



Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación

Máster en Ingeniería del Software, Métodos Formales y Sistemas de la Información

# GOV TEXT Analytics System

## Sistema Gubernamental para el Análisis de Textos

Autor: Humberto Delgado Lázaro

Director: Dr. Joan Fons i Cors

Singapur, Julio 2015

# Resumen

La propuesta que se ha desarrollado en esta tesis trata sobre la creación de un producto que permite capturar, almacenar, procesar y analizar contenidos no estructurados procedentes de diferentes canales, tales como redes sociales, foros de debate, webs corporativas o correos electrónico. El objetivo de esta propuesta es dotar a los usuarios de las diferentes agencias gubernamentales del gobierno de Singapur de un sistema que les permita descubrir patrones de comportamiento, tendencias, detectar desviaciones y monitorizar discusiones.

Se presenta el entorno del proyecto, la metodología empleada además de otros factores de interés de la misma. Posteriormente se presenta un amplio estudio de la plataforma, detallando su arquitectura, virtudes, posibilidades y carencias.

El proyecto incluye la programación de una solución real, la cual ha sido desarrollada durante varios meses, utilizando la metodología scrum. Se presenta la planificación en iteraciones, artefactos generados y manual de usuario que ilustra su uso.

# Palabras clave

Análisis de textos, Scrum, Servicios Web, NLP, S2T, Lexalytics

# Abstract

The proposal has been developed in this thesis is about creating a product that allows users from the different Agencies to capture, store, process and analyze unstructured textual contents collected from both the Social Media space as well as their service contact points (CRM systems). The objective of this proposal is to provide users of the different government agencies in Singapore government a system that allows them to discover patterns, trends, detect deviations and monitor discussions.

The project environment is presented, as well as methodology and other factors of interest. Subsequently, a comprehensive study of the platform is presented, detailing its architecture, strengths, opportunities and weaknesses.

The project includes programming a real solution, which has been developed over several months using the scrum methodology.

# Keywords

Natural Language Processing, Scrum, Web Services, NLP, S2T, Lexalytics

A mi madre Encarna

# Agradecimientos

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que han colaborado de manera directa o indirecta en este proyecto.

En primer lugar, a mis compañeros de trabajo por el apoyo mostrado en todo momento.

I especially want to thank Lizanne Teo for her support, encouragement and patience.

Quiero mostrar mi gratitud a mi director Dr. Joan Fons i Cors, de la Universidad Politécnica de Valencia. Sus continuos ánimos y consejos han sido un gran impulso para esta tesis.

Por último, quiero dar las gracias a mi familia, por confiar en mí y apoyarme en todo momento. Ellos me han inculcado los valores y principios que han servido de cimiento, para que hoy se vea realizado uno de mis objetivos.

# Acrónimos

| | |
|---|---|
| S2T | Simulation Software & Technology |
| OSINT | Open-source intelligence |
| Gov TA | Government Text Analytics |
| HCI | Human-Computer Interaction |
| MVC | Model View Controller |
| SPA | Single-Page Application |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation |
| SVG | Scalable Vector Graphics |
| NLP | Natural Language Processing |
| AJAX | Asynchronous JavaScript And XML |
| XML | Extensible Markup Language |
| OSC | National Intelligence Open Source Centre |
| NOISC | National Open Source Intelligence Centre |
| MND | Ministry of National Development |
| NEA | National Environment Agency |
| IDA | Infocomm Development Authority |
| DMZ | Demilitarized Zone |
| VoIP | Voice over IP |
| SPF | Singapore Police Force |

# Índice general

# Índice de figuras

# Índice de tablas

# 1

# Introducción

## 1.1. Motivación

En el mes abril del año 2013, empecé a trabajar en la empresa "Simulation Software & Technology" (S2T) en Singapur. Una empresa con 10 años de experiencia en los cuales ha proporcionado servicios y productos a decenas de clientes en Singapur y en el extranjero, en áreas como *open source intelligence* (OSINT), desarrollo de software a medida y gestión del conocimiento.

Participar en el desarrollo de un proyecto empresarial, aprender el procedimiento habitual de la realización de un proyecto, participar en sus diferentes fases, plazos, y dinámica general de trabajo fue una de las principales razones por las que decide realizar la tesis en S2T.

Tras investigaciones preliminares de mercado, se ha identificado que cada día un mayor número de empresas, organizaciones gubernamentales y militares tiene un mayor interés en OSINT. Debido a la previa experiencia de S2T en este mercado y al aumento de la demanda este tipo de herramientas se consideró que crear un producto de calidad, siguiendo buenas prácticas y estándares podría ampliar enormemente el mercado de clientes.

Con esta nueva herramienta se pretender dotar a los usuarios de los medios suficientes para conocer el sentimiento de la opinión pública sobre temas de actualidad y conocer cuáles son aquellos problemas que preocupan a los ciudadanos en el día a día.

## 1.2. Objetivos

El objetivo principal es dotar a los usuarios de las diferentes agencias gubernamentales del gobierno de Singapur de un sistema que les permita capturar, almacenar, procesar y analizar contenidos no estructurados procedentes de diferentes canales, tales como redes sociales, foros de debate, webs corporativas o correos electrónico. Además de la ganancia de productividad de los usuarios en el análisis de los datos, GOV TA permite a los usuarios descubrir patrones ocultos, las tendencias de la opinión pública, referencias ocultas en los textos y revelar asuntos de interés.

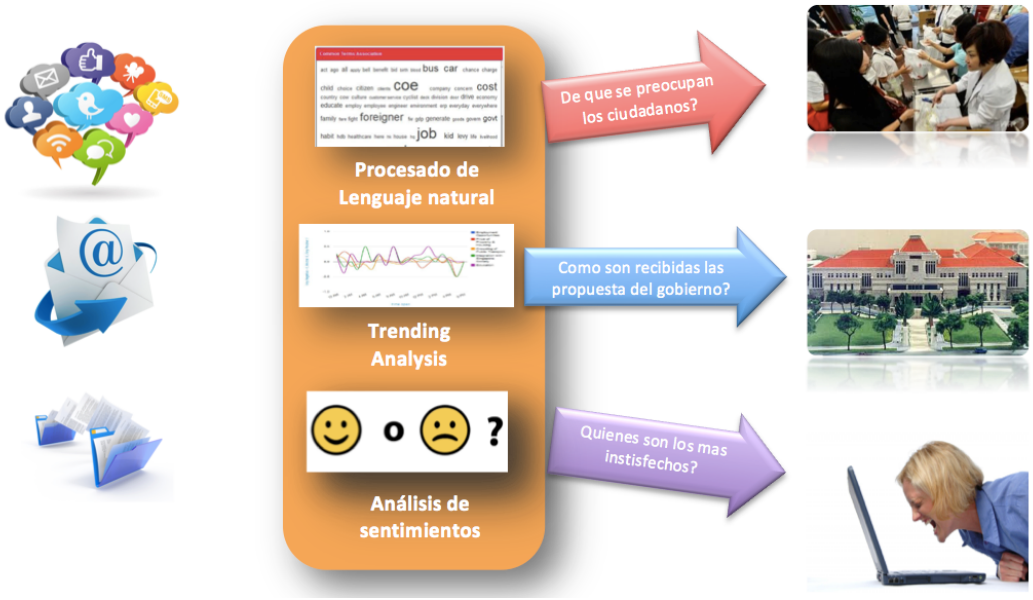



Figura 1: Gov TA

A continuación se muestra un resumen de los principales objetivos del proyecto

- **Gráficos**: Proporcionará una interfaz que permite a los usuarios obtener información sobre los datos. Los usuarios son capaces de crear gráficos de barras, *word cloud* o series cronológicas usando *keywords*, sinónimos, *themes* o *topics*.

- **Alertas**: El usuario es capaz de crear suscripciones basadas en *topics*, *keywords* o *themes*, que notificaran al usuario por correo electrónico cuando el sistema encuentre actualizaciones en alguno de sus temas de interés.

- **Análisis de sentimientos de documentos:** El análisis automatizado sentimientos pretende imitar la capacidad humana de analizar textos.

- **Clasificación de documentos:** El sistema proporciona la capacidad de clasificar textos en diferentes *topics* o temas.

- **Informes:** El sistema permite la generación de informes de los temas de interés en las últimas 48 horas, opinión de los ciudadanos sobre diferentes asuntos o asuntos que están aumentando interés en las últimas semanas.

- **CRM:** Los usuarios podrán utilizar el sistema para realizar análisis con las diferentes fuentes de datos específicas para cada organismo. De esta manera, los usuarios también serán capaces de comparar visualizaciones procedentes de distintos sistemas o de diferentes medios de comunicación social.

- ***Topic classification using Black-box machine learning model*:** El sistema permita la clasificación automática de textos basada en diferentes modelos como "Naïve Bayes Classifier", "Decision Tree Classifier" o "MaxEnt Classifier".

- ***Association rules*:** También conocido como reglas de asociación, permiten describir hechos que ocurren en un conjunto común de datos. El propósito de aplicar el algoritmo de a priori [1] en el sistema es descubrir relaciones en los datos del sistema.

Adicionalmente de estos objetivos, también encontramos otros objetivos personales como por ejemplo poner en práctica los conocimientos adquiridos en el máster, preparar un entorno de trabajo para un proyecto de desarrollo de varios meses de duración e idear soluciones para los problemas que vayan surgiendo durante el proyecto.

## 1.3. Beneficios esperados

Además de las funcionalidades que el sistema va a aportar a los usuarios en su día a día, existe un beneficio general para las distintas organizaciones, el cual es aportar una plataforma única y que a su vez está adaptada a las distintas necesidades de las organizaciones.

Una plataforma única de trabajo para las organizaciones, no tan solo se cubren las necesidades de los usuarios, sino que también se favorece el intercambio de información entre ellas.

Desde el punto de vista económico del proyecto, crear una plataforma única ha aportado un beneficio en la reducción de los costes, ya que en lugar de crear múltiples proyectos para las distintas organizaciones, con diferentes requisitos e interconectarlos entre ellos, se evita todo este proceso creado una plataforma única.

Por el lado de los usuarios, este proyecto les permite obtener un sistema que mejorara en su día a día sus tareas más habituales, generando gráficos que apreciar e interpretar la información de una manera más rápida así como eficiente, generando alertas cuando contenidos de interés de los usuarios han

sido capturados por el sistema o el sistema ha identificado algún tema de interés para el usuario.


Figura 2: Beneficios esperados por los usuarios

## 1.4. Formación para el proyecto

El presente trabajo se enmarca dentro de la tesis de máster necesaria para conseguir el título del Máster en Ingeniería de Software, Métodos Formales y Sistemas de Información ofertado por la Universidad Politécnica de Valencia. En este apasionante año y medio realizando este curso, hemos podido aprender multitud de cosas tales como ingeniería de requisitos, servicios web, inteligencia ambiental entre otros muchos temas.

Los contenidos del máster que me han ayudado a ejecutar el proyecto se dividen en 2 partes:

En la parte teórica del proyecto asignaturas como técnicas de ingeniería de requisitos, ingeniería de requerimientos y el seminario avanzado de IS han sido esenciales a la hora de capturar requisitos del cliente, proponer soluciones, documentar los requisitos del proyecto y transmitir los requisitos al resto del equipo. Además, asignaturas como "Ingeniería del lenguaje natural" me han aportado conocimientos en el análisis y procesado de textos.

En la parte práctica, asignaturas como "Proyecto de Desarrollo de Software", "Tecnología Software para Ambientes Web" y "Desarrollo de Sistemas Ubicuos e Inteligencia Ambiental" han sido inspiradoras y me han aportado ideas a lo largo del desarrollo. También asignaturas relacionadas con servicios y aplicaciones web han reforzado mis conocimientos sobre estos temas, ayudándome en el desarrollo del producto.

De igual modo asignaturas como "Técnicas de HCI (Human-Computer Interaction)" me han ayudado proponiendo mejores y más simples interfaces de usuario, lo que se ha traducido a su vez en reducción del tiempo de desarrollo, reducción de costes y mejorar la experiencia del usuario.

Finalmente S2T también ha contribuido en la formación necesaria para desarrollar este proyecto. Durante el mes de Agosto S2T me comunicó que tenía la posibilidad de participar en un curso sobre la metodología scrum. Era una gran oportunidad de extender los conocimientos previos que obtenido en el título Ingeniería Técnica en Informática de Gestión y en máster sobre metodologías ágiles.

## 1.5. Rol desarrollado en el proyecto

El principal rol que he desarrollado en el proyecto es el de líder de desarrollo pero debido a que en el proyecto hemos querido seguir y aplicar algunas de las recomendaciones de la metodología, también he estado desarrollando el rol de *scrum master*. Es por esto por lo que mis funciones han variado dependiendo de la fase o del estado proyecto.

Al principio del proyecto, S2T organizó diferentes encuentros con las organizaciones, para obtener una visión general de sus expectativas, requisitos e ideas para las cuales querían utilizar el sistema. Atender a estas reuniones y obtener los requisitos de los usuarios, fue mi primera tarea en el proyecto. Pero esta función en el proyecto fue realizada junto con el analista de sistemas y *product owner*.

Terminada esta primera fase era el momento de empezar con la iteración cero, en este tramo del proyecto mi rol fue ayudar al *product owner* a añadir los requisitos capturados en la fase anterior al *product backlog* y ayudarle a priorizarlos.

Una vez el equipo ya estaba organizado, habíamos obtenido una lista general pero no definitiva de requisitos, identificado los riesgos y definido un boceto de la arquitectura, era el momento de empezar con la primera iteración del proyecto. Desde este momento mi rol se basó en desarrollar las siguientes funciones:

- Preparar y mantener actualizada la documentación relacionada con la arquitectura y diseño del producto.

- Desarrollo del sistema. Especialmente tareas relacionadas con el *back-end* del sistema.

- Asignar el trabajo de desarrollo

- Control de calidad del desarrollo (uso de estándares y patrones) y de la metodología
- Quitar los impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada iteración.

- Facilitar las reuniones de scrum (planificación de la iteración, reuniones diarias de sincronización del equipo, demostraciones y reunión retrospectiva)

## 1.6. Estructura de la memoria

El presente proyecto se ha estructurado en cinco capítulos cuyos contenidos se resumen a continuación.

En el capítulo 2 se expondrá la propuesta de la tesina, se introducirá a las distintas organizaciones gubernamentales involucradas en el proyecto, la metodología de trabajo con la que se ha desarrollado el proyecto y finalmente se describirá el contexto tecnológico del proyecto.

En el capítulo 3 se va a introducir el caso de estudio, analizará la propuesta, expondrán la planificación y se detallarán los requisitos funcionales y no funcionales del sistema.

En el capítulo 4 se aborda como se ha conseguido desarrollar un sistema de alto rendimiento, seguro de usar y que además permita expansiones y mejoras futuras. Además, también se detallara el diseño del producto y como el equipo ha alcanzado los objetivos marcados.

El capítulo 5 comienza explicando la implementación y despliegue de la propuesta, así como el impacto del producto, estadísticas del uso y finalmente se expone un caso de éxito de utilización de Gov TA.

En el capítulo 6 se plantean y valoran los resultados obtenidos, se determinan los objetivos alcanzados y se exponen las posibles líneas de desarrollo futuro.

Finalmente, a modo de información adicional y con la intención de completar el contenido expuesto en los capítulos anteriores, se incluye un anexo donde se expone el manual del usuario, explicando con detalle las funciones del sistema y cómo los usuarios interactúan con el mismo.

# 2

# Contexto tecnológico

En este apartado se muestra una visión panorámica de la propuesta, así como una introducción a las organizaciones involucradas en el proyecto, la metodología de trabajo con la que se ha desarrollado el proyecto y el contexto tecnológico del proyecto.

## 2.1. Contexto tecnológico

A la hora de realizar el proyecto debemos considerar y evaluar detenidamente las ventajas e inconvenientes que ofrecen las distintas tecnologías, herramientas y librerías que se van a utilizar en su desarrollo.

### 2.1.1. Tecnologías

Model View Controller es un patrón de la arquitectura de las aplicaciones software, también conocido como MVC. En nuestro proyecto lo hemos utilizado principalmente para separar la lógica del negocio de la interfaz del usuario. De esta manera, hemos incrementado la reutilización de código y la flexibilidad del mismo. Para ellos hemos utilizado el *Fraemwork* ASP.NET MVC[1]

Este patrón se divide en 3 módulos que son *model, view y controller* (modelo, vista y controlador en castellano):

- Modelo: Se caracteriza por ser independiente de cualquier representación de salida o entrada.

- Vista: Muestra la información al usuario, es decir, es la interfaz del usuario.

- Controlador: Actúa como intermediario entre la vista y el modelo. Será el encargado de recibir los eventos de entrada.

[1] Podemos consultar distintos ejemplos en su sitio web http://www.asp.net/mvc

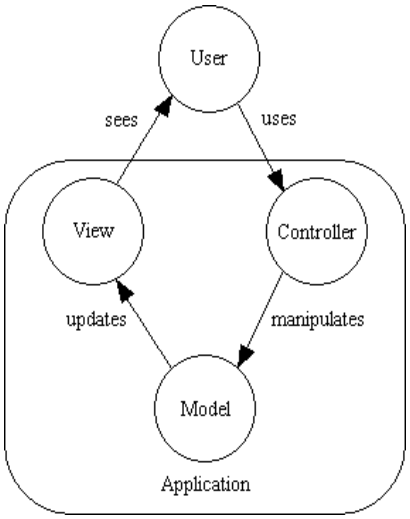



Figura 3: Model view controller

En la figura 9 se muestra un resumen gráfico del patrón. Un ejemplo de utilización se daría cuando un usuario crea una petición http desde su ordenador, esta llegara al servidor que aloja la página web, el cual se encargará de analizar la dirección URL y la despachara al controlador encargado. Aquí se realizará parte lógica de la aplicación y se generará un modelo el cual se rellenará con la información necesaria. Finalmente, el controlador enviará el modelo hacia la vista la cual será la encargada de presentar la información al usuario.

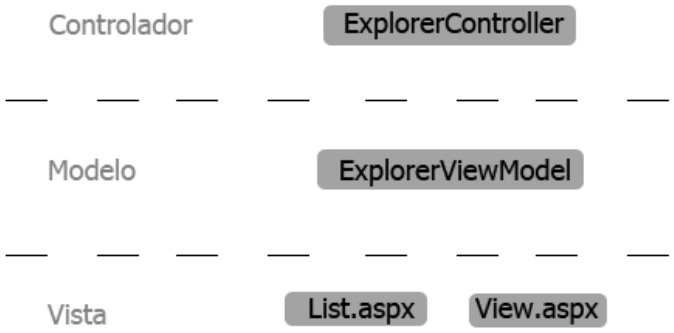



Figura 4: Ejemplo MVC

En la figura anterior se puede observar un ejemplo real que ha sido utilizado en este proyecto. En esta ocasión vemos como existe un único controlador denominado "ExplorerController", el cual tiene un modelo denominado "ExplorerViewModel". Este controlador está compuesto de dos acciones "list" y "view", las cuales tienen asociadas dos vistas con su mismo nombre.

Uno de los grandes problemas en el desarrollo de software es cómo manejar la dependencia de los objetos. La dependencia de los objetos se refiere que para crear un objeto, es necesario utilizar otro objeto y así sucesivamente.

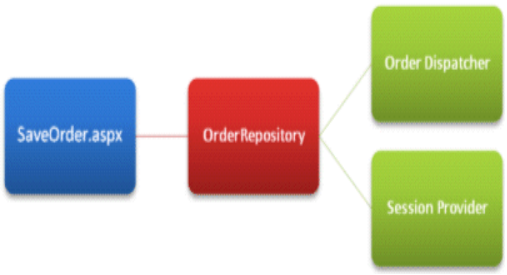



Figura 5: Dependencias de los objetos

Los beneficios que *inversion of control* nos proporciona, es la eliminación de las dependencias entre los objetos y una mayor flexibilidad en el código. La aplicación de este método de programación nos puede ser muy útil también para crear pruebas unitarias con otras herramientas como NUnit

Existen distintas maneras de aplicar *inversión of control,* en nuestro caso hemos decidido aplicarlo mediante Windsor Container[2] .

### 2.1.2. Librerías

OneMap es el proyecto más importante en el área sistemas geoespaciales para las agencias gubernamentales y el sector privado en Singapur. Este proyecto fue diseñado por la universidad Nanyang Polytechnics con la colaboración de varios organismos y agencias gubernamentales que participan activamente contribuyendo con información.

La colaboración entre las organizaciones del gobierno ha permitido construir un mapa común de Singapur que ofrece una gran variedad de servicios y funciones que permiten a los usuarios buscar y navegar de forma inteligente lugares de interés como museos, servicios de cuidado de niños, parques y centros deportivos.

La API de OneMap permite incrustar un mapa interactivo en Gov TA. Utilizando el amplio conjunto de APIs en JavaScript podemos ser capaces de visualizar nuestros propios datos a la vez de hacer uso de los servicios que este OneMap nos ofrece

[2] Podemos obtener más información en http://www.castleproject.org/container/

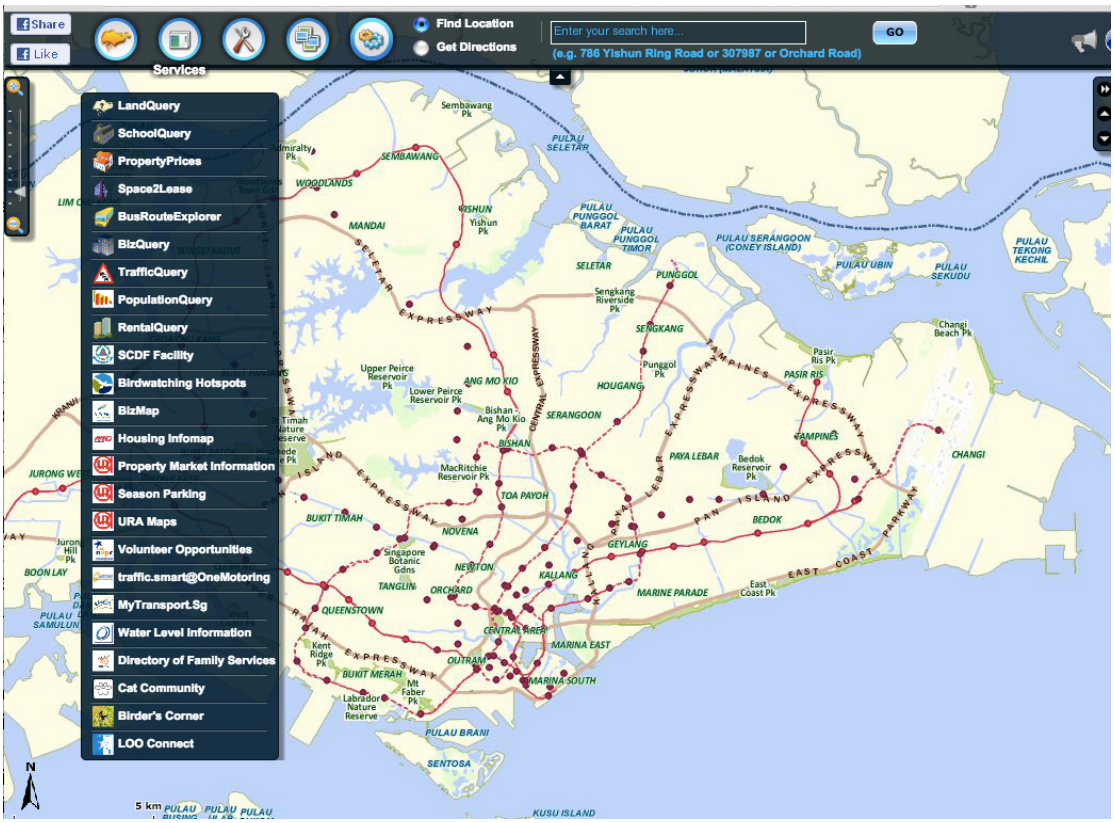



Figura 6: OneMap

Uno de los retos que teníamos era crear páginas extraordinariamente dinámicas con el menor código posible. Para conseguir esta meta debíamos escoger los *frameworks* más adecuados que se adaptaran no tan solo al objetivo del proyecto, sino también que el equipo supiera utilizarlos eficientemente.

Anteriormente en la parte Front-end de las aplicaciones web únicamente teníamos a jQuery (además de otras librerías parecidas como Mootools, Prototype,…) para ayudarnos con el código JavaScript del cliente. Podíamos manipular el DOM de una forma más sencilla, añadir efectos, llamadas Asynchronous JavaScript And XML (AJAX), etc. Pero no teníamos un patrón a seguir. Todo el código JS iba en funciones que íbamos creando según necesitáramos, lo que provocaba que con el tiempo el código fuera difícilmente manejable y se convirtiera en el temido *spaguetti code*.

Por suerte surgieron *frameworks* que nos ayudaban a prevenir *spaguetti code*. y nos ayudaban a separar conceptos. Tres JavaScript *frameworks* fueron evaluados

- **Meteor**[3]: Es una plataforma para crear aplicaciones web en tiempo real construida sobre Node.js. Meteor se localiza entre la base de datos de la aplicación y su interfaz de usuario y se asegura de que ambas partes estén sincronizadas.

[3] Meteor https://www.meteor.com/

El equipo de desarrollo reconoció que esta es una plataforma muy potente y que abstrae muchas de las molestias y dificultades que nos encontramos habitualmente en el desarrollo de aplicaciones web, pero sin embargo decidimos no utilizarla debido a que ningún miembro del equipo había tenido experiencia con ella.

- **AngularJS**[4]: Es un framework MVC de JavaScript para el Desarrollo Web Front-end que permite crear aplicaciones SPA (Single-Page Applications). Entra dentro de la familia de frameworks como BackboneJS o EmberJS.

  El sistema de plantillas en AngularJS es diferente del utilizado en otros *frameworks*. Por lo general es el servidor el encargado de mezclar la plantilla con los datos y devolver el resultado al navegador. En AngularJS el servidor proporciona los contenidos estáticos (plantillas) y la información que se va a representar (modelo) y es el cliente el encargado de mezclar la información del modelo con la plantilla para generar la vista.

  A pesar de sus ventajas, decidimos no utilizar este *framework* debido a que consideramos que no es una librería fácil y requeriría un mayor tiempo adaptarse a esta.

- **KnockoutJS** [5] : Es una librería JavaScript, que ayuda a poner en funcionamiento el modelo MVVM. Sus características principales son las que nos ayudaron a confirmar que este era el JavaScript *framework* que debíamos utilizar:

  - Sistema de *bindings* declarativos, expresados sobre los propios *tags* HTML usando atributos data-bind, que permiten crear un vínculo uni o bidireccional entre elementos de la interfaz de usuario y propiedades o acciones del View-Model, que se actualizarán de forma automática ante cambios.

  - Seguimiento de dependencias, capaz de detectar los cambios realizados tanto en la Vista como en el Modelo-Vista y propagarlos hacia todos los objetos o elementos dependientes.

  - Por supuesto, la interfaz de usuario se actualiza automáticamente para reflejar los cambios en el View-Model.

  - Incluye un sistema de plantillas que facilita la creación porciones de vistas reutilizables.

[4] AngularJS HTML enhanced for web apps, https://angularjs.org/
[5] KnockoutJS, http://knockoutjs.com/

Kendo UI [6] es un *framework* basado en jQuery y HTML5 para la construcción de aplicaciones web modernas. Este *framework* cuenta con un montón de *widgets* y ejemplos de uso de los mismos que permiten un rápido desarrollo de herramientas o aplicaciones webs.

Una de las principales razones para escoger esta librería para el desarrollo de Gov TA es el uso de nuevas tecnologías como Scalable Vector Graphics (SVG) o canvas y la sencillez y rapidez con la que se pueden representar datos mediante gráficos de barras, área, líneas, etc.

**KeyLines**[7] es una tecnología para la construcción de aplicaciones que evalúan relaciones entre los datos. El uso de esta librería permite crear aplicaciones que simplifican la interpretación de los datos, encontrar pautas y la generación de conocimiento útil. KeyLines ha sido utilizada en investigación de actividades delictivas ( detección de fraudes, contraterrorismo e inteligencia ), análisis de la seguridad informática, y la investigación médica.

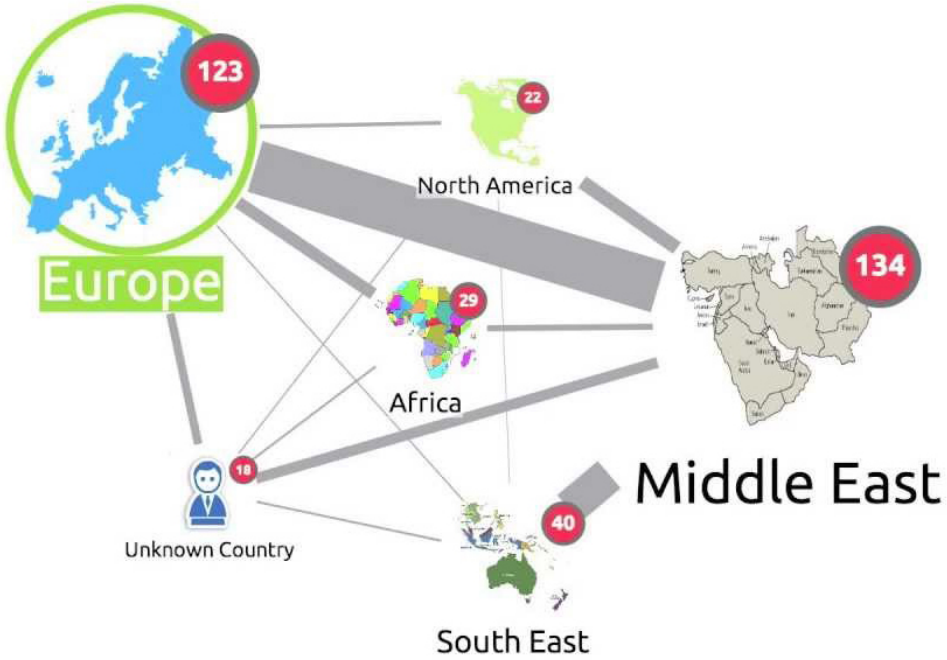



Figura 7: Keylines

**Lexalytics**[8] es una herramienta *Natural language processing* (NLP) que permite analizar y clasificar texto no estructurado a través de un conjunto de técnicas lingüísticas (sintácticas y semánticas), analíticas y predictivas. Gov Ta

[6] Kendo UI HTML5, http://www.telerik.com/kendo-ui
[7] KeyLines Network Visualization, http://keylines.com/
[8] Lexalytics, Inc http://www.lexalytics.com/

utiliza Lexalytics como base para el análisis de textos y sus principales aportaciones al proyecto son las siguientes:

- **Análisis de sentimiento:** es parte de lo que se conoce como *sentiment analysis*. El objetivo de esta característica es generar un valor conocido como *sentiment score* que determinar si el documento es positivo o negativo. Este proceso se realiza a partir de una "red semántica", en la cual se generan listados de términos positivos y negativos, con distintos grados de "positividad" y "negatividad". Por ejemplo, "excelente" se considerará "más positivo" que "bueno". La proximidad de una palabra con cada uno de estos "indicadores de sentimiento" determinará el *sentiment score*. Por lo tanto, la expresión "excelentes previsiones" tendrá una puntuación positiva bastante superior a "buenas previsiones". Sin embargo, hay que destacar que el *sentiment score* no es siempre acertado, básicamente porque el lenguaje es ambiguo. La misma palabra puede referirse a conceptos distintos según el contexto. Sin mencionar los infinitos matices relacionados con la sensación que se quiere transmitir al receptor (sarcasmo, ironía, crítica, ira, agradecimiento, etc.).

- **Extracción de entidades**: indica la capacidad de detectar entidades (ej. localidades geográficas, nombres de empresas, etc.) en un texto. El sistema más simple se basa en un listado de términos que el programa buscará en los textos a analizar, mientras los sistemas más complejos pueden ser "entrenados" para reconocer las entidades automáticamente e identificar, por ejemplo, "NY", "NYC" y "New York" como entidad única.

- **Recapitulación:** Esta funciona permite resumir ordenadamente artículos de gran extensión.

Una de las expectativas de los usuarios sobre el producto es la de realizar búsquedas sobre todo tipo de información que se pueda representar de forma textual. Lucene [9] ha sido ampliamente usado para satisfacer este tipo de requisitos.

Para esta librería no es importante el origen de los datos, el formato o el idioma, siempre y cuando se puedan convertir en texto. Esto significa que podemos usar Lucene para buscar datos en páginas web, documentos almacenados en el sistema local de archivos, archivos de texto simple, documentos Microsoft Word, HTML o PDF.

Lucene se compone de dos procesos o fases para conseguir realizar búsquedas efectivas:

[9] Apache Lucene https://lucene.apache.org/

- **Indexación**: Este proceso consiste en analizar y extraer de entre toda la información disponible, la verdaderamente relevante para el sistema. Posteriormente, con esa información se crea el índice a partir del cual se realizarán las búsquedas.

  El índice es una estructura de datos que permite acceso rápido a la información, algo similar semánticamente a lo que podría ser el índice de un libro.

- **Búsqueda**: Este proceso consiste en consultar el índice para obtener los documentos donde aparecen unas determinadas palabras o bien concuerdan con una determinada expresión de consulta.

### 2.1.3. Herramientas

Dentro de las bases de datos NoSQL, probablemente una de las más famosas sea MongoDB[10]. Esta es una base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Es decir, los documentos de una misma colección (similar a una tabla de una base de datos relacional), pueden tener esquemas diferentes. Fue esta una de las principales razones para utilizar MongoDB en nuestro proyecto. Sin embargo, esta no es la única base de datos que hemos utilizado en el proyecto. Nuestra solución apuesta por una combinación de base de datos NoSQL y SQL.

Microsoft SQL Server fue la apuesta para la base de datos relacional de nuestro proyecto y de las principales razones para el uso de esta es la fácil integración con .NET y Entity Framework.

La herramienta que más nos ayuda a la hora de la implementación y desarrollo del producto ha sido Visual Studio 2013 y extensiones como Resharper que permiten extender ampliamente las capacidades de desarrollo, análisis de código, refactorización, navegación, búsqueda, entre otras tantas tareas.

Por otra parte, también encontramos otra herramienta que nos ha resultado de gran ayuda en las pruebas de la aplicación, esta es NUnit. Esta aplicación es un *framework* para realizar pruebas, que no depende en lo absoluto de Visual Studio, el cual también aporta herramientas para realizar la misma

[10] MongoDB es una base de datos NoSQL desarrollado bajo el concepto de código abierto https://www.mongodb.org/

función, pero en nuestro caso hemos preferido utilizar esta opción debido a que ya conocíamos anteriormente su funcionamiento y en mi caso ya la había utilizado previamente en la universidad. El motivo del uso de NUnit es la automatización de pruebas.

El servidor web escogido ha sido Internet Information Server (IIS) debido a su soporte para páginas Active Server Pages (ASP), ASP.NET y su fácil instalación y configuración.

## 2.2. Servicios web

Existen numerosas definiciones de lo que es un servicio Web [2,3,4], lo que muestra su complejidad a la hora de dar una definición que englobe todo lo que implica. Una de las más completas es la del W3C, la cual explica que "Un servicio Web es un sistema software diseñado para soportar a la interoperabilidad máquina-máquina sobre la red. Este tiene una interfaz descrita en un formato procesable por una máquina. Otros sistemas interactúan con el servicio Web de la manera especificada por su descripción utilizando mensajes SOAP, por lo general transmitidos a través de HTTP con una serialización Extensible Markup Language (XML) y unión con otros estándares relacionados con la Web" [5]

Basándonos en la idea de que la interfaz del usuario está construida principalmente con componentes desarrollados en JavaScript, la aplicación debe ser capaz de realizar peticiones al servidor y obtener respuesta de este en segundo plano (sin necesidad de recargar la página web completa) y usar esos datos para, a través de JavaScript, modificar los contenidos de la página creando efectos dinámicos y rápidos. Esto es posible de conseguir realizando AJAX desde el navegador. Esta tecnología busca evitar las demoras propias de las peticiones y respuestas del servidor mediante la transmisión de datos en segundo plano usando un protocolo específicamente diseñado para la transmisión rápida de pequeños paquetes de datos.

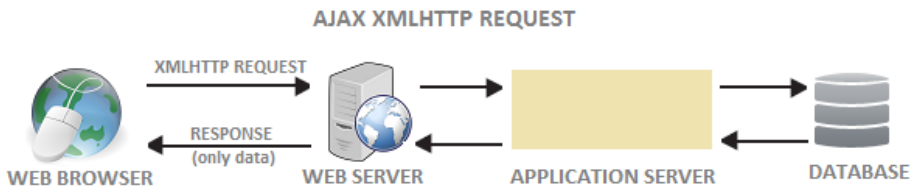



Figura 8: AJAX

En la figura anterior vemos lo que sería un esquema de comunicación usando Ajax, donde el cliente tiene una página web cargada en el navegador web y en segundo plano crea una petición al servidor para que le envíe un paquete de datos que necesita. Una vez el servidor ha procesado la petición, le enviará la

respuesta mucho más rápido, ya que no tiene que elaborar una página web completa, sino solamente preparar un paquete de datos. Por tanto, el tiempo de respuesta es más rápido. Una vez recibido el paquete de datos, el cliente los usa para cambiar los contenidos que se estaban mostrando en la página web. A continuación, se muestran algunas de las ventajas y desventajas de este modelo.

- **Mejor experiencia de usuario.** Este modelo permite que las páginas se modifiquen sin tener que volver a cargarse, dándole al usuario la sensación de que los cambios se producen instantáneamente. Este comportamiento es propio de los programas de escritorio a los que la mayoría de los usuarios están más acostumbrados. La experiencia se vuelve mucho más interactiva.

- **Optimización de recursos.** Al no recargarse la página se reduce el tiempo implicado en cada transacción y también se utiliza menos ancho de banda.

- **Alta compatibilidad.** Ajax es soportado por todos los navegadores modernos

- **El desarrollo de aplicaciones web se puede volver más complejo.** Los sistemas que utilizan Ajax suelen requerir de un mayor tiempo de desarrollo.

Como se ha descrito anteriormente la X de AJAX se refiere a XML porque al principio el formato más usado para intercambiar datos era XML, pero hoy en día el formato más frecuente seguramente sea JavaScript Object Notation (JSON). JSON es un formato mucho más ligero que XML y, además, tiene la ventaja de ser "nativo" para JavaScript.

Uno de los detalles que tenemos que en cuenta es que por motivos de seguridad, usando AJAX únicamente se pueden realizar peticiones al mismo dominio desde el que se ha cargado la página que origina la petición. Es decir, si cargamos una página desde www.domain.com, solamente se podrán hacer peticiones a URLs de la forma www.domain.com/lo-que-sea.

Para resolver el problema anterior apareció CORS. Este permite realizar peticiones a otros dominios siempre y cuando el dominio de destino esté de acuerdo en recibir peticiones del dominio de origen. Es una tecnología implementada a nivel de navegador, que intercambia ciertas cabeceras HTTP con el servidor de destino para saber si debe permitir o no el intercambio de datos.

Los dos métodos HTTP más comunes para el envío de una petición a un servidor son GET y POST.

- El concepto de uso de GET es el de obtener información del servidor, es decir, se utiliza para traer datos que están en el servidor.

- El concepto de uso de POST es el de enviar información desde el cliente para que sea procesada, o que se actualice o agregue información en el servidor.

## 2.3. Arquitectura

En la ingeniería del software encontramos distintos tipos de arquitecturas, las más habituales que podemos encontrar en cualquier software son, monolítica, cliente-servidor y tres capas. Cada una de estas contiene a su vez ciertas ventajas y desventajas[11] .

Durante el desarrollo de esta propuesta hemos seguido una arquitectura de 3 capas. Esto permite separar la capa de presentación, la capa de negocio y los datos. En ocasiones esta separación puede ser física, es decir, podemos tener en distintos ordenadores cada una de las capas, con lo cual obtenemos una aplicación distribuida físicamente.


Figura 9: Arquitectura en 3 capas

De esta manera, nuestra aplicación queda dividía en 3 partes diferenciadas como se aprecia en la figura anterior:

- La capa de presentación se refiere al mecanismo de interacción del usuario con el sistema. Su función es la de presentar la información al usuario y tomar los eventos que el usuario genere. Los tipos de interfaces

[11] Ventajas y desventajas de las arquitecturas
http://www.mitecnologico.com/Main/ArquitecturaAplicacionesWeb

software más comunes son las aplicaciones web y las aplicaciones de ventana.

- La lógica de negocio se refiere al conjunto de reglas que determina cómo funciona un sistema, según su naturaleza, bajo que parámetros y condiciones de acuerdo con las necesidades de los clientes o los usuarios. Los elementos fundamentales de esta capa son los objetos de dominio, los cuales son simples objetos que solamente contienen los datos que representan.

- La capa de acceso a datos se refiere a la media a través del cual podemos acceder y manipular los datos existentes en un sistema.

El uso de esta arquitectura a lo largo del desarrollo de esta propuesta nos ha aportado los siguientes beneficios:

- El desarrollo se realiza en distintos niveles permitiendo el desarrollo en paralelo.

- La aplicación es más robusta debido al encapsulamiento.

- Las tareas de mantenimiento y soporte son más sencillas.

- Mayor flexibilidad, se pueden añadir y eliminar funcionalidades de una forma más simple.

## 2.4. Metodología de desarrollo de software

Cada proyecto de software es diferente al resto, por esta razón en una empresa en la que distintas personas participan en muchos proyectos es necesario adoptar una metodología o lo que es lo mismo, establecer un marco de trabajo para estructurar, planificar y controlar el proceso de desarrollo.

Hasta el momento S2T había utilizado el desarrollo en cascada o iterativo como metodología de trabajo. Los resultados habían sido positivos pero creíamos que era posible obtener mejores resultados con scrum y por este motivo desde un principio del proyecto decidimos adoptar esta metodología para la gestión de desarrollo del producto.

La elección de escoger scrum como metodología de trabajo fue basada también en factores como los requisitos poco definidos del proyecto, el número de miembros del equipo, entrega modulares y la reciente formación que tuve en el curso de *Scrum master*.

Scrum es una metodología ágil de trabajo que nació en el ámbito del desarrollo de software y se basa en un proceso iterativo e incremental y sus fundamentos son los siguientes:

- Cada *sprint* tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas.

- Scrum define 3 roles, el primero de ellos es el *scrum master* que es quien guía al equipo para que cumpla las reglas y procesos de la metodología. El segundo rol es *product owner* es el representante de los accionistas y clientes que usan el software. El último rol es el equipo que es un grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta

- El *product owner* crea una lista de deseos o requisitos llamada *product backlog*, en esta lista se encuentran las historias de usuario ordenadas por su prioridad.

- El inicio de cada *sprint* el equipo se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo.

- De existir una constante colaboración y comunicación entre el equipo y el *product owner*.

- Al final de cada iteración se demuestra al cliente el resultado obtenido, de manera que pueda tomar las decisiones necesarias en función de lo que observa y del contexto del proyecto en ese momento.

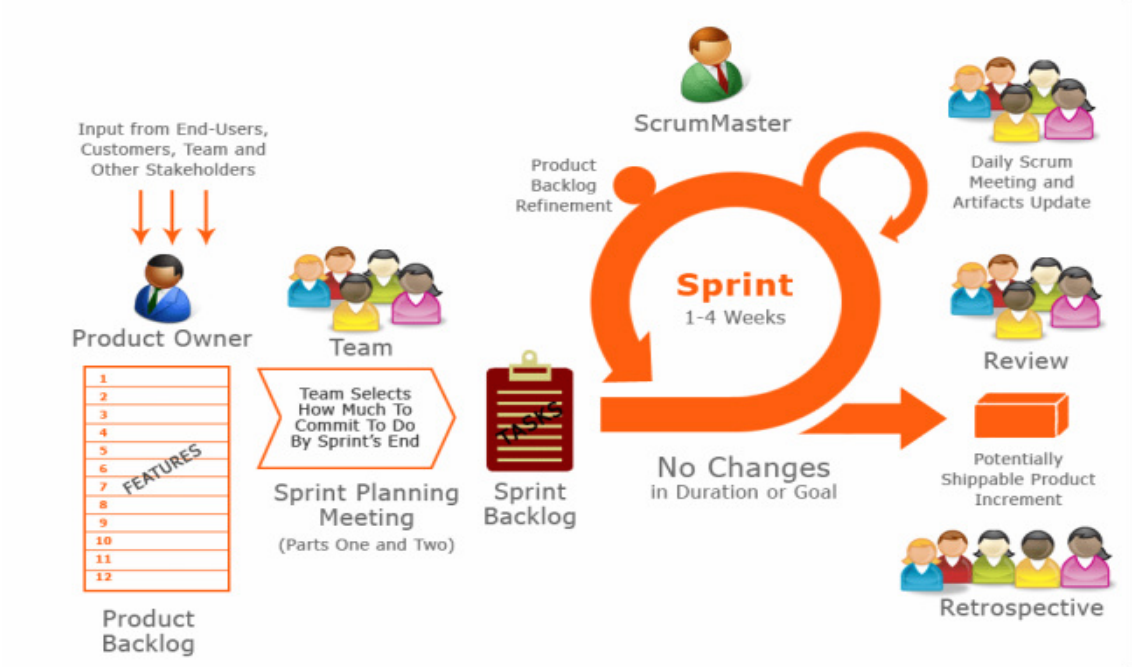



Figura 10: Metodología de trabajo scrum

## 2.4.1. Scrum meetings

Al comienzo de cada *sprint*, el *product owner* y el equipo tienen una reunión de planificación de *sprint*. En esta reunión se negocian que historias de usuario se van a tratar de convertirse en producto.

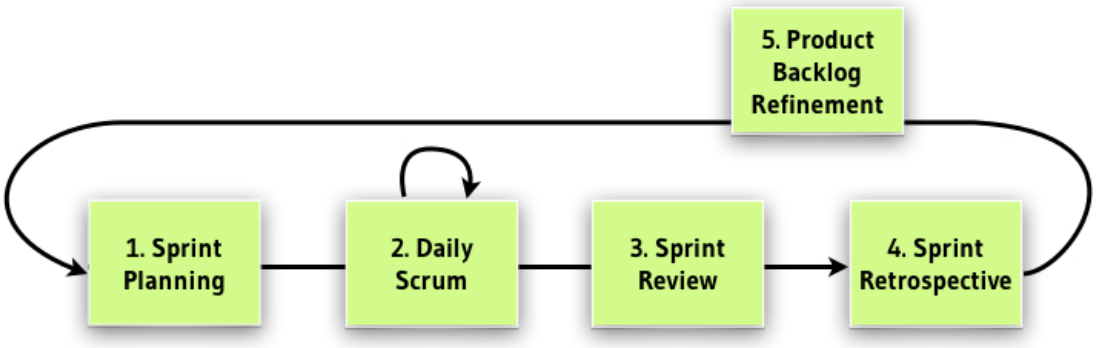



Figura 11: *Scrum meetings*

El *product owner* es responsable de declarar qué elementos son los más importantes para el negocio y asignar la prioridad. Por otra parte, el equipo es responsable de la selección de la cantidad de trabajo.

Una vez seleccionadas las historias de usuario, el equipo crea una lista inicial de tareas por cada una de ellas y las coloca en el *Sprint Backlog*.

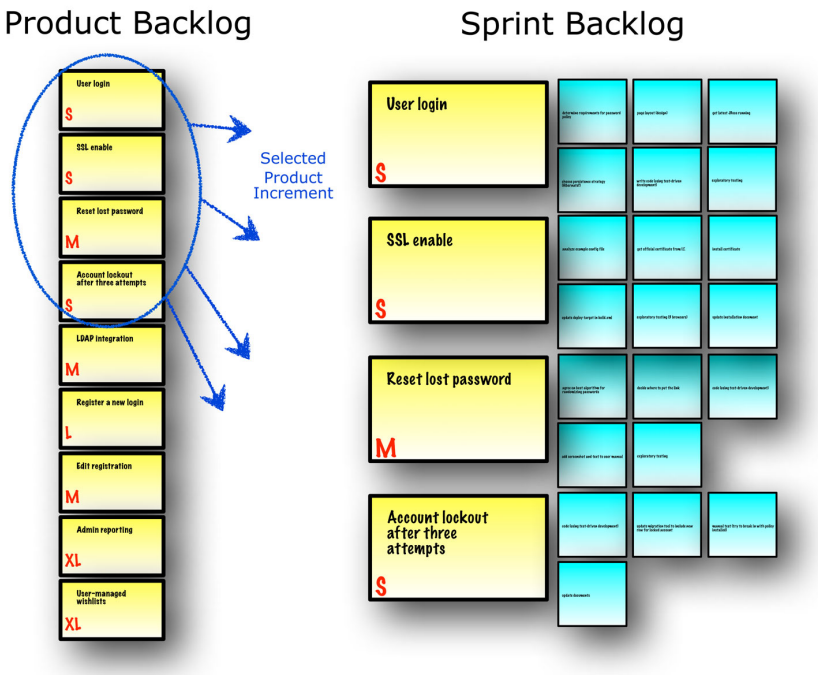



Figura 12: *Product backlog* y *Sprint Backlog*

Uno de los problemas a los que nos enfrentamos durante los primeros *scrum meetings* fue la duración de la reunión. Debido a que buscábamos soluciones a problemas a los que algún miembro del equipo se estaba enfrentando y esto no resultaba productivo para aquellos miembros del equipo que no estaban relacionados con el problema. Por esa razón para no alargar la duración de la reunión y que este resultara productiva para todos los miembros del equipo se decidió abarcar los problemas una vez acabada la reunión y con solo aquellos miembros involucrados.

Una vez acabada la reunión una de mis tareas era la de actualizar el *Sprint Backlog*. Esta lista permite hacer visible a todos los miembros del equipo el estado en el que se encuentra el *sprint*.

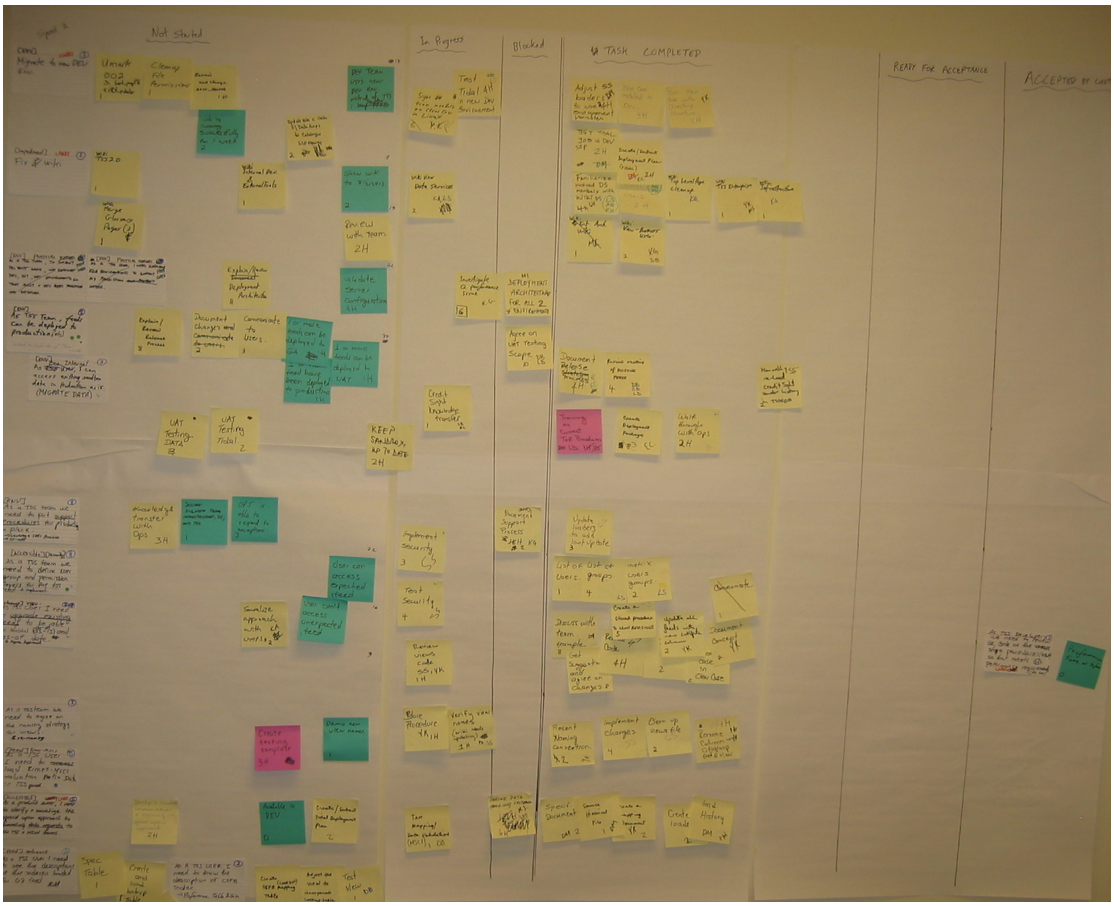



Figura 13: *Sprint Backlog*

En scrum, se requiere por cada *sprint* un incremento del producto. Esto significa que el equipo ha producido una pieza de software, la cual ha pasado las pruebas y es potencialmente entregable. El equipo se reúne con el *product owner* y con otros *stackeholders* para mostrar aquellas nuevas funcionalidades logradas durante el *sprint*.

En el equipo siempre intentábamos tener una reunión un poco informal, por ese motivo no utilizamos diapositivas ni informes, simplemente mostrábamos el producto.

Una vez terminada la demostración de las nuevas funcionalidades, el *product owner* evalúa el proyecto en función de los objetivos marcados durante la reunión de planificación del *sprint*. Esta es la oportunidad de inspeccionar y adaptar el producto a medida que emerge y es aquí donde el equipo recibía muchas observaciones o comentarios sobre las funcionalidades mostradas.

Cada *sprint* termina con una retrospectiva. En esta reunión, el equipo reflexiona sobre la iteración, inspeccionan su comportamiento y toma medidos para seguir mejorando en la siguiente *sprint*.

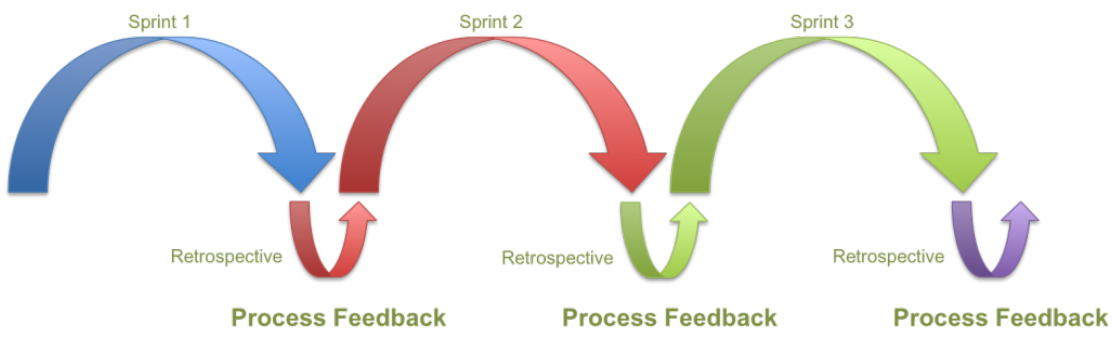



Figura 14: Efecto esperado de la reunión retrospectiva

Hay muchas maneras de conducir esta reunión, nosotros intentábamos hacerlo de una manera sencilla creando una lista de acciones o cosas que deberíamos empezar a hacer o que deberíamos seguir haciendo o por el contrario que no deberíamos seguir haciendo.

Estoy convencido que gracias a esta reunión se mejora la calidad y el ambiente de trabajo. Además, pusimos a prueba nuevas ideas, que algunas funcionaron y otras simplemente no lo hicieron, pero al menos tuvimos la oportunidad de intentarlo.

### 2.4.2. Valores y buenas prácticas aplicadas al proyecto

Scrum es una metodología para la gestión de desarrollo de productos en la que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

- **Compromiso:** El equipo es autosuficiente, y por esta razón al inicio de cada iteración el equipo se compromete a convertir en producto aquellas historias de usuario que selecciona.

- **Respeto y franqueza:** A medida que trabajamos juntos compartimos éxitos y fracasos. También es importante mencionar que el equipo debe expresar sus preocupaciones.

- **Scrum póker:** Es una técnica para estimar el esfuerzo o el tamaño de las historias de usuario. Los miembros del equipo hacen estimaciones con las cartas, en lugar de hablar en voz alta. Las tarjetas se revelan, y se discuten a continuación las estimaciones.


Figura 15: Scrum *póker*

La razón para usar utilizar esta técnica es evitar la influencia de los demás participantes, ya que obliga al equipo a pensar de manera independiente y proponer sus números de forma simultánea.

- **Pruebas:** La ejecución automatizada de pruebas ha sido un elemento clave en el proyecto. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema.

  Uno de los objetivos principales de las pruebas no es corregir errores, sino prevenirlos. Elaborar las pruebas antes de escribir el código exige pensar por adelantado cuáles son los problemas más graves que se pueden presentar, y cuáles son los puntos dudosos. Esto evita muchos problemas y dudas, en lugar de dejar que aparezcan "sobre la marcha".

- **Diseño sencillo.** El objetivo es utilizar el diseño más sencillo que consiga que todo funcione. Se evita diseñar características extras porque a la hora de la verdad la experiencia indica que raramente se puede anticipar qué necesidades se convertirán en reales y cuáles no.

  Kent Beck define [6] el diseño simple como:

- Pasa todos las pruebas.
- No contiene código duplicado.
- Deja clara la intención de los programadores en cada línea de código.
- Contiene el menor número posible de clases y métodos.

- **Refactorización*:*** es una actividad constante que consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, legible y no duplicar el código. La refactorización también se utiliza cuando resulta conveniente modificar código existente para hacer más fácil implementar nueva funcionalidad.

- **Estándares de codificación:** Los estándares de codificación mantienen el código legible para los miembros del equipo, facilitando los cambios. El objetivo es que el código del sistema se vea como si fuera escrito por una única persona muy competente.

# 3

# Caso de estudio

Una vez estudiado el contexto tecnológico de este trabajo, se va a introducir el caso de estudio, analizará la propuesta, expondrá la planificación y se detallarán los requisitos funcionales y no funcionales del sistema.

## 3.1. Descripción general del caso de estudio

Tras investigaciones preliminares de mercado, se identificó que el gobierno de Singapur no disponía de un sistema que permitiera capturar, almacenar, procesar y analizar contenidos textuales no estructurados recogidos de recursos abiertos, es decir, de información disponible públicamente.

Por otra parte, también se identificó que no existía un organismo específico para temas como el OSINT sino que diferentes organismos tenían departamentos específicos para esta tarea. Algunos países como Estados Unidos han creado órganos específicos como el National Intelligence Open Source Center (OSC), o Australia con su National Open Source Intelligence Centre (NOSIC).

Basándose en los anteriores indicios, S2T decidió proponer un sistema multi-organismo que permitiera a las organizaciones del gobierno a mantenerse al tanto de las tendencias de los medios sociales, tales como redes sociales, foros de debate o webs corporativas. Esta potente herramienta les permitiría descubrir patrones de comportamiento, tendencias, detectar desviaciones y monitorizar discusiones.

### 3.1.1 Organizaciones y sistemas relacionados con el proyecto

Fueron 3 las organizaciones que decidieron estudiar y participar en el proyecto.

La primera de ellas y que más involucrada ha estado en el proyecto es el Ministry of National Development (MND) que dirige e implanta las políticas relacionadas con el desarrollo de la vivienda, cohesión de las comunidades y la planificación urbana. Sus objetivos son la mejora de la calidad de viviendas e infraestructuras para los ciudadanos, mejorar los lazos entres las diferentes comunidades y desarrollo de espacios verdes.

La segunda organización que ha participado en el proyecto es National Environment Agency (NEA) responsable de la mejora y el mantenimiento del medio ambiente en Singapur. Desarrolla y lidera iniciativas y programas ambientales a través de asociaciones y los sectores público y privado. Se ha comprometido a motivar a cada individuo a asumir su papel en el medio ambiente y cuidarlo como forma de vida.

La tercera organización que estudio la propuesta de S2T fue Infocomm Development Authority of Singapore, la cual es responsable del desarrollo y crecimiento del sector de las telecomunicaciones en Singapur.

### 3.1.2 Propuesta

Gov TA consta de dos aplicaciones web. La primera aplicación web está disponible para los usuarios en general, en ella podrán llevar a cabo las funciones operativas de Gov TA. La segunda la aplicación web permitirá la administración del sistema Gov Ta con acciones tales como la gestión de usuarios, seguimiento de auditoría, corrección de sentimiento y entidades. Cada usuario tiene asignado un rol, esto asegura que los usuarios solo puedan realizar acciones que se les conceden hacer.

Los usuarios del sistema podrán obtener datos precisos sobre las últimas noticias, debates y tendencias que están ocurriendo en redes sociales (Facebook, Twitter o YouTube), portales de noticias, blogs o web corporativas. Nuestros rastreadores web o *crawlers* están diseñados para escanear internet de manera eficiente y anónima a través del uso de distintos *proxies*.

Archivos CSV o Excel también son capturados por el sistema. De manera que los usuarios pueden subir archivos o carpetas con información confidencial que permanecerá aislada del resto de usuarios de la organización.

Una vez los datos han sido recogidos en el sistema, estos son automáticamente clasificados y categorizados en diferentes temas. Esto permite al usuario intuitivamente filtrar de manera sencilla el contenido e identificar los resultados relevantes. Además de la ganancia de productividad de los usuarios en el análisis de los datos recogidos de recursos abiertos, Gov TA permite a los usuarios descubrir patrones ocultos, tendencias de la opinión del público y las cuestiones que preocupan a los ciudadanos.

Los usuarios pueden definir alertas basadas en reglas. De esta forma, cuando se cumple una regla, el sistema genera una alerta que permite notificar al usuario acerca de la información relevante (por ejemplo, un documento utiliza una serie de palabras clave o el sistema ha encontrado una nueva información sobre alguna persona de interés para el analista o usuario del sistema).

El análisis de los datos se representa mediante múltiples visualizaciones y cuadros estadísticos con el fin de identificar patrones de comportamiento, detectar desviaciones o actividades web anormal. El analista puede comparar los resultados a través del tiempo, utilizando diferentes filtros o usando diferentes fuentes de información.

El sistema cuenta con soporte para múltiples espacios de trabajo que permiten realizar análisis individuales o compartidos con otros departamentos u

organizaciones. De este modo diferentes usuarios pueden participar en el análisis de los datos.

## 3.2. Requisitos

Una de las primeras actividades en el desarrollo de proyectos de software es la fase de especificación de requisitos, donde se detallan los requerimientos, expectativas y las limitaciones que se esperan del producto.

Sin embargo, en scrum no se define una fase de especificación de requisitos, pero habitualmente se realiza una iteración conocida como *sprint zero*, la cual presenta una perfecta oportunidad para reunir y organizar los requisitos y preparar el equipo.

En el *sprint* zero fueron tres las actividades seguidas para obtener una lista de historias de usuario *o blacklog items* que posteriormente fueron añadidos al *product backlog*

- **Extracción de ideas**: A través de varias reuniones con usuarios y representantes de las organizaciones se descubrieron parte de los requisitos del sistema.

- **Análisis**: Una vez obtenidas las ideas y expectativas de los usuarios se realizó un análisis entre algunos de los miembros del equipo. La finalidad de esta actividad es resaltar los posible problemas y buscar alternativas y soluciones.

- **Especificación**: Se describieron brevemente las historias de usuario que formaran parte del *product backlog.*

### 3.2.1. Requisitos funcionales

A continuación se enumeran los requisitos funcionales más importantes que los representantes de los usuarios del sistema han identificado

- El sistema debe ser capaz de generar gráficos de barras, líneas, circulares, diagramas de dispersión, burbujas, radar, tablas, nube de etiquetas y series de tiempo.

- El sistema debe generar *dashboards* o informes que permitan a los usuarios obtener un análisis de sus temas de interés. A su vez los usuarios deben ser capaces de obtener los datos subyacentes, es decir, aquellos contenido que se ocultan detrás de los gráficos.

- El sistema debe permitir a los usuarios filtrar los datos de manera dinámica. Se deberán proporcionar filtros de búsqueda o rango.

- Los usuarios deberán ser capaces de añadir anotaciones a los gráficos.

- El motor de análisis de texto deberá ser capaz de analizar contenidos textuales no estructurados en el contexto local de Singapur y tiene que ser capaz de manejar Singlish[12]

- Errores tipográficos: El sistema deberá permitir la corrección de errores tipográficos.

- El sistema debe aportar funcionales para realizar un análisis demográfico de los datos.

- Autores influyentes en las redes sociales: El sistema permitirá a los usuarios identificar personas de influencia en las diferentes redes sociales.

- El etiquetado de contenidos: El sistema debe permitir el etiquetado manual de contenidos de medios sociales y CRM.

- El sistema permitirá a los usuarios exportar los gráficos y resultados en formato Microsoft Excel y PDF.

- Alertas y notificaciones: El sistema deberá tener un mecanismo de alerta para notificar a los usuarios si un indicador ha sido superado o no se llegan a los límites establecidos.

- El usuario debe ser capaz de clasificar aportaciones de otros medios de comunicación no social en categorías predefinidas utilizando definiciones personalizados y específicas para cada organismo

### 3.2.2. Requisitos no funcionales

Los requisitos no funcionales, aquellos requisitos que describen las restricciones y obligaciones que el sistema debe contener. También fueron capturados durante las reuniones con los representantes de los usuarios. Seguidamente se muestra un listado de requisitos no funcionales:

- La función de clasificación de documentos en temas o *topics* será capaz de clasificar con un nivel de precisión del 80%.

[12] Inglés Coloquial en Singapur https://en.wikipedia.org/wiki/Singlish

- La solución debe ser 100% Web Based y toda la parametrización y administración debe realizarse desde un navegador

- Cuando haya hasta 50 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 20 segundos

- El sistema debe implementarse en el nuevo servicio G-Cloud[13].

- El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible.

- El sistema debe presentar mensajes de error que permitan al usuario identificar el tipo de error y comunicarse con el administrador del sistema.

- El sistema deberá estar complemente documentado, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de administración y de usuario.

## 3.3. Planificación y plan de entregas

Debido a que el estimado tiempo de captura de requisitos, desarrollo, pruebas y despliegue del producto era mayor de 8 meses, S2T propuso a las diferentes agencias dividir el proyecto en 2 fases. Esto generó un debate, ya que se deberían decir que funcionalidades se incluirían en las diferentes fases del producto.

Se decidió que la primera fase del producto genera la base del producto, proporcionará a los usuarios una interfaz que permite obtener información sobre los datos creando distintos gráficos y permitiera crear alertas, de forma que notificara a los usuarios por correo electrónico eventos de interés. El resto de funcionalidades quedaban para la segunda fase del proyecto.

Realizar este plan de entregas no suponía ningún problema, ya que el equipo había decidido usar scrum como metodología de trabajo, pero se debían tomar medidas para conseguir los objetivos de la primera fase:

- El *product owner* dio mayor prioridad a los requisitos que debían entregarse en esta prioridad, dejándose aconsejar por el equipo.

[13] El Gobierno de Singapur ha puesto en funcionamiento el modelo de distribución de *software as a service* en 2013 para el uso de sitios web gubernamentales

- El equipo se comprometió a escoger aquellos requisitos con mayor prioridad en el *product backlog* y al final de cada iteración mostrar el resultado generado, de manera que se pueda tomar las decisiones necesarias en función del contexto del proyecto en ese momento.

A continuación se muestra la planificación del proyecto, el cual tendrá una duración total de 36 meses.

| Tarea | 2 wks | 2 wks | Month 2 to 4 | Month 5 to 6 | Month 7 to 9 | Month 10 to 11 | Month 12 to 36 |
|---|---|---|---|---|---|---|---|
| **Planificación del proyecto** | | | | | | | |
| **Obtención de los requisitos del sistema** | | | | | | | |
| **Desarrollo e implementación (Fase 1)** | | | | | | | |
| **UAT fase 1** | | | | | | | |
| **Fase 1 "Go Live"** | | | | | | | |
| **Fase Desarrollo e implementación (Fase 2)** | | | | | | | |
| **UAT fase 2** | | | | | | | |
| **Fase 2 "Go Live"** | | | | | | | |
| **System Training** | | | | | | | |
| **Garantía del producto** | | | | | | | |

Tabla 1: Planificación del proyecto

# 4

# Análisis y diseño de la solución

La visión del equipo no es tan solo de la de conseguir los satisfacer los requisitos de los usuarios, sino también desarrollar un sistema de alto rendimiento, seguro de usar y que además permita expansiones y mejoras futuras. En los siguientes puntos se detallara el diseño del producto y como el equipo ha alcanzado los objetivos marcados.

## 4.1 Diseño

El diseño del sistema se divide en 2 partes. La primera de ellas es donde se encuentra el rastreador o *crawler* web, esta área está expuesta a internet ya que requiere obtener información de los múltiples sitios webs. La segunda parte del sistema se encuentra en la nube del gobierno de Singapur, y esta es solo accesible por los miembros del gobierno.

Ambas partes del sistema requieren una comunicación casi continua, ya que el rastreador web tiene que enviar la información para que esta sea analizada y el sistema enviará la lista de sitios web de los que necesita obtener información. Esta comunicación será posible mediante servicios web usando el protocolo HTTPS.

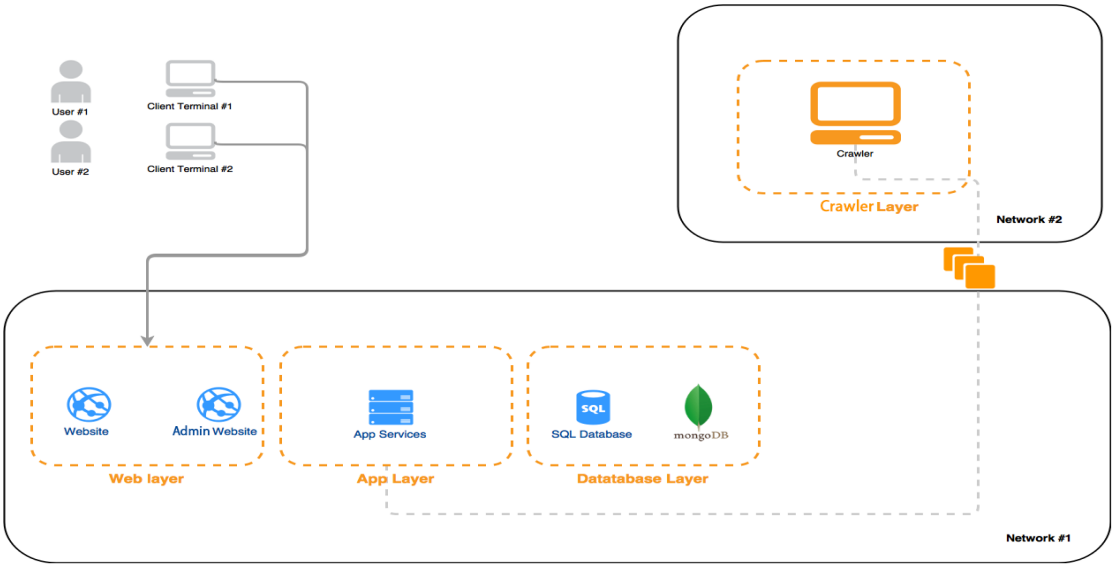



Figura 16: Diseño del sistema

Una vez el rastreador web ha obtenido una serie de documentos de internet estos son ordenados de forma cronológica y almacenados de manera temporal en una base de datos MS SQL a espera de que el sistema pida nuevos documentos.

Cuando el sistema pide nuevos documentos estos serán descargados desde el rastreador y puestos en una nueva cola ha espera de ser analizados. Previamente a empezar el proceso de analizar un documento, el documento es revisado para

asegurarse que no contiene publicidad. Una vez confirmado que el documento tiene un contenido relevante, se procederá a analizar el documento, este proceso se divide en 4 etapas, la primera de ellas es la de identificación del tema del documento, esto proceso se basa en las diferentes palabras claves que se encuentran en el documento. Basándose en el tema del documento este será categorizado basándose en los criterios que las diferentes organizaciones han establecido. La tercera etapa del proceso requiere extraer las diferentes entidades del documento, es decir, el documento identificará aquellos nombres, compañías o lugares que el documento menciona. La última etapa de este proceso generará un valor que indica el sentimiento del documento, es decir, positivo o negativo.

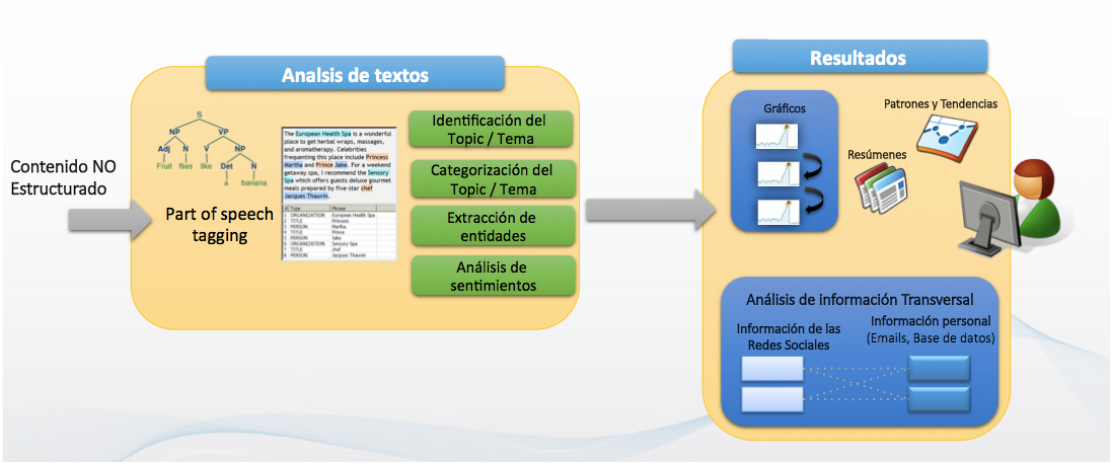



Figura 17: Proceso de análisis de contenidos no estructurados

Una vez los documentos son analizados estos son almacenados en una cola temporalmente a la espera de poder ser indexados por el sistema. Cuando el documento está preparado para ser indexado este se introducirá primeramente en MongoDB y después algunos de sus metadatos se insertaran en el índice de lucene.

Con este último paso se completa el proceso de cómo una página web es descargada, analizada e indexada en el sistema, quedando disponible para la consulta por parte de los usuarios a través de los servicios webs que el sistema dispone.

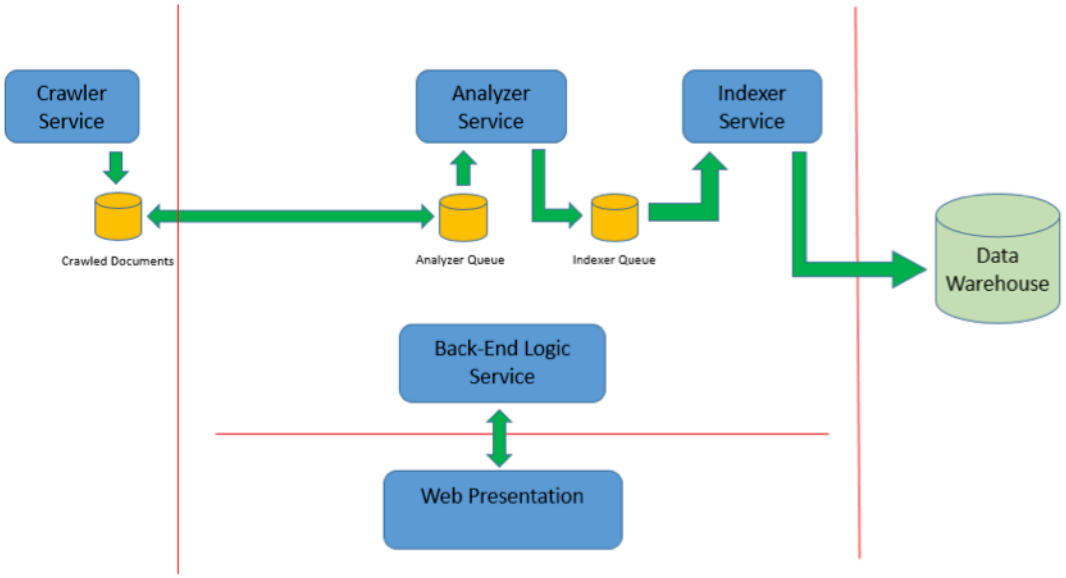



Figura 18: Visión general del sistema

## 4.2 Rendimiento

El rendimiento del sistema ha sido uno de los desafíos más grandes del proyecto. Scrum aconseja que los retos más importantes del proyecto se deben afrontar en las primeras iteraciones.

Siguiendo este consejo el equipo decidió afrontar en la 3 iteración el proceso de analizar e indexar documentos en el sistema. Una vez terminada la iteración, pudimos observar que este proceso acarreaba un gran consumo de recursos en el sistema y más tiempo del que estimábamos al principio del proyecto.

Después de las conclusiones previas debíamos mejorar este proceso. Debido a que el sistema utiliza 8 procesadores teníamos que hacer uso de todos ellos. Es por ello por lo que en la siguiente iteración se creó un analizador en cada uno de los procesadores. Al final de la misma se compararon los resultados con los de la previa iteración y se observó que aunque los resultados eran mejores, no eran suficientes.

En la siguiente iteración, decidimos analizar con mayor detalle cuáles eran las operaciones que requieran más tiempo durante el proceso de analizar e indexar nuevos documentos. Se identificó que la operación que más tiempo acarreaba era la obtención de la longitud y latitud de los diferentes lugares o calles que los documentos mencionaban.

Cada vez que el sistema identifica un lugar o una calle en el documento intentaba encontrar las coordenadas utilizando Google Maps, este era un proceso

muy tedioso. Por este motivo el equipo decidió que en lugar de buscar las coordenadas utilizando servicios web, el sistema debía buscar las coordenadas en una extensa base de datos interna.

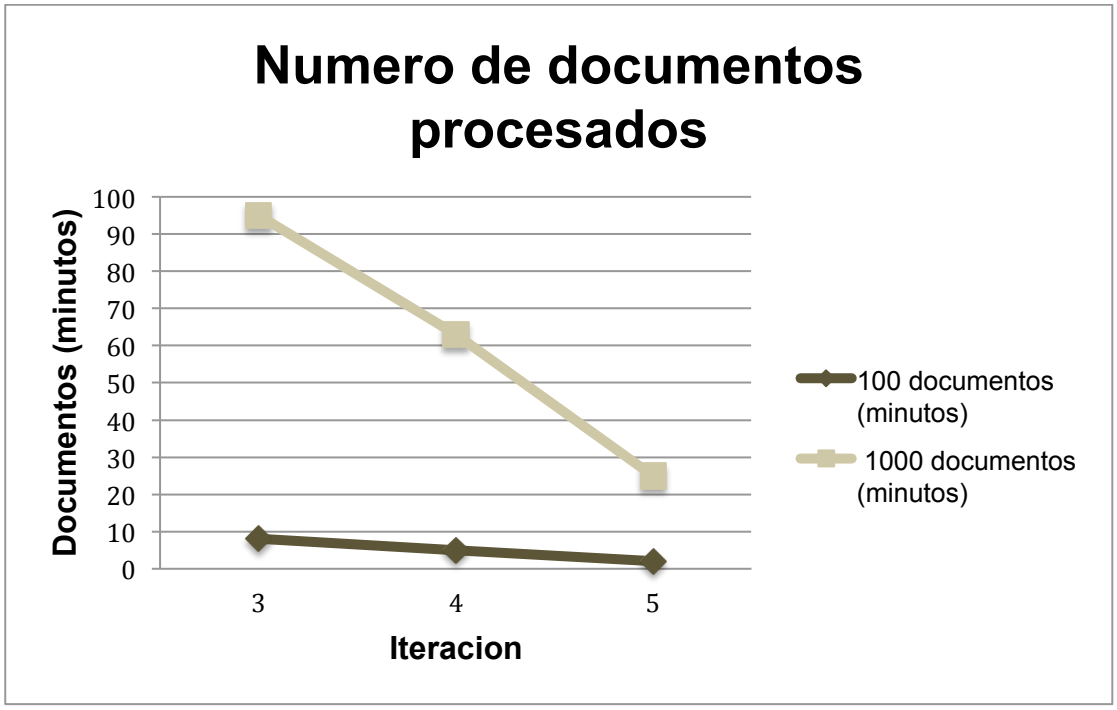



Figura 19: Numero de documentos procesados

Seguir el consejo de scrum nos ayudó positivamente, ya que si al final del proyecto hubiéramos descubierto que el sistema no tenía el rendimiento deseado, hubiera sido demasiado tarde y hubiera acarreado problemas mayores. De esta manera, pudimos atajar el problema a tiempo y conseguir el objetivo deseado.

## 4.3 Usabilidad

Desde el principio del proyecto el cliente ha mostrado su interés en que el proyecto no tan solo debe tener un diseño impecable, sino también debe proporcionar información de forma inteligente, persuasiva y con una agradable experiencia de uso.

Es por ello que el equipo decidió seguir los siguientes principios:

- Visibilidad del estado del sistema: utilizando barras de proceso y animaciones.

- Consistencia entre el sistema y los usuarios: El sistema debe seguir el mismo orden en el que los usuarios hacen las cosas.

- Consistencia y estándares: El sistema debía seguir el mismo estándar de diseño en todas las páginas.

- Prevención de errores: Debíamos intentar evitar que el usuario cometiera errores usando el sistema, especialmente durante las primeras veces que están utilizando el sistema, para ello incluimos diversos mensajes con ayudas y haciendo comprobaciones en tiempo real.

- Diseño estético y minimalista

- Ayuda al usuario a reconocer, diagnosticar y recuperarse de los errores: Los mensajes de error tienen que estar escritos en un lenguaje que el usuario pueda entender y deben siempre sugerir una solución o un camino de salida.

Siguiendo los previos principios se ha obtenido un sistema que ha resultado fácil de usar y entender.

## 4.4 Escalabilidad y extensibilidad

Las exigencias y requisitos requeridos del sistema van a seguir creciendo a lo largo del proceso de desarrollo y una vez terminado el mismo. Es por ello por lo que el sistema debe tener una infraestructura que permita crecer y adaptarse a las necesidades de los usuarios. Existen dos maneras de escalar una aplicación:

- **Escalado vertical**: implica añadir más recursos físicos al sistema actual para que pueda atender más solicitudes. Así, por ejemplo, se le añaden más procesadores o más memoria a un servidor. El escalado vertical presenta un límite claro y llega un punto en el que la aplicación necesita ser rediseñada para poder escalar más.

- **Escalado horizontal**: en este caso el escalado se consigue simplemente añadiendo más nodos al conjunto del sistema, por ejemplo, poniendo otra máquina en paralelo a funcionar.

Entre las dos propuestas para escalar nuestro sistema, la más recomendada es el escalado de tipo horizontal.

También se han aplicado algunas buenas prácticas para conseguir un mejor rendimiento de la aplicación. A continuación se muestra un listado de ellas

- Ajustes y configuración sobre SQL Server 2008R2

- Análisis de rendimiento de la aplicación y detección de cuellos de botella

- Optimización de los procesos con alta concurrencia

- Mejoras en la indexación y en las tareas de mantenimiento del servidor
- Optimización del consumo de CPU global
- Optimización en las operaciones masivas
- Optimizaciones de MongoDB

## 4.5 Seguridad

La información es el elemento principal a proteger y resguardar debido la confidencialidad de los datos con los que los usuarios trabajan.

Existen gran cantidad de métodos para asegurar una red informática, uno de los más usados es el de *demilitarized zone*[8] (DMZ) o zonas desmilitarizadas. En esta parte de la red es donde se colocaran aquellos servicios que son utilizados por otros usuarios externos a la red como por ejemplo servicios web, servidores web, FTP o Voice over IP (VoIP) servers. En nuestro proyecto la capa conocida como DMZ es la llamada "Web layer".

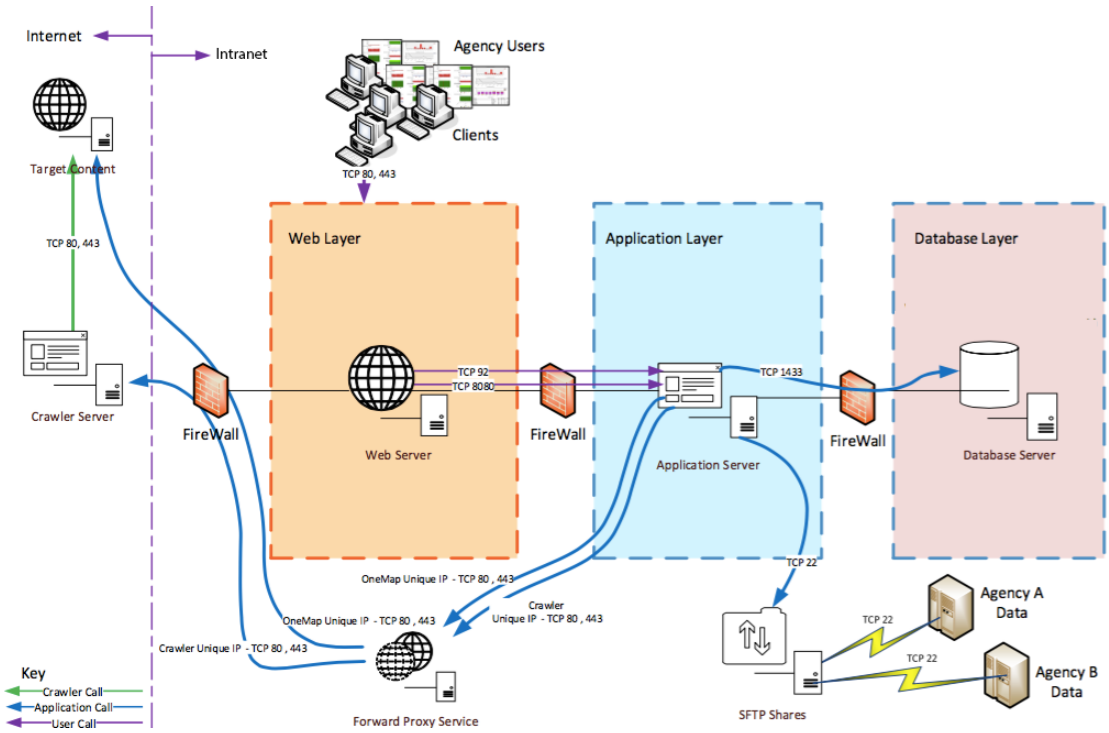



Figura 20: Diagrama de red

Como se observa en la figura número 20, el desarrollo de esta propuesta ha seguido una arquitectura de 3 capas. Esto permite separar la capa de presentación, la capa de negocio y los datos. En ocasiones esta separación puede

ser física, como se muestra en la figura 12. Es decir, podemos tener en distintos ordenadores cada una de las capas, con lo cual obtenemos una aplicación distribuida físicamente.

El uso de esta arquitectura ha permitido obtener los siguientes beneficios respecto a la seguridad:

- Limita y controla el acceso a los elementos de red, servicios y aplicaciones.

- Garantía de la procedencia de la información y de que esta solo es accesible por las entidades, sistemas o personas autorizadas.

- La información no es modificada o corrompida de manera alguna, desde su transmisión hasta su recepción.

- La información que fluye en la red se mantiene privada.

# 5

# Implementación y Despliegue

En este capítulo se explica la implementación y despliegue de la propuesta, así como el impacto del producto, estadísticas del uso y finalmente se expone un caso de éxito de utilización de Gov TA.

## 5.1. Implementación de la propuesta

La implementación del producto está dividida en dos partes diferenciadas. La primera de ellas es *front-end*, es decir la interfaz gráfica de la aplicación web, está basada principalmente en la tecnología JavaScript y ASP.NET. La segunda parte del sistema es el *back-end* y está desarrollada en principalmente en .NET.

### 5.1.1. Front-end

Los proyectos MVC tienen una estructura no muy compleja en la que prima la buena organización de los ficheros para facilitar el desarrollo. A continuación se describe la estructura.

- App_Data: Contendrá datos de la aplicación como archivos SQL
- App_Start: Contendrá archivos de configuración de la aplicación (entre ellos el RouteConfig)
- Content: Contendrá archivos estáticos como CSS
- Controllers: Contendrá los Controladores de nuestra Aplicación
- Filters: Contendrá clases para el comportamiento antes y después de las acciones de los controladores
- Images: Contendrá imágenes que utiliza nuestra aplicación.
- Models: Contendrá los Modelos de nuestra aplicación.
- Scripts: Contendrá archivos de Script como JavaScript
- Views: Contendrá las Vistas de nuestra aplicación.

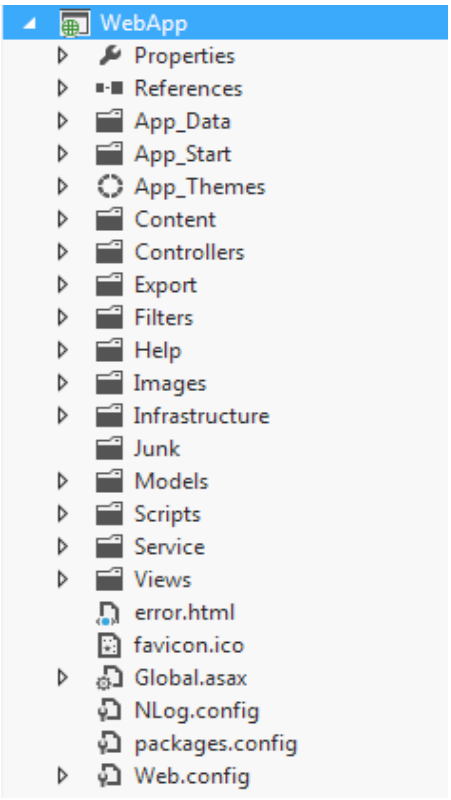



Figura 21: Estructura de nuestro proyecto

Sin embargo, la tecnología más importante en esta parte del proyecto ha sido JavaScript. Este lenguaje de programación ha sido ampliamente subestimado e infravalorado. Originalmente fue desarrollado primero para un solo navegador (Netscape Navigator), pero pronto al ver su potencial, fue integrado a la mayoría de navegadores web por sus fabricantes.

En los últimos 5 años JavaScript se ha convertido en el lenguaje de referencia en desarrollo web. Esta tecnología se ha convertido en el motor de las aplicaciones más conocidas en el ámbito de Internet como Google, Facebook o Twitter.

En JavaScript, la orientación a objetos es conceptualmente muy diferente a la orientación a objetos de C#, Java y otros lenguajes. Por otro lado, en JavaScript los objetos son dinámicos, es decir, pueden cambiar estructuralmente durante la ejecución del programa (En los lenguajes estáticos como C# o Java, no se pueden modificar las clases en tiempo de ejecución. Es decir, no se pueden añadir o quitar atributos o métodos de los objetos de una clase durante la ejecución del programa) Al principio parece que estos dos detalles no son importantes y que conceptualmente C# y JavaScript son bastante parecidos, pero luego te das cuenta de que esas pequeñas diferencias superficiales afectan de forma importante a la forma de programar con un lenguaje u otro.

Como la fundación Mozilla explica “JavaScript es un lenguaje basado en objetos que en lugar de estar basado en clases, se basa en prototipos. Debido a

esta diferencia, puede resultar menos evidente que JavaScript te permite crear jerarquías de objetos y herencia de propiedades y de sus valores."[9].
Aunque resulte menos evidente, JavaScript permite definir tipos de objetos y crear instancias de objetos de diferentes maneras:

- La más común es crear una función con métodos definidos internamente.

```
function Apple (type) {
       this.type = type;
       this.color = "red";
       this.getInfo = function() {
              return this.color + ' ' + this.type + ' apple';
            };
 }
```

Figura 22: Funciones en JavaScript (1)

Un inconveniente de este método es que la función getInfo() se vuelve a crear cada vez que se crea un nuevo objeto.

- Otra forma de crear una clase es usando protoype.

```
function Apple (type) {
     this.type = type;
     this.color = "red";
}

Apple.prototype.getInfo = function() {
     return this.color + ' ' + this.type + ' apple';
};
```

Figura 23: Funciones en JavaScript (2)

- Usando objetos. Este ha sido el estándar más habitual escogido para nuestro proyecto.

```
	var apple = {
		type: "macintosh",
		color: "red",
		getInfo: function () {
			return this.color + ' ' + this.type + '
apple';
		}
	}
```

Figura 24: Funciones en JavaScript (3)

En este caso no es necesario (y no posible) crear una instancia de la clase, ya previamente es creada. Es decir, únicamente se tiene que hacer uso del objeto.

```
	apple.color = "reddish";
	alert(apple.getInfo());
```

Figura 25: Usando objetos en JavaScript

En lenguajes como C# o Java este objeto es conocido como *singleton*. Es decir, que solo se puede tener una única instancia de esta clase en cualquier momento.

Haciendo uso de JavaScript hemos creado una larga lista de objetos, con propiedades y funciones que nos han ayudado a completar los requisitos de los usuarios en la interfaz gráfica.

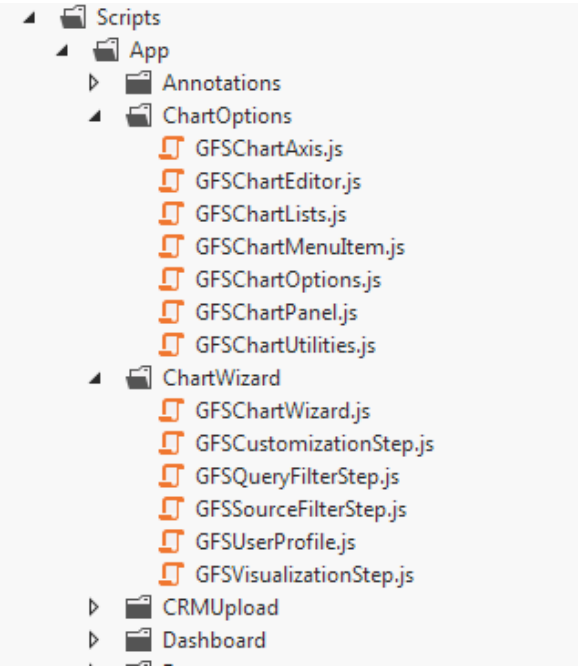



Figura 26: Lista de objetos creados en JavaScript

## 5.1.2. Back-end

El *back-end* de nuestra aplicación esta dividió en 3 partes todas ellas están desarrolladas en C#.

- El analyzer es el encargado de analizar y extraer datos de interés de los documentos.

- Inderxer es el encargado de insertar en lucene y en MongoDB los documentos, permitiendo que estos sean accesibles a los usuarios.

- Back-End Logic es un conjunto de funciones que permiten a los usuarios realizar múltiples acciones en el sistema.

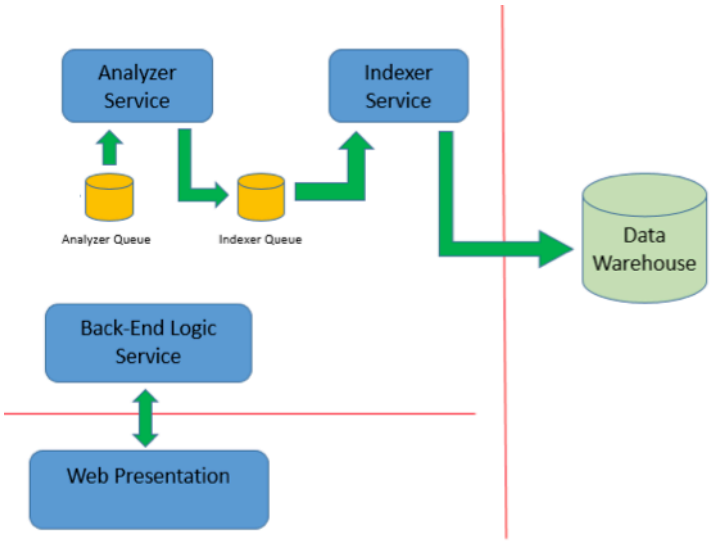



Figura 27: Back-End

Sin embargo, todas las anteriores partes tienen dos cosas en común, la primera es que necesitan de una continua comunicación para poder realizar sus acciones y la segunda es que algunas de estas funciones debían de poder ser accesibles desde la capa DMZ. Basado en los anteriores requisitos entendimos que todas las partes debían ser accesibles a través de servicios web.

ServiceStack es un *framework opensource* que se presenta como alternativa a WCF para la creación de servicios web en .NET sobre HTTP, además provee una forma rápida y limpia de desarrollar servicios web orientados a la utilización. Cada servicio en ServiceStack está formado por tres partes:

- Un DTO de petición
- La implementación del servicio
- Un DTO de respuesta

```
namespace ServiceStack.Web
{
  [Route("/saludo")]
  [Route("/saludo/{Nombre}")]
  public class Saludo
 {
        public string Nombre {get;set;}
 }
}
```

Figura 28: Creando la petición del servicio Saludo

```
namespace ServiceStack.Web
{
  public class SaludoResponse
 {
        public string Respuesta {get;set;}
 }
}
```

Figura 29: Creando la respuesta del servicio Saludo

```
namespace ServiceStack.Web
{
  public class SaludoService : IService
 {
        public object Any(Saludo request)
        {
              return new SaludoResponse() {
                    Respuesta = "Hola"+request.Nombre,
              };
        }
 }
}
```

Figura 30: Implementación del servicio Saludo

Al utilizar "Any" como nombre de método, estamos indicando que este método se encarga de gestionar cualquier tipo de llamada HTTP (Get, Post, Put, Delete). Si queremos utilizar la convención de nombres para cada tipo de llamada HTTP, entonces podemos crear métodos acordes llamados Get, Post, Put y Delete.

Utilizando ServiceStack hemos creado multitud de servicios web que permiten la comunicación de las diferentes partes del sistema.

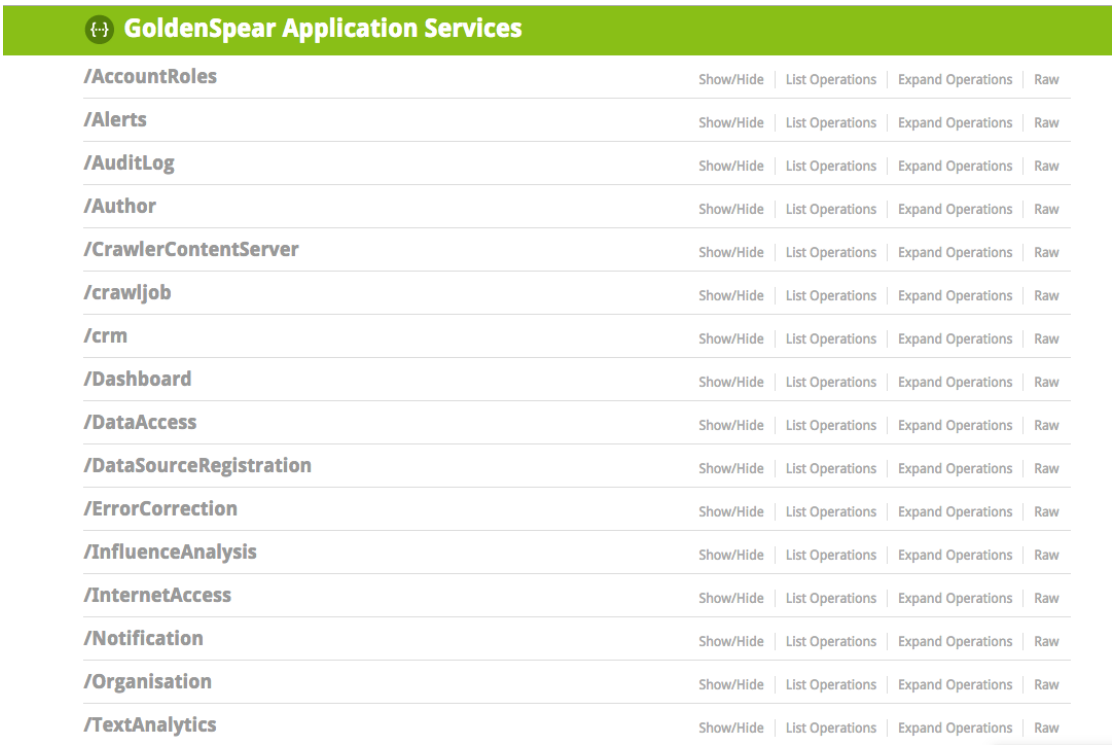



Figura 31: Gov TA - API

## 5.2. Estadísticas de uso e Impacto del producto

Una de las preguntas más habituales una vez terminado un proyecto de software es saber cómo los usuarios han recibido el sistema y conocer si están haciendo uso del mismo. Para resolver estas cuestiones decidimos hacer uso del módulo auditoría del sistema.

Este módulo registra diferentes actividades y eventos que ocurren en el sistema para fines de auditoría. Algunos de los que resultan más interesantes para conocer si los usuarios están utilizando el sistema son los siguientes

- Inicio de sesión en el sistema
- Actualizaciones de contraseñas o roles
- Registro de actividades que los usuarios realizan tales como búsquedas, uso técnicas de análisis de textos o alertas
- Creación o Modificación de gráficos
- Usuarios activos del sistema (Hace menos de 30 días del último inicio de sesión en el sistema)

- Archivos almacenados en el sistema

Basándonos en los eventos anteriores mencionados, decidimos conocer al final de cada mes cual era el número de usuarios que habían empezado utilizado el sistema en las diferentes organizaciones. A continuación, se muestra un gráfico con los resultados del mismo.

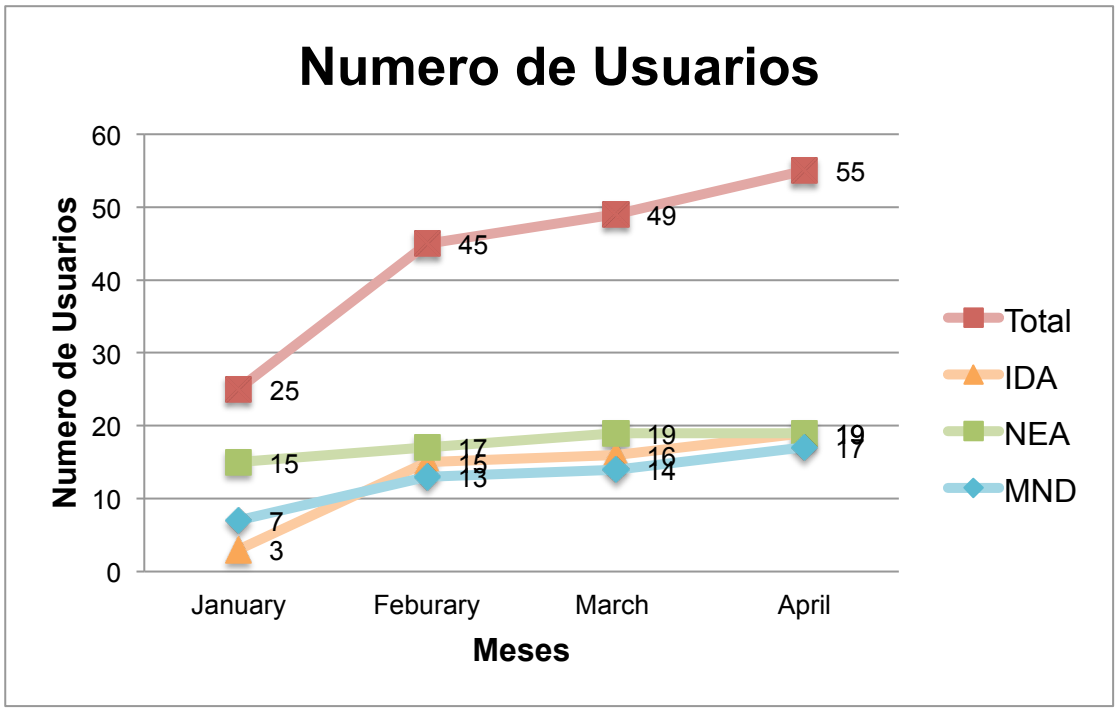



Figura 32: Número de usuarios del sistema

En el mes de enero se comunicó por correo electrónico a los posibles usuarios del sistema, que oficialmente este estaba accesible y se les proporcionó un usuario y contraseña a cada uno de ellos.

En el gráfico anterior se puede observar que la organización NEA presentó un mayor número de usuarios que habían utilizado el sistema en comparación con las otras dos. Este dato es debido a que en la segunda semana del mes se realizó una sesión práctica con diferentes usuarios de la organización, lo que provocó que los usuarios empezaran a hacer un mayor uso de la herramienta como se muestra a continuación.

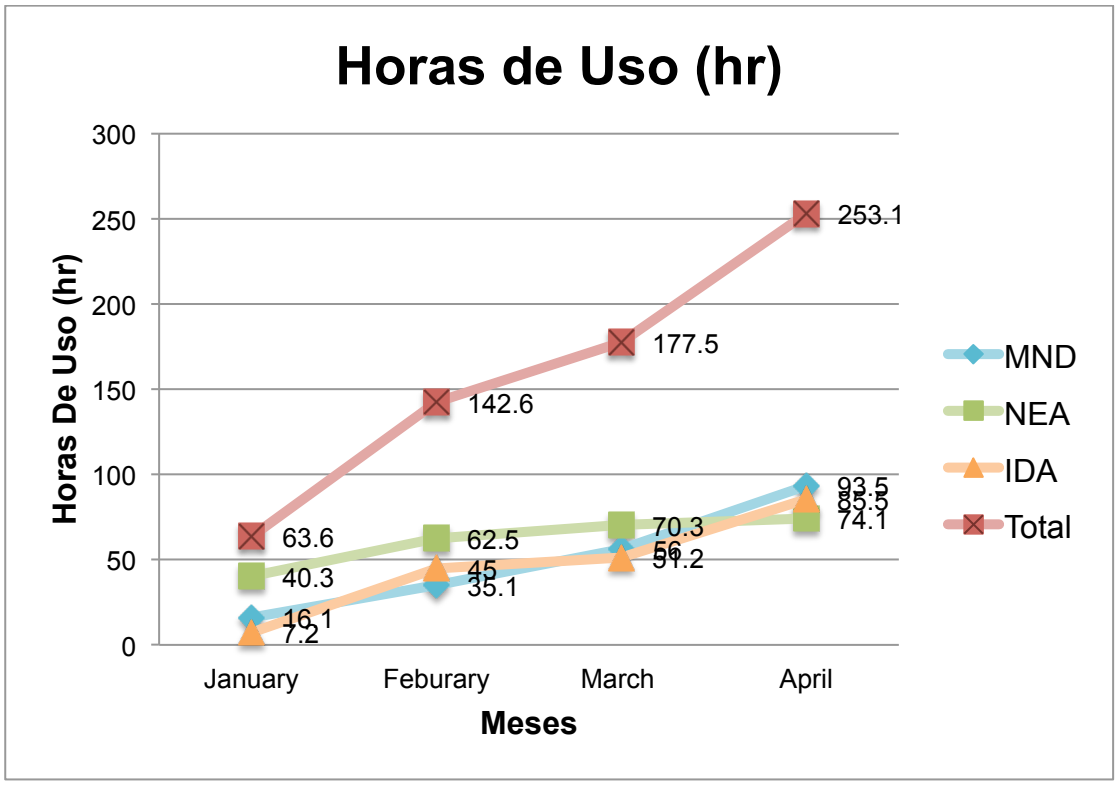



Figura 33: Horas de uso del sistema

## 5.2.1. Sitios web de interés

A medida que los usuarios están utilizando el sistema se les está preguntando cuales son aquellos sitios web de interés que les gustaría incluir en el sistema.

El resultado de esta cuestión fue que la gran mayoría de los usuarios quieren monitorizar diferentes páginas o perfiles en Facebook. Esta red social permite de una manera fácil compartir noticias, eventos y a la vez escuchar las diferentes opiniones de otros usuarios.

Los usuarios también han mostrado interés por otras plataformas de blogs como Blogspot, WordPress o Tumblr. A diferencia de Facebook, en estas plataformas los usuarios tienden a redactar artículos mucho más completos de los cuales la herramienta puede extraer más información, pero por el contrario el número de comentarios u opiniones sobre los artículos es mucho menor.

Otra fuente de información para los usuarios sigue siendo los foros. A pesar de que en los últimos años estos tienen un menor impacto debido a la aparición de otros medios, para los usuarios del sistema estos siguen siendo un referente.

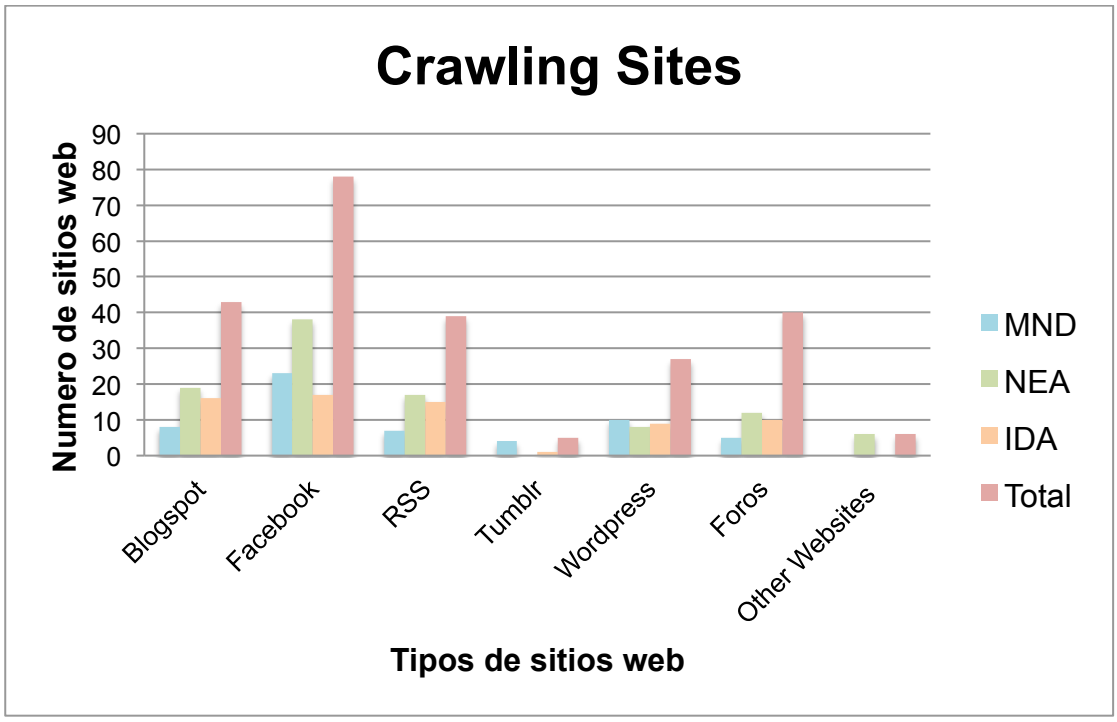



Figura 34: Fuentes de información del sistema

## 5.3. Caso de éxito – Wireless@SG

Wireless@SG es parte del programa Intelligent Nation 2015 del gobierno de Singapur para fomentar el acceso a internet de todos los ciudadanos, permitiéndoles acceso gratuito en lugares públicos como cafés, medios de transporte, restaurantes, centros comerciales, bibliotecas y otros lugares públicos.

Este programa empezó en 2006 y con los años el programa ha introducido varias mejoras para mejorar la experiencia del usuario. El programa también ha evolucionado para animar a los operadores a desarrollar servicios innovadores que aprovechan la red Wireless@SG. Estos servicios permiten generar ingresos para los operadores y permiten un modelo auto sostenible para apoyar la provisión continua de conexión Wi-Fi gratuita para el público.


Figura 35: Puntos Wi-Fi

La organización IDA quiso hacer unos análisis del estado del programa y como los ciudadanos estaban aceptando el programa. Para ello necesitaban hacer uso de una herramienta de software que les permitiera analizar datos en tiempo real de diferentes fuentes de información como redes sociales, foros, formularios electrónicos o correos electrónicos.

IDA escogió realizar el análisis a través de nuestro sistema Gov TA, ya que este sistema les permite hacer un análisis usando datos internos de su CRM y de redes sociales como foros, blogs y redes sociales.

Los resultados de este análisis les permitieron identificar los puntos de interés.

- Cuales son los problemas más habituales de los usuarios utilizando Wireless@SG.
- Datos demográficos de quienes son los que más utilizan la red Wi-Fi
- Cuales son las áreas con mayor numero de incidencias
- Dispositivos más habituales
- Opinión general de la población sobre el estado del programa y las expectativas futuras de los ciudadanos sobre el mismo.

-

# 6

# Conclusiones

En este apartado quedan reflejados los objetivos alcanzados por el proyecto, así como los requisitos que se han cumplido, las impresiones personales que he obtenido con la realización de este proyecto y aquellas futuras mejoras que podemos aplicar al proyecto.

## 6.1. Conclusiones

El proyecto elaborado ha cumplido el principal objetivo marcado, proporcionar a los usuarios de las diferentes agencias gubernamentales del gobierno de Singapur de un sistema que les permita capturar, almacenar, procesar y analizar contenidos no estructurados procedentes de diferentes canales, tales como redes sociales, foros de debate, webs corporativas o correos electrónico. Alcanzar este objetivo ha sido posible gracias al gran esfuerzo y dedicación de todo el equipo.

Realizar una solución a un problema real, dentro de una empresa, ha cumplido todas las expectativas que tenía puestas en el proyecto, ya que me ha permitido estar involucrado en todas las fases del desarrollo de un nuevo producto software, además de poder contribuir mejorando la variedad de productos de software de S2T y permitir la expansión de la empresa en un nuevo y excitante mercado como es el *big data*. Es por ello valoro de forma muy positiva la experiencia personal alcanzada en la realización de este proyecto.

He podido poner en práctica alguna de los muchos conocimientos que he aprendido en la Máster en Ingeniería del Software, Métodos Formales y Sistemas de la Información, a la vez que he ampliado los mismos a través de la experiencia que mis compañeros me han transmitido y de las situaciones en las que nos hemos encontrado.

Asimismo durante todo el proceso hemos trabajado utilizando una metodología de desarrollo ágil, que con la ayuda de esta y de otros factores ha llevado al éxito del proyecto. Quizás esta no sea una metodología apropiada para todos los equipos de desarrollo, pero en nuestro caso ha demostrado ser conveniente y ha proporcionado libertad para que el grupo estableciera sus propias recomendaciones y prácticas.

La elaboración del proyecto, también me ha servido para conocer mejor mis capacidades y poder medir con más éxito el tiempo requerido en la planificación de las tareas. Así como, para ser más metódico y ordenado en el desarrollo. Pero además también me ha permitido ser consciente de que aun me falta mucho por aprender pero que tengo una buena base de conocimiento.

## 6.2. Líneas de trabajo futuro

Si bien es cierto que se han logrado alcanzar gran parte de los objetivos marcados al inicio del proyecto hay algunos aspectos del mismo que son

mejorables; pero además de esto, también hemos encontrado nuevas líneas de trabajo que podrían aumentar la potencia y la utilidad del mismo.

A continuación, mostraremos los posibles trabajos futuros relacionados con la mejora de funcionalidades de la aplicación:

- **Notificaciones *push***: Después de hablar con los usuarios y conocer como utilizan el sistema de alertas día a día, nos dimos cuentas que hay soluciones más eficientes de enviar notificaciones a los usuarios. Por este motivo nuestra propuesta es llevar las notificaciones al navegador en lugar de enviar un correo electrónico.

  Una notificación *push* es aquella en la que una aplicación en el lado del servidor (una aplicación web externa a tu dispositivo) envía un mensaje al cliente (teléfono o navegador) para avisarle de que algo ha sucedido, la peculiaridad de este tipo de comunicación es que a diferencia de lo que estamos acostumbrados esta es iniciada por el servidor no por nosotros.

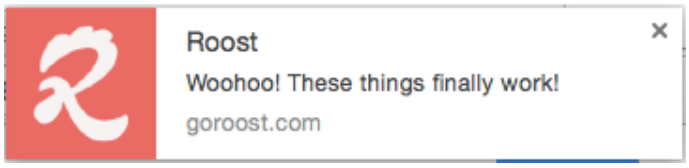
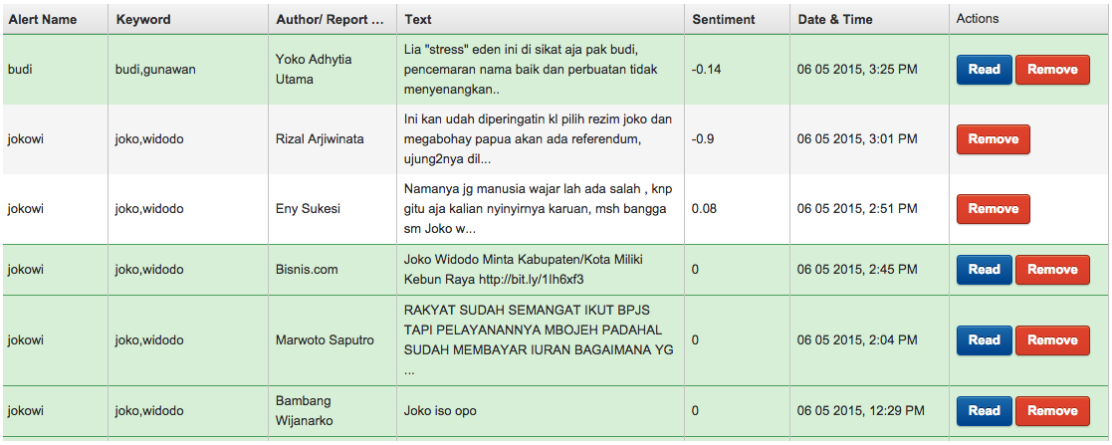



Figura 36: Notificaciones *push*

  Una vez el usuario ha recibido la notificación esta debería ser reflejada en el sistema web, en caso de que este decida consultar su contenido.



| Alert Name | Keyword | Author/ Report ... | Text | Sentiment | Date & Time | Actions |
|---|---|---|---|---|---|---|
| budi | budi,gunawan | Yoko Adhytia Utama | Lia "stress" eden ini di sikat aja pak budi, pencemaran nama baik dan perbuatan tidak menyenangkan.. | -0.14 | 06 05 2015, 3:25 PM | Read Remove |
| jokowi | joko,widodo | Rizal Arjiwinata | Ini kan udah diperingatin kl pilih rezim joko dan megabohay papua akan ada referendum, ujung2nya dil... | -0.9 | 06 05 2015, 3:01 PM | Remove |
| jokowi | joko,widodo | Eny Sukesi | Namanya jg manusia wajar lah ada salah , knp gitu aja kalian nyinyirnya karuan, msh bangga sm Joko w... | 0.08 | 06 05 2015, 2:51 PM | Remove |
| jokowi | joko,widodo | Bisnis.com | Joko Widodo Minta Kabupaten/Kota Miliki Kebun Raya http://bit.ly/1Ih6xf3 | 0 | 06 05 2015, 2:45 PM | Read Remove |
| jokowi | joko,widodo | Marwoto Saputro | RAKYAT SUDAH SEMANGAT IKUT BPJS TAPI PELAYANANNYA MBOJEH PADAHAL SUDAH MEMBAYAR IURAN BAGAIMANA YG ... | 0 | 06 05 2015, 2:04 PM | Read Remove |
| jokowi | joko,widodo | Bambang Wijanarko | Joko iso opo | 0 | 06 05 2015, 12:29 PM | Read Remove |

Figura 37: Alertas

- **Mapa**: Desde un principio este proyecto ha estado enfocado para dar un mejor servicio a los funcionarios del gobierno de Singapur, es por ello por

lo que decidimos utilizar el servicio de mapas de OneMap, que aporta una gran variedad de funciones en Singapur.

Sin embargo, esta opción es únicamente valida para empresas u organizaciones dentro Singapur, debemos mejorar esta funcionalidad si queremos exportar el producto a otros mercados. Con esta idea en mente hemos decidido utilizar MapServer for Windows[14], el cual es un potente servidor de mapas desarrollado por la Universidad de Minnesota.

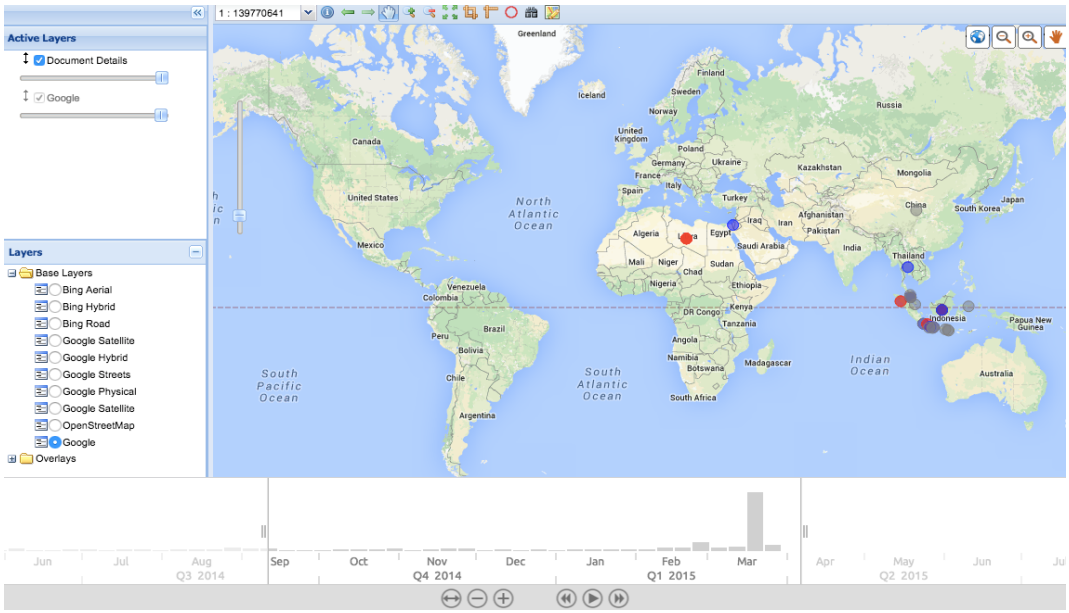



Figura 38: MapServer for Windows

A medida que el proyecto ha ido tomando forma, las organizaciones se han dado cuenta del potencial que el producto les puede otorgar y han decido añadir nuevas funcionalidades. Un sencillo ejemplo es la organización NEA, la cual ha decido usar la capacidad de analizar textos y la clasificación automática de textos para predecir a que departamentos deben redirigir las reclamaciones que llegan a su correo electrónico día a día, actualmente este proceso es manual, en el cual es un oficial el que determina a que departamento debe redirigir las quejas de los usuarios.

Por último, también me ha resultado sorprendente que otras organizaciones ajenas al proyecto han mostrado interesadas en integrar el producto en su organización. A razón de este hecho hemos estado realizando diversas demostraciones del producto en los últimos meses y la organización más interesada ha sido el Cuerpo Nacional de Policía (SPF), la cual mostró un gran interés debido a la funcionalidad que permite explorar las relaciones entre la información política y la opinión de los ciudadanos. Su objetivo es explorar las relaciones entre informes policiales y otras fuentes externas de información.

[14] MapServer for Windows http://www.maptools.org/ms4w/

# Bibliografía


[1] Rao, R. Gupta, “Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm”, International Journal of Computer Science And Technology, pp. 489-493, Mar. 2012

[2] Microsoft. Web Service Definition. 2012. http://msdn.microsoft.com/enus/library/ms950421

[3] K. Gottschalk, S. Graham, H. Kreger, and J. Snell. "Introduction to Web services architecture" IBM Systems Journal, no. ISSN: 0018-8670, 2002.

[4] A. Benharref, M. A. Serhani, S. Bouktif, and J.Bentahar. "A managerial community of Web Services for management of communities of Web Services," in New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference on., 2010.

[5] W3C. Web Services Glossary. 2004 http://www.w3.org/TR/ws-gloss/.

[6] Beck, K. “Extreme Programming Explained. Embrace Change”, 1999.

[7] Robert J. Shimonski, Will Schmied, Dr. Thomas W. Shinder, Victor Chang, Drew Simonis, “Building DMZs for Enterprise Networks”, 2003.

# A.1

## Manual del usuario

## 1.1. Información general

El objetivo de Gov TA es dotar a los usuarios de las diferentes agencias gubernamentales del gobierno de Singapur de un sistema que les permita capturar, almacenar, procesar y analizar contenidos no estructurados procedentes de diferentes canales, tales como redes sociales, foros de debate, webs corporativas o correos electrónico. Además de la ganancia de productividad de los usuarios en el análisis de los datos, GOV TA permite a los usuarios descubrir patrones ocultos, las tendencias de la opinión pública, referencias ocultas en los textos y revelar asuntos de interés.

## 1.2. Primeros pasos

Para iniciar sesión en la aplicación web, abra el enlace de Gov TA utilizando Google Chrome o IE (versión 9 o superior). Introduzca el nombre de usuario y contraseña y haga clic en el botón "Login"

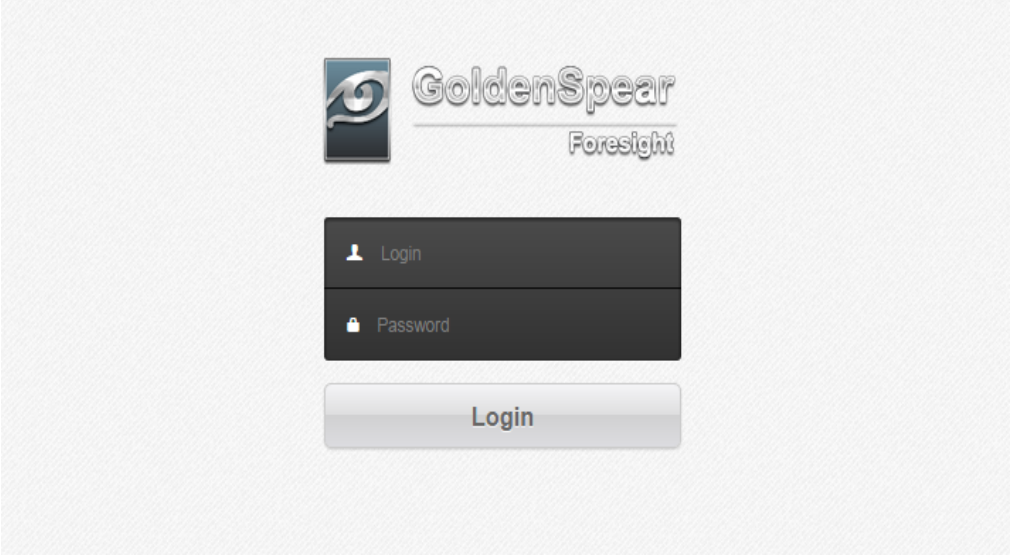



Figura 39: Página de *login* de Gov TA

Una vez validado el usuario, el sistema mostrará la página de inicio.

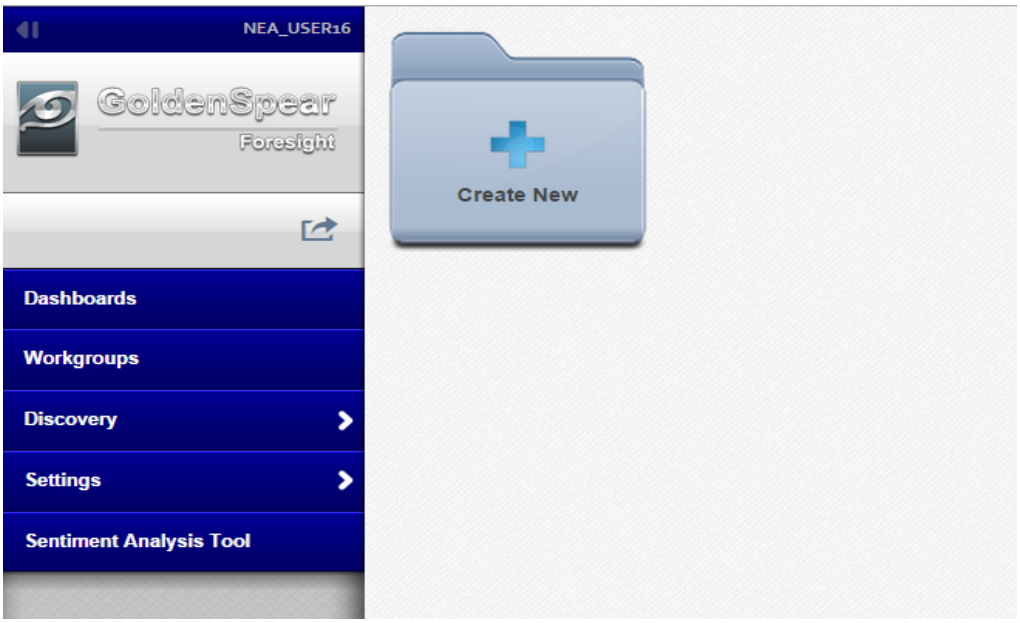



Figura 40: Página de inicio

## 1.3. Dashboards

El sistema cuenta con diferentes tipos de visualizaciones que permitan a los usuarios revisar y compartir los conocimientos analíticos del sistema. A continuación, vamos a mostrar un ejemplo de ellas

Haga clic en 'Dashboard' en la barra lateral izquierda y después haga clic en el icono ‘Create new’. El sistema le pedirá al usuario que introduzca el nombre.

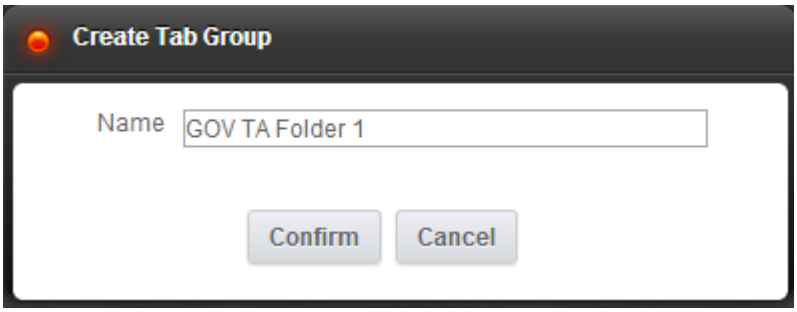



Figura 41: Nombre del *dashboard*

Introduzca el nombre para el nuevo grupo de pestañas y seleccione "Confirmar". Una vez hecho esto, el usuario será dirigido al Grup Tab.

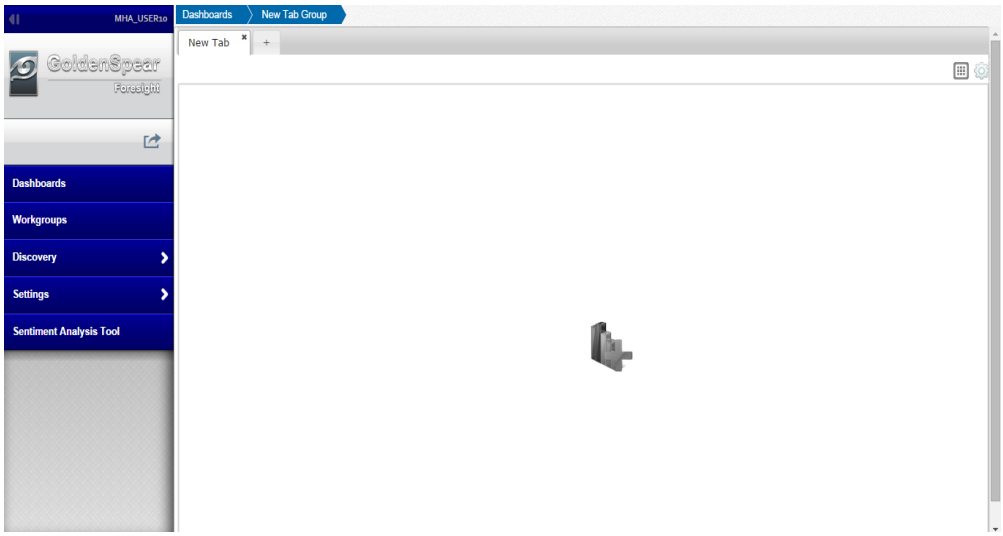



Figura 42: Crear gráfico

Una vez el usuario se encuentra en esta página deberá hacer doble clic en el icono para crear el gráfico. En la siguiente imagen el usuario deberá seleccionar aquellos *topics* o temas de los cuales está interesado y hacer clic en "Next".

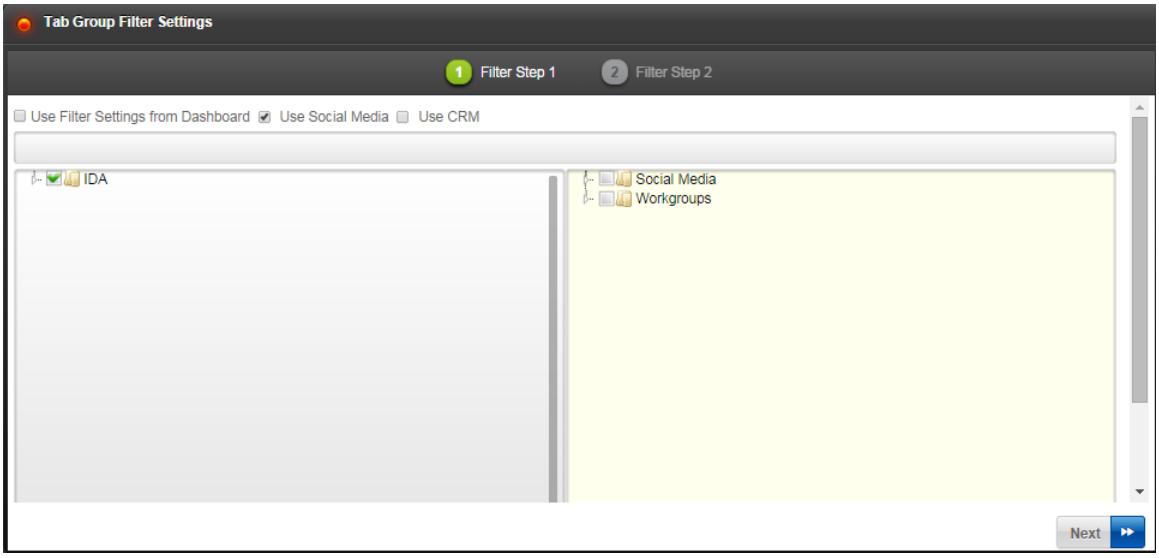



Figura 43: Seleccionar el *topic* o tema

El usuario puede seleccionar entre diferentes visualizaciones. A continuación se detallan las mismas

a) Área

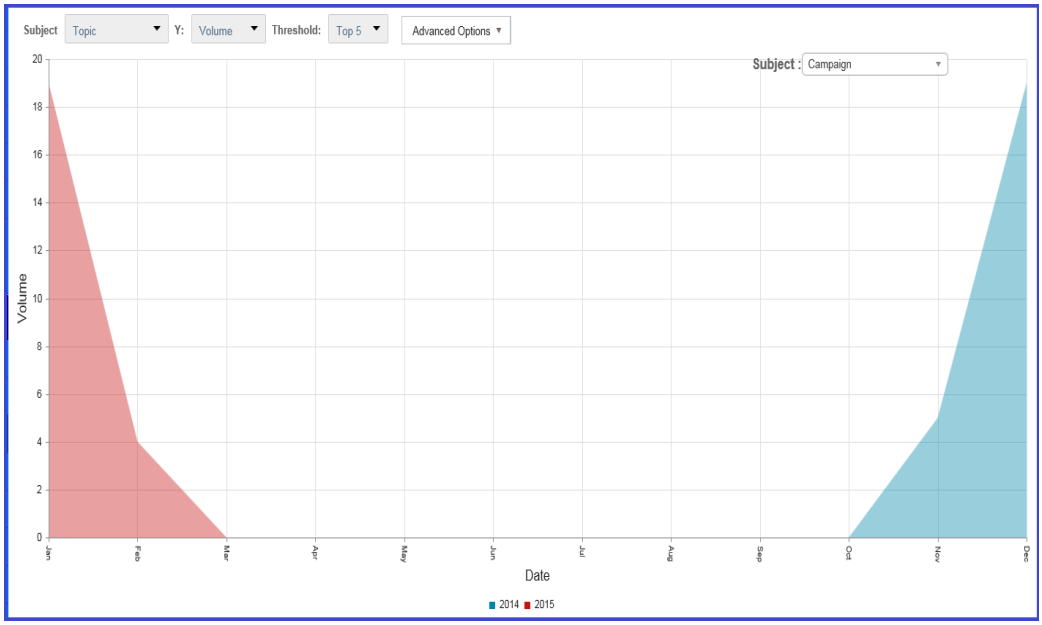



Figura 44: Área

b) Gráfico de burbujas

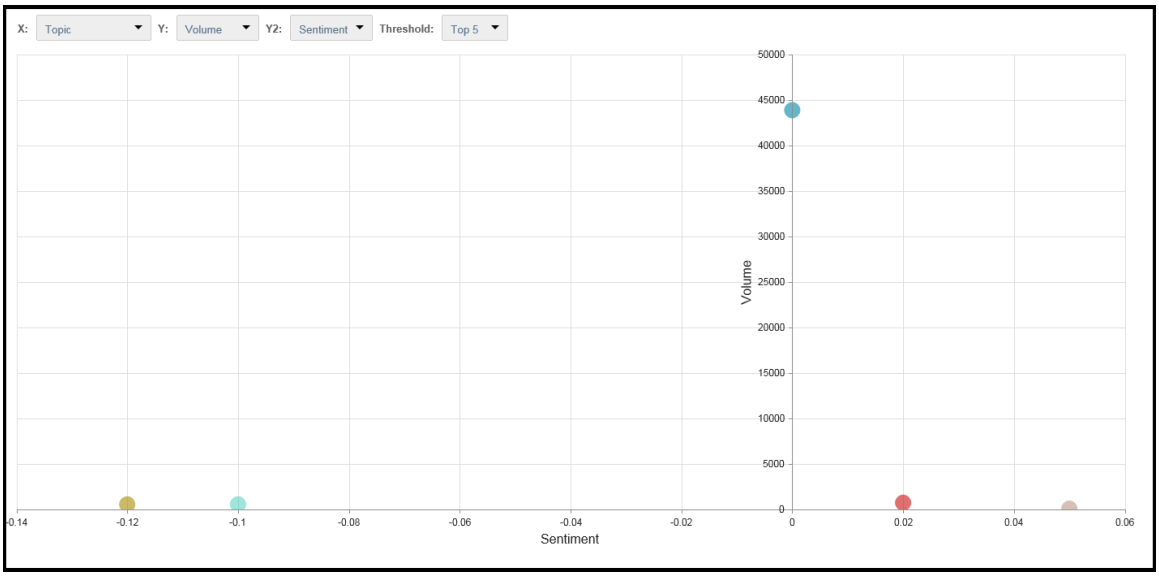



Figura 45: Gráfico de burbujas

## c) Donut

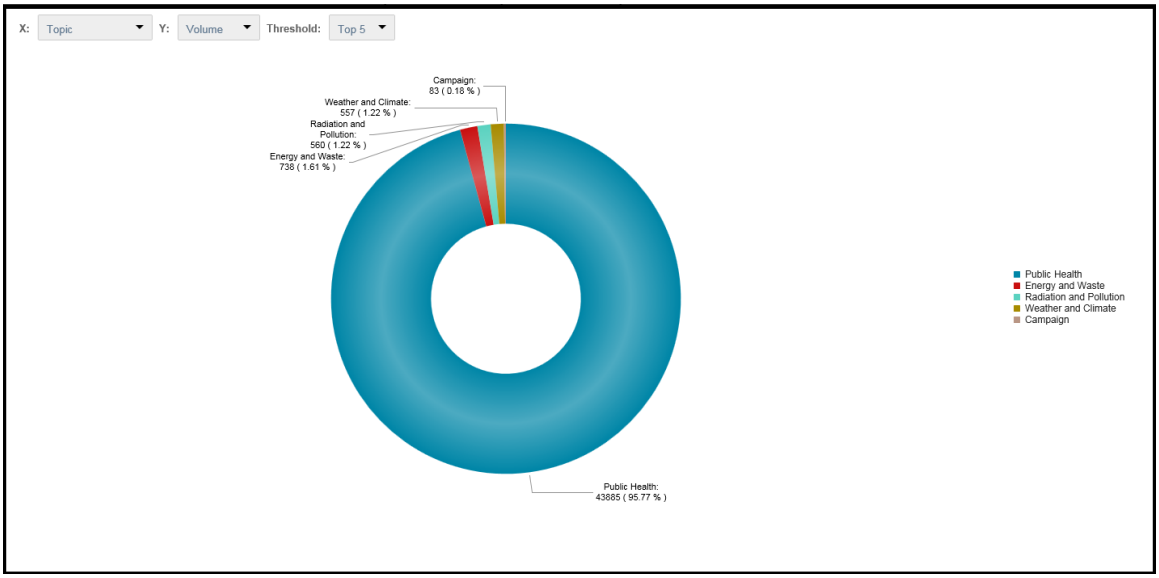



Figura 46: Donut

## d) Mapa

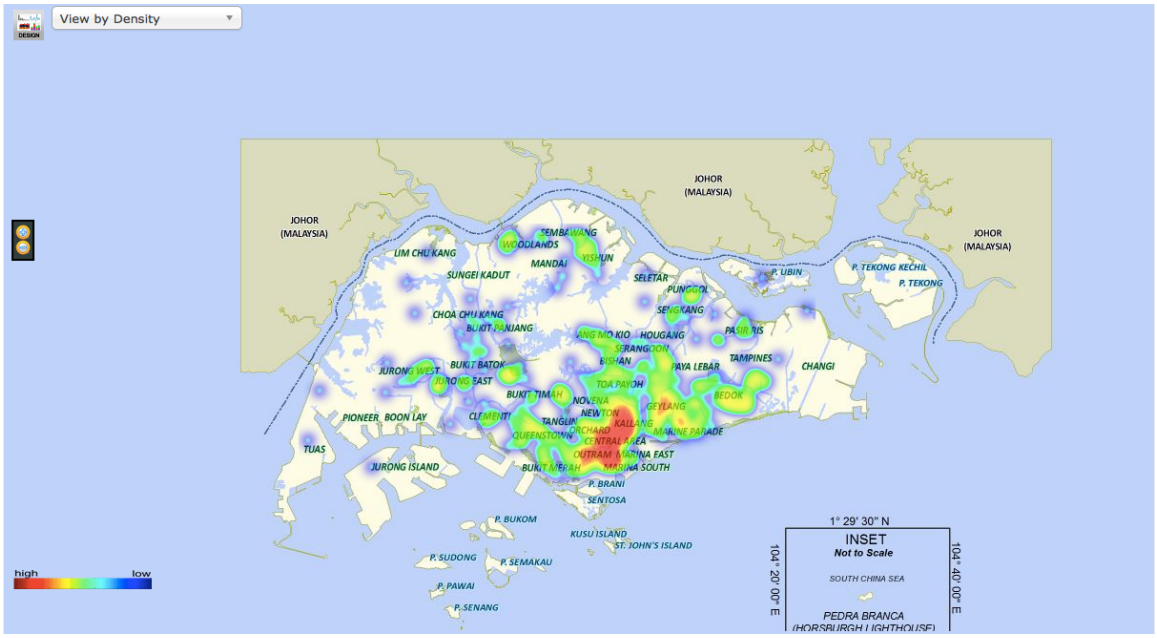



Figura 47: Mapa

## e) Gráfico circular

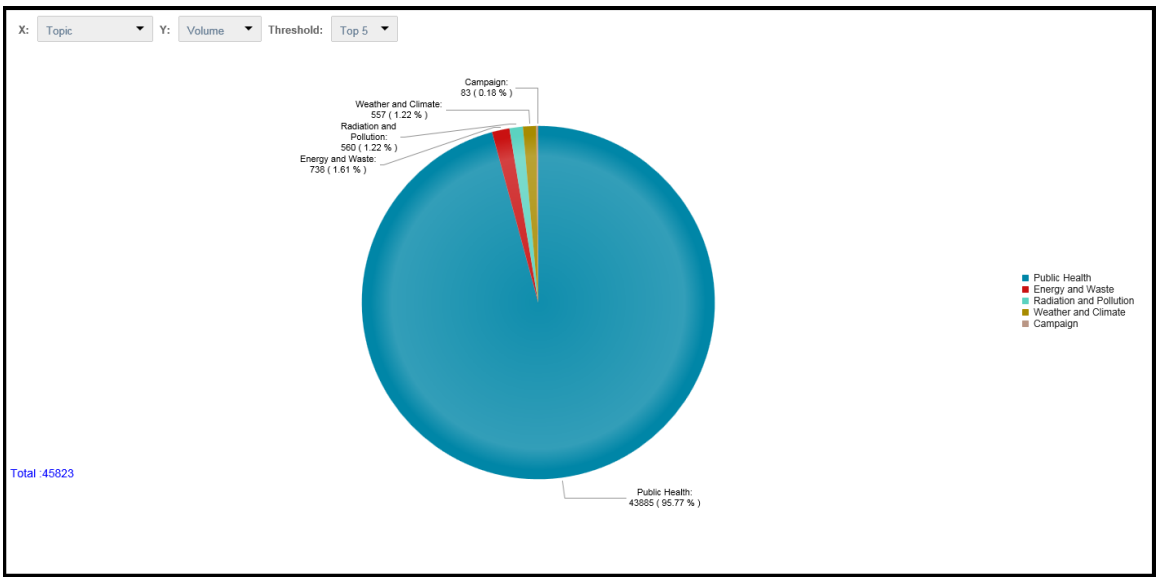



Figura 48: Gráfico circular

## f) Gráfico de dispersión

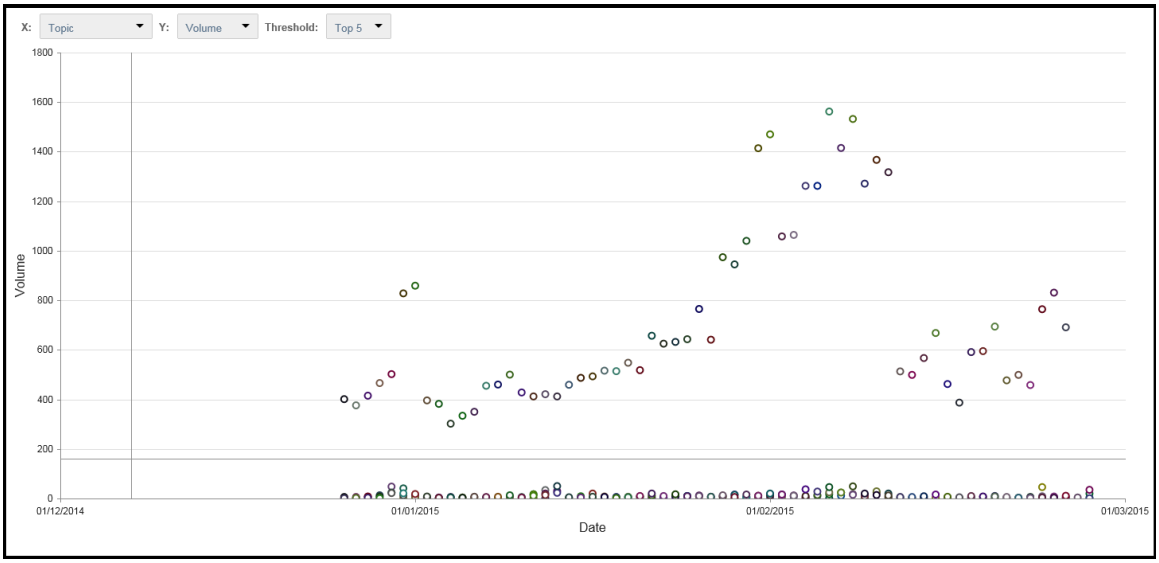



Figura 49: Gráfico de dispersión

g) Nube de etiquetas: permite al usuario ver los temas emergentes. Los colores representan el sentimiento de los temas


Figura 50: Nube de etiquetas

h) Ratio

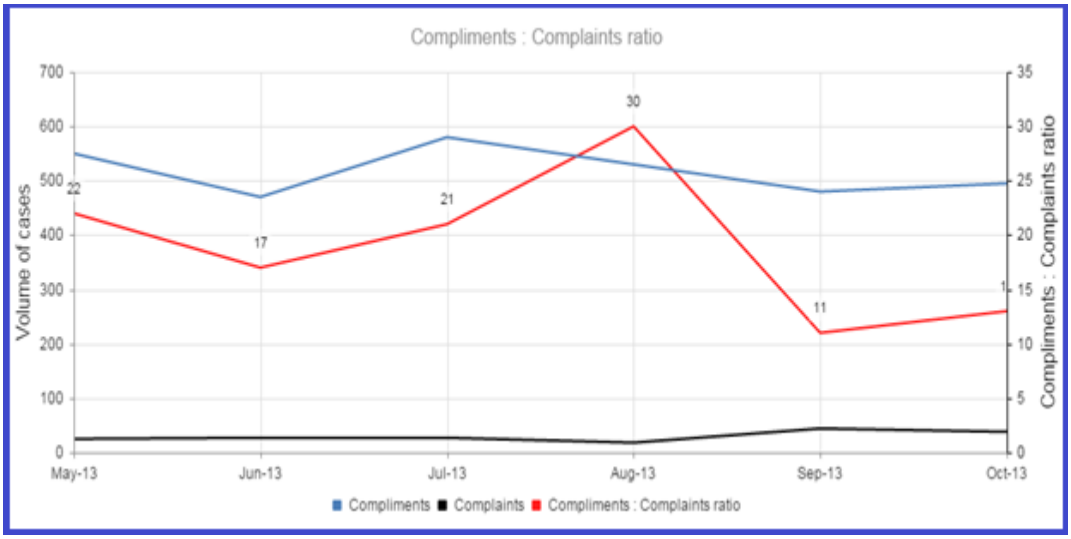



Figura 51: Ratio

## i) Gráfico de barras

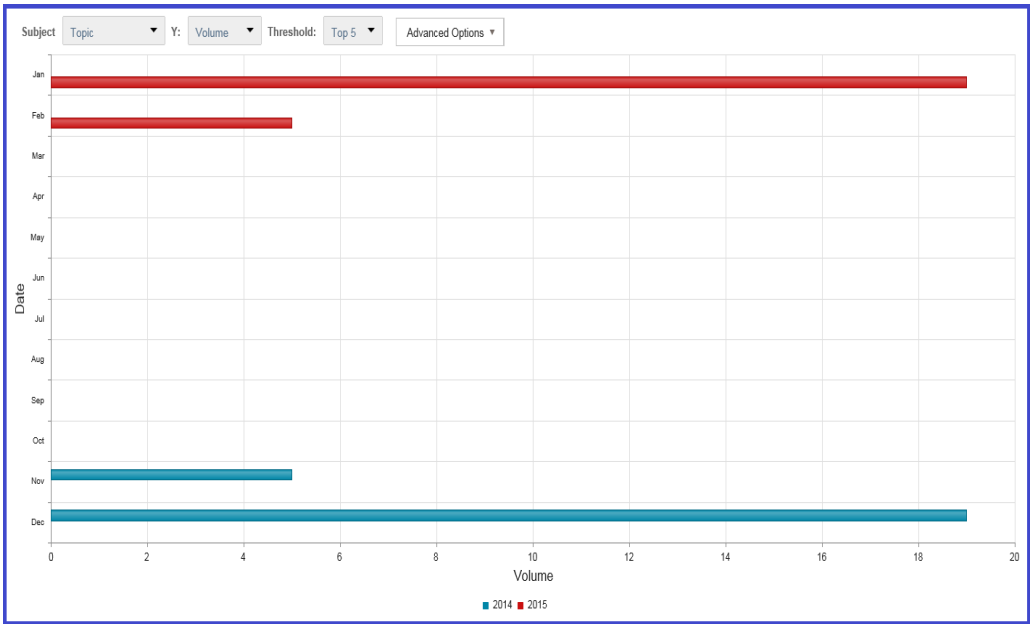



Figura 52: Gráfico de barras

## j) Gráfico de columnas

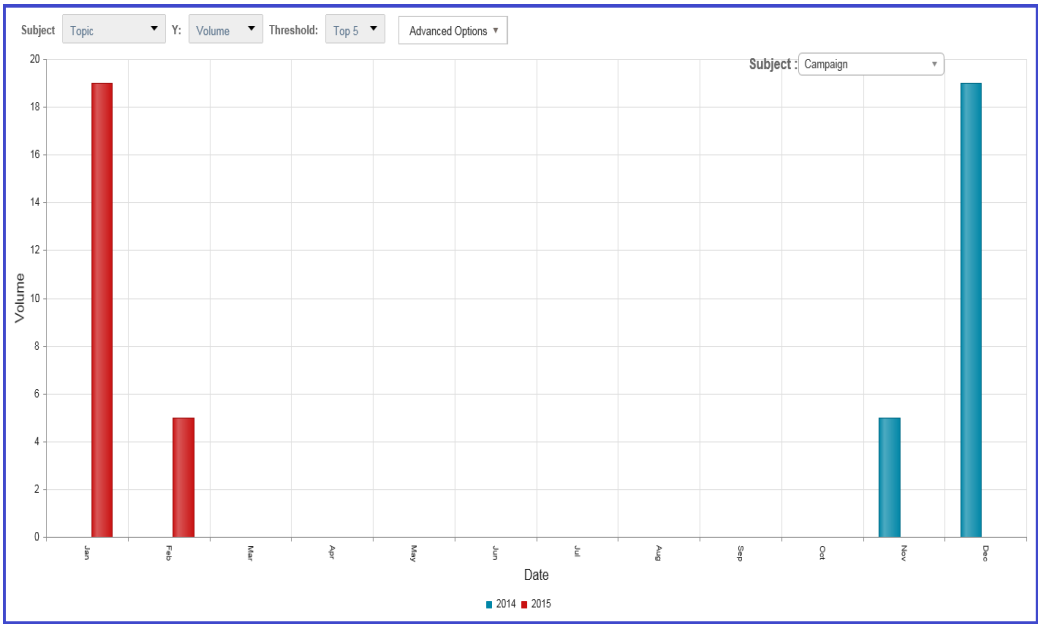



Figura 53: Gráfico de columnas

k) Múltiples líneas

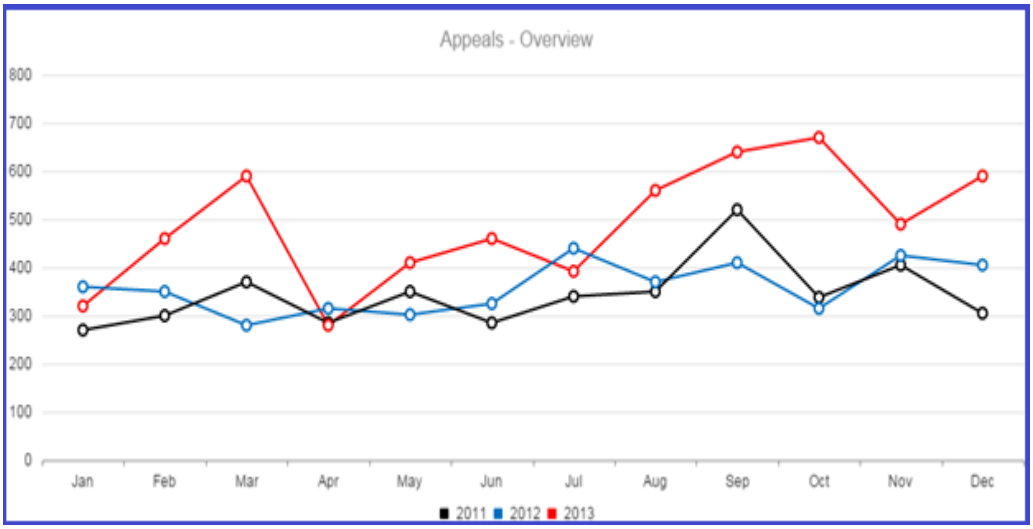



Figura 54: Gráfico con múltiples líneas

l) Área y líneas

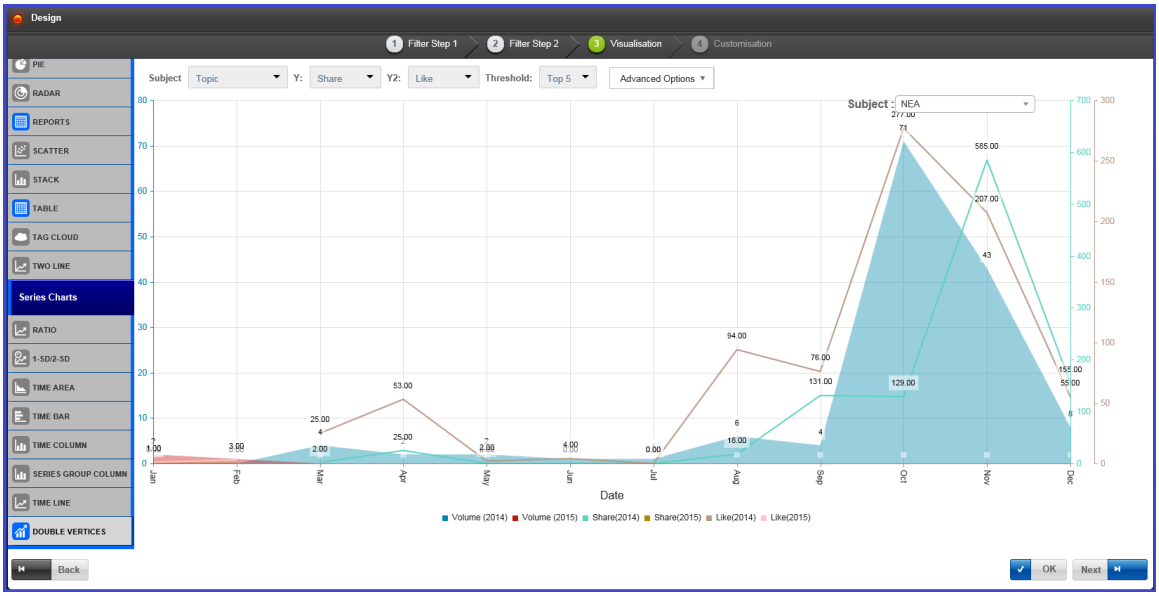



Figura 55: Gráfico con área y líneas

# 1.4. Datos subyacentes.

Los datos subyacentes son los contenidos que se ocultan detrás de los gráficos. Estos permiten revelar los patrones que se observan en los datos.

El sistema mostrará un resumen del contenido de los artículos, pero además también destacara sus entidades, el tema en el que se clasifica el contenido, la fecha de publicación y otros datos de interés.

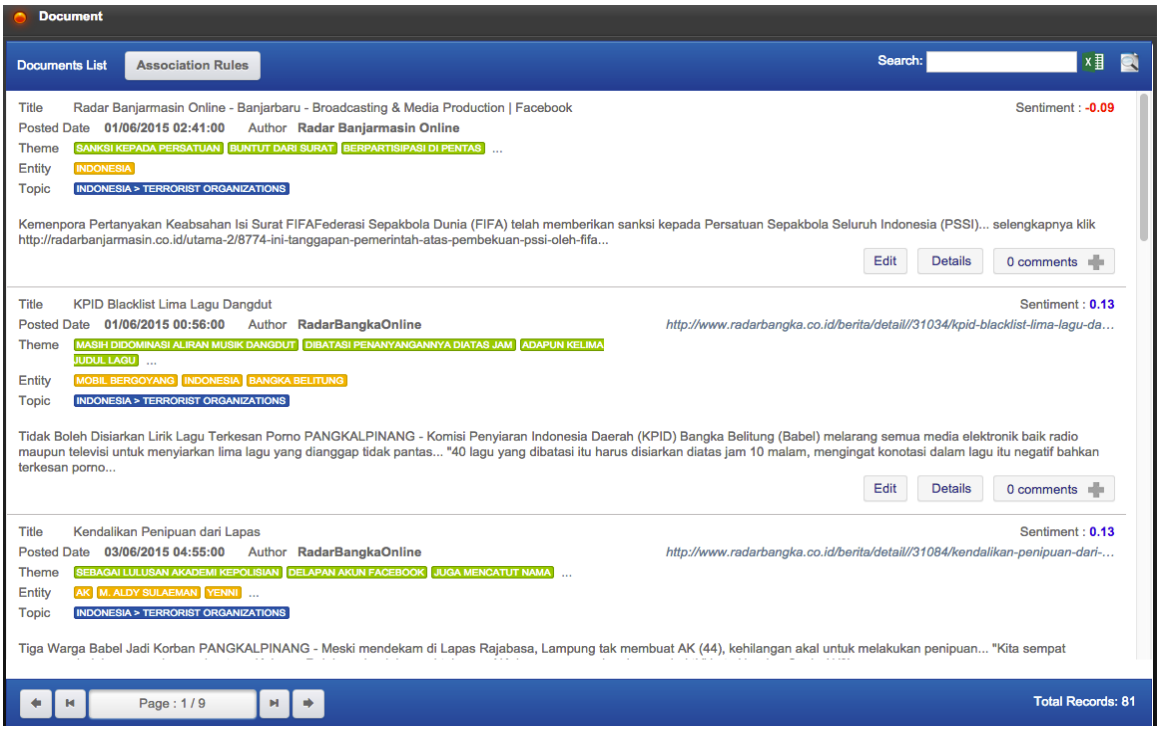



Figura 56: Datos subyacentes

## 1.5. Análisis de sentimientos

El sistema no tan solo analiza automáticamente todos los documentos que almacena en el sistema, también ofrece la posibilidad al usuario de analizar textos externos. El usuario podrá lanzar la herramienta de análisis de textos en el menú principal.

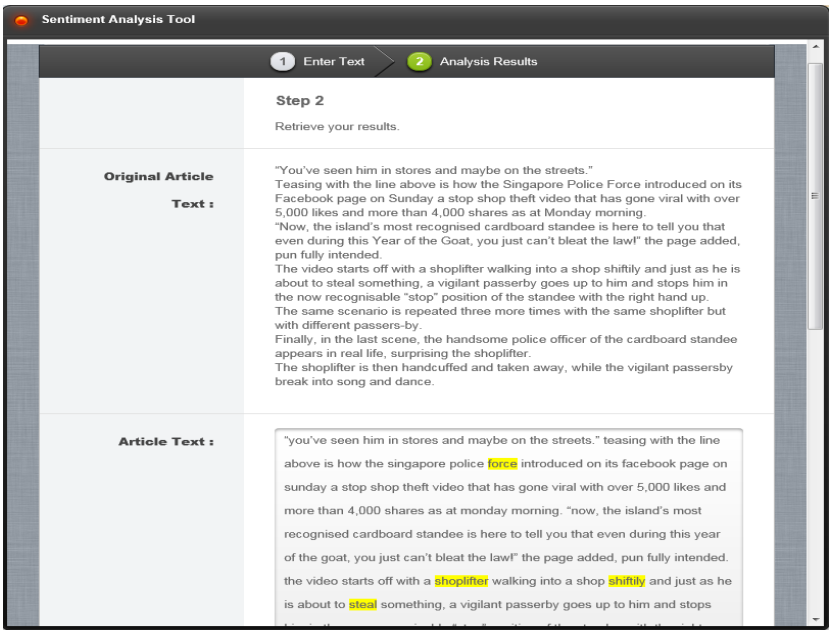



Figura 57: Herramienta para el análisis de sentimientos

El usuario encontrará un campo de texto libre en el cual puede introducir cualquier texto de hasta 1500 caracteres. Una vez que el usuario ha introducido su texto, tendrá que hacer clic en el botón "Next"

El sistema procesara el texto y le mostrará el sentimiento del documento, así como distintas palabras que influencian el resultado del mismo

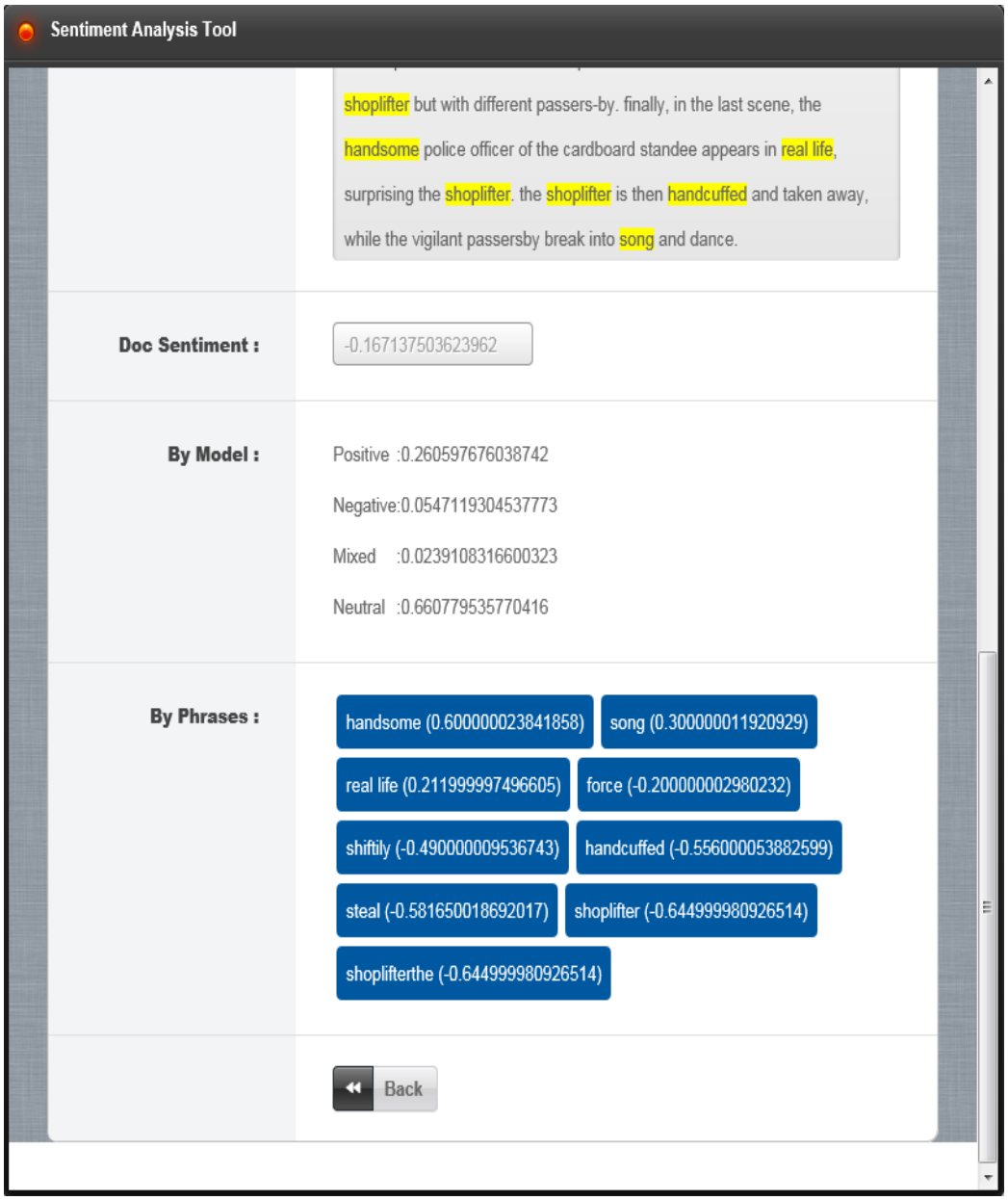



Figura 58: Resultado del análisis de sentimientos

## 1.6. Alertas y notificaciones

El sistema permite a los usuarios recibir notificaciones en su correo electrónico cuando el sistema ha encontrado documento con palabras o textos que el usuario ha predefinido. Esta es funcionalidad es muy útil cuando existe un determinado

grupo de usuarios que está enfocado a ciertas tareas como analizar el sentimiento de un tema político o social.

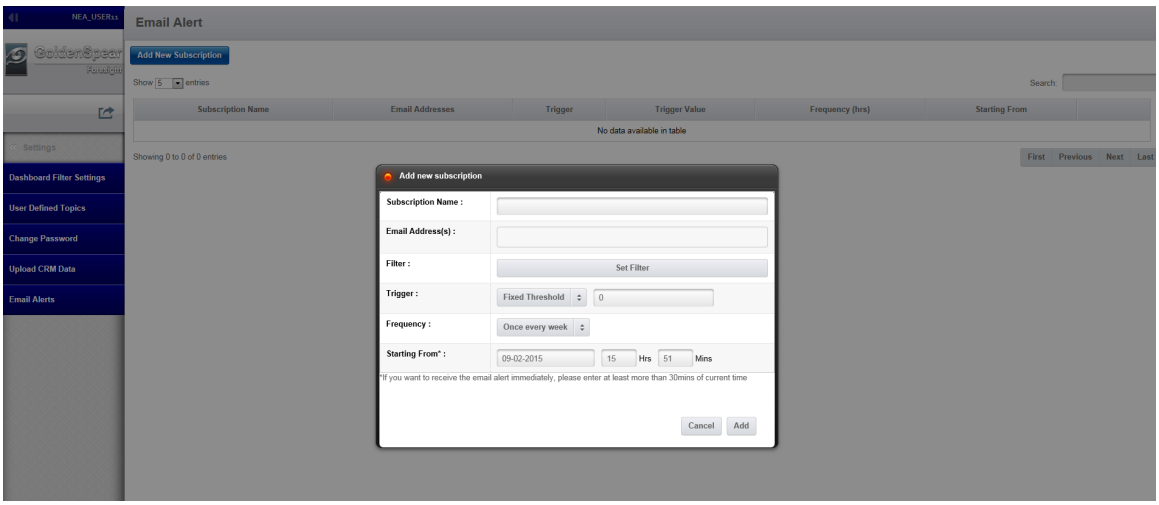



Figura 59: Nueva alerta

El usuario encontrará dentro del menú preferencias la opción “Email Alerts”. Una vez en esta página, el usuario tiene un listado de aquellas alertas que se encuentran activas.

Si el usuario desea añadir una nueva alerta deberá hacer clic en *“Add new alert”*, una ventana emergente se abrirá, en ella se introducirán los datos de la alerta.

Una vez el sistema encuentre algún documento con las palabras introducidas por el usuario, este recibirá una notificación por correo electrónico con la URL donde se encuentra el artículo. A continuación se muestra un ejemplo

| Posted Date | Name of Site | Thread Name | Author | URL | Sentiment |
|---|---|---|---|---|---|
| **2/7/2013**<br><br>**9:44:17 AM** | Website URL | ThreadName_x | AuthorName1 | http://xx.news.yyy.com/123456.html | 0.5 |
| **2/7/2013**<br><br>**10:55:17 AM** | Website URL | ThreadName_y | AuthorName2 | http://xx.news.yyy.com/765432.html | -0.5 |

Tabla 2: Ejemplo de notificación por correo electrónico

## 1.7. Data discovery

En el menú principal encontramos la opción ‘Discovery’. Esta funcionalidad está dividida a su vez en dos diferentes opciones ‘Web Search’ o ‘Local Search’

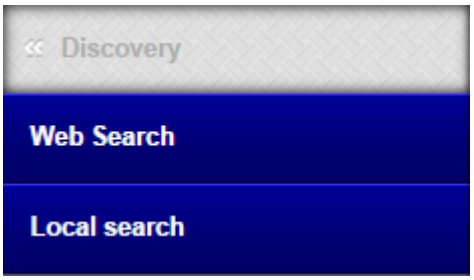



Figura 60: Menú Discovery

'Web Search' permite a los usuarios buscar contenido en internet. Una vez el usuario ha encontrado los enlaces que le interesan, GOV TA analizará el contenido y mostrará un resumen del mismo.

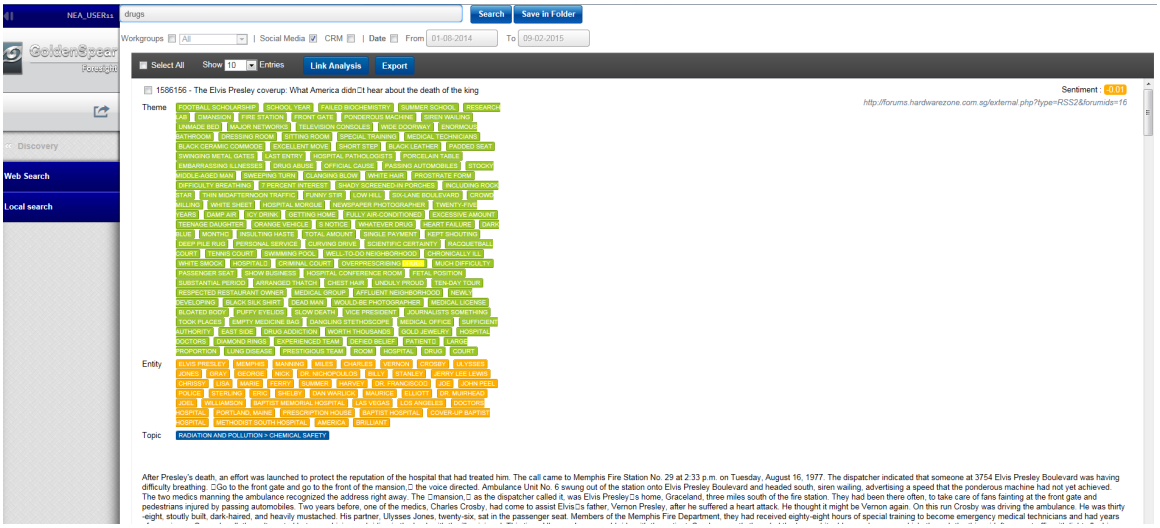



Figura 61: Web Search

Por otra parte, con la opción 'Local search' podremos buscar contenidos de nuestro interés dentro de la base de datos del sistema GOV TA. Una vez obtenido los resultados, es posible generar un grafo que muestra las conexiones entre los resultados.

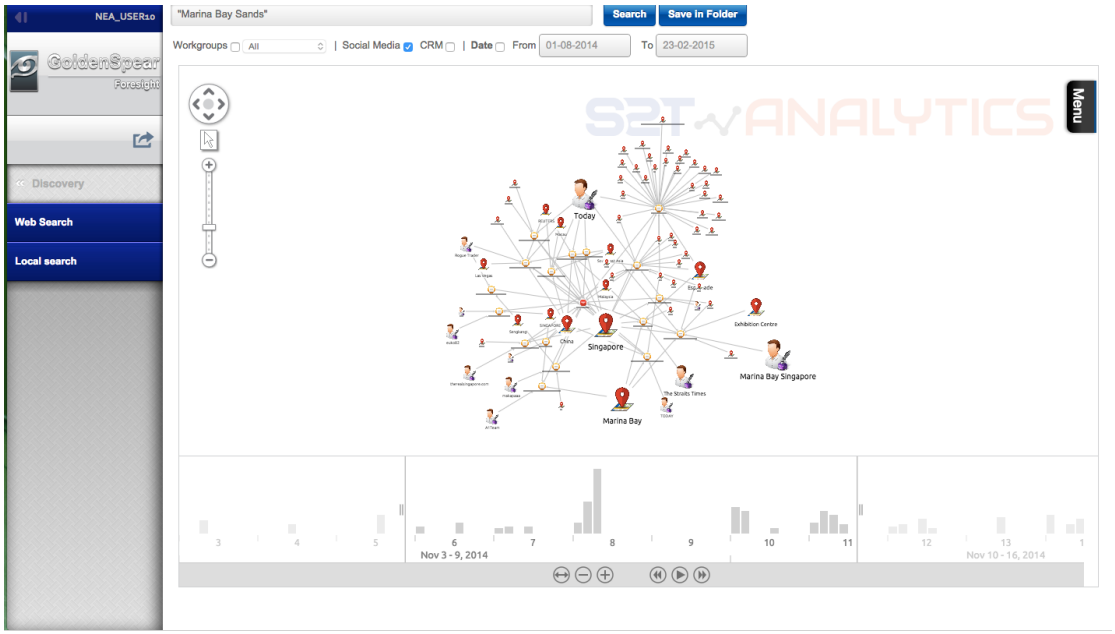



Figura 62: Link Análisis