 28 Pantalla búsquedas limpiadas

7 Resumen, conclusiones y bibliografía

7.1 Resumen

En resumen, el proyecto realizado se trata de un módulo de una aplicación móvil que busca desarrollar Diario de Navarra. El módulo debe dar un soporte al periodista para que pueda realizar búsquedas semánticas en la hemeroteca de noticias que hayan sido publicadas en la página web de Diario de Navarra. El módulo consta de una interfaz en la que el periodista tiene tres opciones: actualizar la lista de noticias sobre las que trabajar, limpiar búsquedas realizadas con anterioridad; y, por último, la opción de realizar una búsqueda semántica. La búsqueda se puede realizar tanto de una palabra o concepto como de una frase, el resultado será una lista de noticias relacionadas con los términos buscados. Las noticias resultantes serán formateadas con Embedly para mostrar homogéneamente todas las noticias en un formato agradable.

7.2 Conclusiones

Tras desarrollar el proyecto se puede llegar a las siguientes conclusiones:

- Python es un lenguaje bastante sencillo de utilizar en el que la importación de su diversa cantidad de módulos y paquetes le aporta un valor añadido. En particular me gusta que no haya tipos en las variables lo que provoca que se puedan tratar prácticamente igual.
- Embedly es un conjunto de herramientas de formateo bastante interesante si se trabaja con diferentes fuentes de información y se quiere estandarizar la presentación.
- TextAlytics nos proporciona una herramienta bastante completa para extraer el significado o palabras, entidades… clave dentro de un texto.
- Cyclone se trata de un framework interesante a la hora de trabajar asíncronamente en Python por la facilidad de compresión que tienen. Se trata de un servidor recomendable para la iniciación en el uso de servidores en Python.

7.3 Bibliografía

- Manual Pyhton en Español. Disponible en: http://docs.python.org.ar/tutorial/2/contenido.html
- API herramienta TextAlytics Disponible en: http://textalytics.com/inicio
- Florian Bauer, Martin Kaltenböck, Linked Open Data - The Essentials
- Steven Bird, Ewan Klein, Edward Loper, Natural Language Processing with Python
- Toby Segaran,Programming Collective Intelligence
- Wes McKinney, Python for Data Analysis
- Cyclone framework de servidor web asíncrono. Disponible en: http://cyclone.io/
- Embedly herramienta formateo de noticias. Disponible en: http://www.embed.ly