



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Medición de la actividad de los grupos de Facebook para la evaluación del aprendizaje de los alumnos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Andriy Yatsyk

Tutor: José Luis Poza Luján
Ángeles Calduch Losa
Miguel Rebollo Pedruelo

2016/2017

Resumen

Conocer la actividad y el estado emocional de los alumnos son factores que ayudan a un docente a moldear y crear un ambiente de aprendizaje óptimo. En este proyecto se presenta un método para conseguir este objetivo en un grupo de Facebook (FB). El método se basa en el análisis de las publicaciones, comentarios y las expresiones mediante los *emoticonos*. De esta manera se ayuda a entender y moldear la enseñanza de la asignatura a los alumnos de una manera mas instructiva y entendible. Para ello se ha llevado a cabo una aplicación web que realiza un análisis sobre los datos estadísticos. El profesorado puede consultar las gráficas resultantes y ver la evolución de los grupos que gestiona en FB.

Palabras clave: Facebook, *emoticonos*, aplicación web, datos estadísticos, gráficas resultantes.

Resum

Conèixer la activitat i el estat emocional dels alumnes són factors que ajuden a un docent a modelar i crear un ambient de aprenentatge òptim. En aquest projecte es presenta un mètode per a conseguir aquest objective. El mètode es basa en analitzar les publicacions, comentaris i les expressions mitjançant els *emoticones*. Ajudant d'aquesta manera a entendre i modelar la ensenyança de la assignatura als alumnes de una manera més instructiva i intel·ligible. Per a això s'ha dut a terme una aplicació web que realitza una anàlisi sobre les dades estadístiques. El professorat pot consultar les gràfiques resultants i veure l'evolució dels grups que gestiona en FB.

Paraules clau: Facebook, emoticones, aplicació web, dades estadístiques, gràfiques resultants.

Abstract

To know the activity and the emotional state of students are factors that help a teacher to shape and create an optimal learning environment. This project presents a method to achieve this goal. The method is based on the analysis of publications, reviews and expressions using emoticons. In this way helping understand and shape the subject teaching students in a more instructive an understable way. This is carried by a web application that performs an analysis of statistical data. The faculty can check the resulting graphs and see the evolution of groups that they supervise in FB.

Keywords: Facebook, emoticons, web application, statistical data, resulting graphs.

Índice

Apartados

1.	Introducción	11
1.1.	Planteamiento	11
1.2.	Objetivo	11
1.3.	Metodología	12
1.4.	Descripción del documento	14
2.	Entorno de desarrollo.....	16
2.1.	Introducción	16
2.2.	Sistemas similares	16
2.2.1.	Facebook Insights (www.facebook.com/insights).....	16
2.2.2.	AgoraPulse (https://www.agorapulse.com/es/)	17
2.2.3.	Blitzmetrics (https://blitzmetrics.com/)	18
2.2.4.	Komfo (https://komfo.com)	18
2.2.5.	PostAcumen (http://postacumen.com)	19
2.2.6.	Quintly (https://www.quintly.com/).....	19
2.2.7.	Socialbakers (https://www.socialbakers.com)	20
2.2.8.	Sumall (https://sumall.com/)	20
2.2.9.	Crowdbooste (http://crowdbooste.com/)	21
2.2.10.	Grytics (https://grytics.com/es/).....	21
2.3.	Análisis.....	22
2.3.1.	Análisis general.....	22
2.3.2.	Análisis del alcance en Facebook	23
2.4.	Síntesis	24
2.5.	Conclusión.....	25
3.	Tecnologías utilizadas	26
3.1.	Django	26
3.1.1.	Frameworks	26
3.1.2.	Proyecto Django	27
3.1.3.	Conceptos básicos.....	27
3.2.	Base de datos no relacional.....	27
3.3.	Diccionario	28



3.4.	Terminal (Consola de Comandos).....	29
3.5.	Github	29
3.6.	Conclusión.....	29
4.	Especificación de requisitos.....	30
4.1.	Introducción	30
4.2.	Propósito	30
4.3.	Resumen	30
4.3.1.	Personal involucrado	30
4.3.2.	Definiciones, abreviaturas y acrónimos.....	31
4.4.	Descripción general	32
4.4.1.	Perspectiva del producto	32
4.4.2.	Funcionalidad del producto.....	33
4.4.3.	Características de los usuarios	34
4.4.4.	Restricciones	35
4.4.5.	Precondiciones y dependencias	35
4.4.6.	Evolución previsible del sistema.....	35
4.5.	Requisitos específicos.....	36
4.5.1.	Requisitos comunes de las interfaces	36
4.5.2.	Requisitos funcionales.....	37
4.5.3.	Otros requisitos	38
4.6.	Conclusión.....	38
5.	Diseño.....	39
5.1.	Introducción	39
5.2.	Problemas al recoger la información	39
5.3.	Especificación conceptual.....	39
5.4.	Arquitectura de capas	40
5.4.1.	Capa de persistencia	40
5.4.2.	Capa lógica o de negocio.....	42
5.4.3.	Capa de presentación.....	46
5.5.	Conclusión.....	50
6.	Implementación y evaluación	51
6.1.	Entorno de desarrollo	51
6.1.1.	Entorno hardware	51
6.2.	Instalación y configuración	51

6.2.1.	Configuración de settings.....	51
6.3.	Comunicación entre Django y MongoDB	52
6.3.1.	Configuración de URL's	52
6.3.2.	Configuración de Views en Django	53
6.3.3.	Funcionamiento de Templates en Django	57
6.4.	Conclusión.....	59
7.	Pruebas	60
7.1.	Behat.....	60
7.2.	Conclusión.....	62
8.	Conclusiones	63
8.1.	Conclusiones generales	63
8.2.	Desafíos.....	64
8.3.	Aportaciones tecnologías.....	64
8.4.	Posibles mejoras.....	65
9.	Bibliografía.....	66

Ilustraciones

Ilustración 1: Desarrollo en espiral.....	13
Ilustración 2: Versiones de Django.....	27
Ilustración 3: Estructura de diccionario.....	29
Ilustración 4: Casos de uso.....	33
Ilustración 5: Especificación conceptual del proyecto.....	39
Ilustración 6: Diagrama BD.....	40
Ilustración 7: CU01 Identificación del usuario.....	42
Ilustración 8: CU02 Monitorización de actividad en FB.....	43
Ilustración 9: CU03 Cálculo de sentimientos.....	44
Ilustración 10: CU04 Cálculo de impacto de las publicaciones.....	45
Ilustración 11: CU05 Visualización de resultados.....	46
Ilustración 12: Mockup de inicio de sesión.....	47
Ilustración 13: Grupos disponibles.....	48
Ilustración 14: Pantalla de carga para extracción.....	49
Ilustración 15: Resultados análisis grupo.....	50
Ilustración 16: Automatización Behat.....	60

Tablas

Tabla 1: Análisis general.....	23
Tabla 2: Análisis del ámbito de la analítica.	24
Tabla 3: Características del sistema.	25
Tabla 4: Miembro Andriy Yatsyk.	31
Tabla 5: Miembro Jose Luis Poza Luján.....	31
Tabla 6: Miembro Ángeles Calduch Losa.	31
Tabla 7: Miembro Miguel Rebollo Pedruelo.	31
Tabla 8: Miembro Alumnos.	31
Tabla 9: Casos de uso.	34
Tabla 10: Usuario profesor/profesora	35
Tabla 11: Usuario Scraper	35
Tabla 12: Requisitos funcional RF01	36
Tabla 13: Requisitos de interfaz RF02.....	37
Tabla 14: Requisitos de interfaz RF03.....	37
Tabla 15: Requisitos de interfaz RF04.....	37
Tabla 16: Requisitos de interfaz RF05.....	38
Tabla 17: Tabla Requisitos Generales	38

Código

Código 1: Configuración de settings.	51
Código 2: Configuración URL base.	51
Código 3: Consulta a MongoDB.....	52
Código 4: Redirección de accesos a URL's.	52
Código 5: Carga de diccionario.	53
Código 6: Configuración login.....	53
Código 7: Comprobación del usuario en BD.....	54
Código 8: Guardado el grupo del usuario.	54
Código 9: Renderizado de HTML.	54
Código 10: Extracción de contenido.	55
Código 11: Bucle calcula sentimientos.	56
Código 12: Almacenamiento de resultado.	56
Código 13: Recuperación de datos de BD.....	57
Código 14: Cálculo de impacto de las publicaciones.	57
Código 15: Visualización de los resultados.	58
Código 16: Resultado del test.....	61

1. Introducción

1.1. Planteamiento

Para el docente tiene mucha importancia el conocimiento del estado emocional de sus alumnos. Dicho conocimiento permite y ayuda a comprobar si las clases se desarrollan de la manera esperada o no. Gracias a ello es posible mejorar el estado emocional para el mejor desarrollo de las clases y el aprendizaje de los alumnos. Llevar a cabo esta tarea es bastante complejo, ya que requiere realizar un análisis del estado emocional de los alumnos sin influir en su decisión ni su reacción en la respuesta a la hora de adquirir las medidas.

Hoy en día, el uso de dispositivos móviles llamados *Smartphone* y las redes sociales ha evolucionado hasta el punto en el que su uso es casi diario en la vida cotidiana, sea por ocio, estudio o simplemente fuente de información en algunos casos. La gran mayoría de los jóvenes dispone de una cuenta en las redes sociales donde principalmente comparten o discuten temas de mayor interés.

Con el fin de llevar a cabo esta idea para el mejor desarrollo de las clases, este documento describe el desarrollo un programa de *software* cuyo objetivo es analizar y medir la actividad de los alumnos mediante la red social de Facebook. Para ello se extraen los datos de la red social, se almacenan en una base de datos y posteriormente se analizan para que el profesorado pueda visualizarlos. También indica recomendaciones para mejorar la actividad en el grupo. Este programa está orientado como objetivo principal al docente, independientemente de la temática de la asignatura o tema impartido.

1.2. Objetivo

El objetivo principal de este proyecto es desarrollar un sistema que permita presentar un análisis visualizado, proporcionando de esta manera información al profesorado de cómo afecta el desarrollo de su docencia al alumnado. Esto permite al docente tomar decisiones sobre la evolución u orientación de la docencia que imparte.

Se ha tomado como base del presente proyecto el artículo "*Propuesta de parámetros y caracterización de los grupos de las redes sociales orientados a la docencia universitaria: experiencia y resultados*" (Poza, 2014), escrito entre otros por mis tutores del proyecto.

Todo esto se ha desarrollado para cumplir los siguientes objetivos específicos:

- Estudiar sistemas de desarrollo similares que den solución al problema planteado describiendo sus características generales para facilitar la especificación del sistema.
- Especificar mediante el uso de estándares, los requisitos del sistema y el alcance del proyecto, desde el punto de vista de la Ingeniería Informática.
- Diseñar el sistema mediante el uso de estándares para garantizar la cobertura de todos los requisitos este proyecto.
- Implementar el diseño realizado y comprobar su funcionamiento mediante técnicas de testeo para garantizar que los requisitos iniciales han sido llevados a cabo.
- Desarrollar una herramienta que extraiga la actividad emocional del texto de los usuarios en un grupo de FB.
- Ayudar a los docentes a conocer el grado de participación de sus alumnos en la asignatura mediante la red social de Facebook.

1.3. Metodología

En este apartado se describe el tipo de metodología utilizado para el desarrollo del sistema. Ha sido seleccionado el modelo en espiral, ya que nos permite reducir riesgos e introducir mejoras durante el desarrollo del proyecto desde los primeros pasos hasta el final de realización de este.

El **modelo en espiral**, propuesto originalmente por Boehm (1976), es un modelo de proceso de software evolutivo donde se junta la naturaleza de construcción de prototipos con los aspectos controlados y sistemáticos de otros modelos como el modelo lineal y secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software que no se basa en fases claramente definidas y separadas para crear un sistema.

En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones la versión incremental podría ser un modelo en papel o un prototipo, durante las últimas iteraciones se producen versiones cada vez más completas del sistema diseñado.

Las actividades a considerar en cada ciclo son descritas a continuación:

- **Análisis:**
En este paso se estudian los requisitos, sistemas similares, se identifican los riesgos y las posibles soluciones a estos.
- **Diseño:**
En esta etapa, con los datos obtenidos del paso anterior, se realiza un diseño del sistema.
- **Implementación:**
En el tercer paso se realiza la programación y las evaluaciones unitarias, tras haber elegido el modelo y lenguaje de desarrollo necesario.
- **Testing:**
En el último paso es donde se lleva a cabo un análisis exhaustivo del proyecto. Se realiza un test completo y la evaluación frente los objetivos propuestos. También se plantea si se debería continuar con un ciclo más, en el caso afirmativo, se crearán planes para la siguiente fase.

Cada iteración crea una versión del software, que se vuelve más completa hasta que el proyecto pasa a ser un producto cumpliendo los objetivos de la extracción y el análisis de datos.

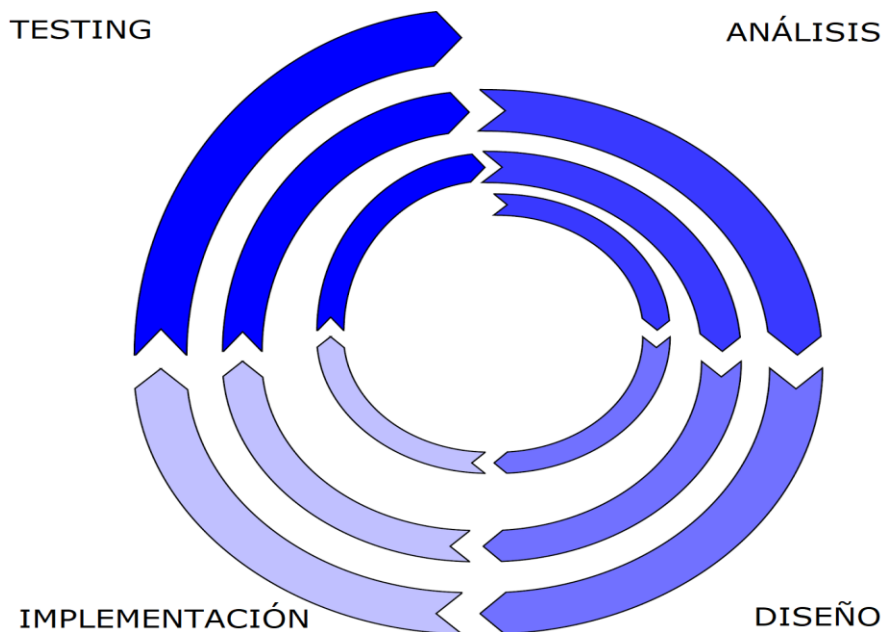


Ilustración 1: Desarrollo en espiral.

1.4. **Descripción del documento**

Este documento está estructurado en seis apartados para así detallar de una forma más concreta el desarrollo del sistema implementado. El contenido de los cinco siguientes se especifica a continuación.

- Apartado 2: Entorno de desarrollo

Uno de los primeros pasos a la hora de desarrollar un proyecto es hacer un estudio exhaustivo obteniendo un análisis de sistemas similares existentes actualmente. Esto nos permite valorar el estado y la situación del mercado disponible, los puntos fuertes y las características más relevantes para nuestro sistema. En este apartado se muestra un análisis de sistemas similares gracias los cuales posteriormente se crearon las características consideradas más importantes para el sistema.

- Apartado 3: Especificación de requisitos

Se realiza una especificación de requisitos de manera formal, utilizando el estándar IEEE 830, a partir de las características determinadas en el apartado anterior. De esta manera se ha creado la funcionalidad que tiene el sistema, ya que una vez determinados los requisitos funcionales de nuestro sistema se pudo llevar a cabo el diseño.

- Apartado 4: Diseño

Antes de cualquier implementación del sistema es importante crear su diseño de éste. Para abordar dicho desafío se creó un diagrama conceptual completo. Después del análisis, este bloque se encargará de detallar los diagramas de clases, los diagramas de secuencia, la estructura de las tablas que forman las bases de datos internas y externas, el diseño de la interfaz gráfica.

- Apartado 5: Implementación

El sistema se implementa según el diseño del apartado anterior. Para aumentar el detalle de los aspectos referentes a la implementación del proyecto, la explicación se separa mediante subapartados para cada parte del sistema. Se muestran capturas de pantalla del desarrollo del sistema con el propósito de validación del apartado de diseño comentado anteriormente. También se especifica qué tipo de tecnología es necesaria instalar para cada usuario del sistema.

- Apartado 6: Pruebas

Para verificar el correcto funcionamiento del producto final se muestran las pruebas realizadas. Finalmente se describen los test para validar que el sistema cumple con los requisitos propuestos.

- Apartado 7: Conclusiones

Para concluir la documentación, se extraen y comentan las conclusiones más relevantes. Se detallan también las aportaciones que se podrían desarrollar en un futuro, dejando las referencias para el apartado final.

2. Entorno de desarrollo

2.1. Introducción

Facebook es una red social extendida mundialmente, por lo que tiene la escalofriante cifra de 1.860 millones de usuarios⁽ⁱ⁾ activos al mes en todo el mundo. La gran mayoría de estos usuarios necesita usar herramientas que ayudan a la gestión de su perfil, grupos de páginas y eventos, para generar contenido, atención o ver el impacto que tiene la información en dicha red social.

Hoy en día podemos encontrar una gran cantidad de herramientas que nos facilitan estas tareas, pudiendo mostrar el número de mensajes publicados, reacciones, actividad del grupo y mucho más. La cantidad de información que podemos recuperar puede variar dependiendo de los datos de entrada, o bien de qué tipo de aplicación utilizamos para este fin.

La gran mayoría de sistemas para este uso de análisis y gestión son aplicaciones web. Además, es muy importante para el usuario utilizar la herramienta que sea más adecuada para su uso deseado.

En este apartado se ha realizado una búsqueda de diferentes tipos de sistemas similares probando y analizando su funcionamiento. Una vez hecho el análisis, se ha determinado las características que tiene nuestro sistema y de esta manera se explica porqué se ha hecho el uso de la tecnología en la que está desarrollado nuestro proyecto.

2.2. Sistemas similares

2.2.1. Facebook Insights (www.facebook.com/insights)

Es una herramienta para ver estadísticas de nuestras páginas o grupos. Nos indica las reacciones, comentarios, contenido compartido y puede ayudar a rastrear el número de usuarios activos para comprender mejor el rendimiento.



Lo que proporciona es:

- Descarga informes de todos los parámetros de la página en formato CSV y Excel.
- El número de las publicaciones con más “Me gusta” en los grupos de Facebook.

(i) 1.860 millones de usuarios. (TreceBits)

<http://www.trecebits.com/2017/02/02/facebook-ya-tiene-1-860-millones-de-usuarios/>

- El número de visitas a la página de Facebook.
- Gráficos de todos los parámetros.
- Selecciona y analiza cada una de nuestras publicaciones con todos los datos de interacción.
- Análisis de todos los vídeos subidos, con sus correspondientes parámetros como son las menciones de personas y mensajes *hashtag*.
- Fecha y hora de publicación.
- El tipo de publicación: Enlace, foto, vídeo o estado (texto).
- Las interacciones que realizan las personas en el grupo como comentar post publicados.

2.2.2. **AgoraPulse** (<https://www.agorapulse.com/es/>)



Herramienta que ofrece utilidades y aplicaciones para analizar estadísticas, gestionar el CRM, es decir administración basada en la relación con los clientes e incrementar el alcance de nuestras

publicaciones en FB.

Proporciona información sobre:

- Día y hora para publicar. Cuándo es más probable que las personas vean las publicaciones.
- Temas más llamativos (eficaces) para publicar. Formato de contenido y temas (tipo de contenido) que atraen a los participantes.
- Medición del alcance viral y la difusión de las publicaciones.
- Gestión de CRM. Identificación de personas más activas e influyentes.
- Moderación de comentarios y administración de páginas. Permite crear palabras clave y acciones personalizadas para realizarlas aún no estando presente.

2.2.3. **Blitzmetrics** (<https://blitzmetrics.com/>)

Herramienta para el análisis de métricas en diferentes redes sociales, ofrece versión gratuita y de pago.



La información que proporciona es:

- Informes por fecha. Ofrece una gran cantidad de rango temporal y se pueden recuperar informes de hace un año o períodos concretos.
- Informes gráficos. Dispone de diferentes formatos como el diagrama de barras o el diagrama de sectores.
- Informe demográfico. Esto es posible siempre que los usuarios tengan el perfil de Facebook público, de esta manera se obtiene de dónde son los usuarios o dónde viven y en qué tema participan más o cuál es el tema que impactó más (reaccionaron más usuarios) a las personas de una zona concreta.

2.2.4. **Komfo** (<https://komfo.com>)

Herramienta que ayuda a entender cómo a través de las redes sociales se puede impactar en los resultados de un negocio.



Esta herramienta proporciona:

- Analíticas de publicación por cada publicación individual. Incluye categorización para poder identificar claramente las publicaciones que son virales, las que son spam, las que generan *engagement* (interés) o las de mayor impacto.
- Permite el cálculo del ROI o "Retorno de Inversión" para analizar el rendimiento que la empresa tiene desde el punto de vista financiero. Esto se aplica más a las páginas de Facebook que obtienen beneficio monetarios.
- Monitorización en formato gráfico diferenciando entre pagado, orgánico y viral.
- Actividad de los fans y tabla de clasificación de influenciados, es decir, las personas que han hecho una publicación ante la que han relacionado más usuarios.

2.2.5. **PostAcumen** (<http://postacumen.com>)

Herramienta que ofrece un análisis de forma detallada de páginas de Facebook ofreciendo también funcionalidades de gestión como las publicaciones directas en la página de FB desde la plataforma de **PostAcumen**.



Esta herramienta realiza:

- Análisis de la competencia. Estadísticas funcionales de la competencia como son la cantidad de usuarios que están suscritos a la página o cuántas personas están online en un instante concreto.
- Visualizador en tiempo real. Posibilidad de monitorizar el contenido y actividad en tiempo real, como el número de personas que están visualizando la página.
- Análisis de fotos. Una de las características a destacar, las fotos compartidas en una página pudiendo observarlas en un panel visual.

2.2.6. **Quintly** (<https://www.quintly.com/>)

Quintly es una web basada en aplicaciones desarrolladas para ayudar a las empresas a monitorizar su rendimiento en las redes sociales, establecer puntos de referencia y optimizar el alcance de sus estrategias de medios sociales.



Las características de esta herramienta son:

- Análisis centralizado por ubicación de los usuarios.
- Reportaje elegante, es decir, dispone la posibilidad de exportar los resultados en formato **PDF** con una plantilla predefinida.
- Tableros con funcionamientos personalizados, de este modo, contactando con la empresa y por un precio establecido por ésta podemos solicitar la posibilidad de extraer datos según nuestras necesidades siempre y cuando sea solicitado y aprobado por Quintly.
- Nos proporciona una lista de los influenciadores clave, es decir, los usuarios que tuvieron más impacto de publicaciones.
- Diferentes formas de exportar los resultados, como **PowerPoint**, **XML**, **MySQL**.

- Posibilidad de conectar con la plataforma de **Facebook Analytic** en caso de tener más información poder mezclar los resultados. Esto es útil para los usuarios que empezaron a usar **Facebook Analytic** y ahora tienen más características que cubrir o simplemente quieren recolectar la información unificada.

2.2.7. **Socialbakers** (<https://www.socialbakers.com>)



socialbakers

Sirve para ayudar a clientes con una necesidad particular de tener una gran base de datos. Une principalmente la información valiosa sobre cómo el cliente funciona contra sus competidores, y evalúa con estrategias propias (no indicadas por la compañía).

Las características de esta herramienta son:

- Enorme banco de datos.
- Retroactividad de todos los datos de la página de Facebook mostrando solamente la información y sin realizar ningún análisis sobre esta.
- Fácil intercambio de perfiles en el caso de gestionar varias cuentas.
- Sin precedentes en recoger y comparar los conocimientos de los competidores (siempre que estén usando socialbakers).

2.2.8. **Sumall** (<https://sumall.com/>)

Es un tablero de instrumentos de análisis que no proporciona análisis por cuenta propia, pero en su lugar agrega datos de múltiples sitios como Google Analytics, Paypal, Raya (aplicación *social media* exclusiva para smartphones), MailChimp (proveedor de servicios de marketing por correo electrónico), Twitter y Facebook, sólo para cubrir los aspectos básicos. Esto permite ver todos los datos en un solo lugar.

sumall

Hay una versión gratuita, pero está muy limitada, ya que apenas nos proporciona opciones para el análisis, o por un precio de \$9/mes para la versión mejorada, que incluye alertas personalizadas, acceso prioritario a nuevas características y un gerente de cuenta para la recolección de las métricas.

2.2.9. **Crowdboost** (<http://crowdboost.com/>)



Crowdboost es una **social media analytics tool** que mide la presencia y el rendimiento de una empresa en los medios sociales, proporcionando cifras y análisis para optimizar sus estrategias. La plataforma está integrada con potentes herramientas que ayudarán a llegar a decisiones basadas en los datos. **Crowdboost** ayuda a determinar qué estrategias funcionan para los negocios y qué mensajes sirven de utilidad a su audiencia y pueden generar impactos positivos, es decir, aumentar el número de usuarios en cada post publicado.

Las características de esta plataforma son las siguientes:

- La plataforma procesa los datos en tiempo real.
- El sistema viene con las potentes herramientas de generación de informes, como son la ordenación, filtrado o búsqueda de datos.
- Poder programar automáticamente la publicación tanto de un mensaje como de un *tweet*.
- Proporciona métricas en tiempo real de Twitter y cuentas de Facebook.

2.2.10. **Grytics** (<https://grytics.com/es/>)

Grytics es una solución SaS (software como servicio) para realizar análisis en grupos de Facebook, para community managers (encargados de la gestión de redes sociales a nivel



GRYTICS
ANALYTICS FOR GROUPS

empresarial) que pueden realizar gestiones de las pagina. De esta manera se pueden beneficiar tanto las empresas pequeñas como las más grandes.

Las características de este servicio para los grupos de Facebook son:

- Importación del grupo de Facebook, es decir, importamos los grupos a analizar en la plataforma web de **Grytics**.
- Proporciona una estadística de los mensajes calculando el número de mensajes realizados por cada usuario.
- Muestra la cantidad de *emoticonos* totales que hay en la página de Facebook.

2.3. Análisis

Al realizar el estudio de sistemas existentes cuyas funciones principales son los análisis de datos en Facebook, se han aplicado dos tipos de análisis. Un análisis general y otro llamado de ámbito de la analítica. Los dos se muestran mediante tablas que indican el tipo de característica que cubren.

El análisis general, muestra las características de los sistemas con los objetivos más relevantes sobre los que se analizaría o extraería la información que deseamos tratar. Esta información se recoge en la Tabla 1.

Seguidamente se ha realizado el análisis del ámbito de la analítica. En éste se trata de analizar en que ámbito o sección de Facebook se pueden aplicar los sistemas recogidos. Esto se muestra en la Tabla 2.

En las dos tablas, la respuesta a las características puede ser de tres tipos diferentes:

- **SÍ:**
En caso de que dispone de dicha característica.
- **NO:**
Cuando no dispone de la característica especificada.
- **NA:**
Caso en el que no se especifica si dispone o no la característica.

2.3.1. Análisis general

Las características y especificaciones que han sido consideradas para el análisis general y que corresponden las columnas de la Tabla 1 son las siguientes:

- **Sistema:**
Nombre de la herramienta.
- **Gratuita parcial:**
Si una aplicación puede usarse de forma gratuita estando limitada en servicios que proporciona la versión de pago.
- **Gratuita:**
Indica que la aplicación es completamente gratuita o no.
- **Analítica de comentarios:**
Si los comentarios se pueden analizar más exhaustivamente, siempre teniendo en cuenta las palabras del comentario.

- **Solo páginas web Facebook:**
Si la herramienta solo está orientada a páginas de FB o a más redes sociales.
- **Cálculo de emociones sobre publicaciones:**
Si puede contar la cantidad de reacciones mediante los emoticonos que realiza la gente en las publicaciones de la página.
- **Freemium (Aplicación parcialmente gratuita):**
Estas son aplicaciones o software que disponen de una versión gratuita ya sea de prueba de días determinados o sin fecha límite. Sin embargo, para obtener todas las características y ventajas que el sistema proporciona, hay que realizar una compra o suscripción, normalmente mensual del programa.

SISTEMA	FREEMIUM	GRATUITA	ANALÍTICA DE COMENTARIOS	SOLO PAGINAS WEB FACEBOOK	CALCULO DE EMOCIONES SOBRE PUBLICACIONES.
Facebook Insights	NA	SÍ	NO	NO	NO
AgoraPulse	SÍ	NO	SÍ	NO	NO
Blitzmetrics	SÍ	NO	SÍ	NO	NO
Komfo	SÍ	NO	NO	SÍ	NO
Postacumen	SÍ	NO	NO	SÍ	NA
Quintly	SÍ	NO	SÍ	NO	NO
Scoreboard Social	SÍ	NO	NO	NO	NA
Socialbakers	SÍ	NO	NA	NO	NO
Sumall	SÍ	NO	NO	SÍ	NA
Crowdbooster	SÍ	NO	NO	NO	NO
Bevolve	SÍ	NO	NA	SÍ	NA
KeyHole	SÍ	NO	SÍ	SÍ	SÍ
Grytics	SÍ	NO	NO	SÍ	NO

Tabla 1: Análisis general.

2.3.2. Análisis del alcance en Facebook

A continuación, vemos las características que se consideran para el análisis del ámbito de la analítica que corresponde a la Tabla 2.

- **Sistema:**
Nombre de la herramienta.
- **Muro:**
Sección dentro de la red social de FB en la cual los usuarios publican contenido de interés.

- **Página:**
Las páginas de FB son las creadas por usuarios por un interés común, marketing para empresas, etc...
- **Grupo:**
Sección en la cual participan varios usuarios, sea por un tema de discusión o creación de eventos.
- **Perfiles**
Esto es referido a los perfiles de los usuarios de Facebook, ya que se puede consultar la información de estos.

También aparece una opción adicional como respuesta "Sí (público)". Esto se refiere en la característica de los perfiles, ya que los perfiles de los usuarios se pueden consultar siempre y cuando son públicos. De lo contrario no es posible acceder a la información del usuario por la política de privacidad.

SISTEMA	MURO	PÁGINA	GRUPOS	PERFILES
Facebook Insights	NO	SÍ	SÍ	NO
AgoraPulse	NO	NO	SÍ	NO
Blitzmetrics	NO	NO	NO	NO
Komfo	NO	NO	NO	NO
Postacumen	NO	NO	NO	NO
Quintly	NO	NO	SÍ	SÍ (público)
Scoreboard Social	NO	SÍ	NA	SÍ (público)
Socialbakers	NO	SÍ	NA	SÍ (público)
Sumall	NO	NO	NA	NO
Crowdbooster	NO	NO	NA	NO
Bevolve	NO	SÍ	NA	NO
KeyHole	NO	SÍ	NA	NO
Grytics	NO	SÍ	SÍ	SÍ (público)

Tabla 2: Análisis del ámbito de la analítica.

2.4. Síntesis

Una vez se ha realizado el análisis de diferentes sistemas similares, podemos establecer qué características serían las más deseables para nosotros, representadas en la Tabla 3. Esta tabla tendrá dos columnas con el siguiente significado:

- **Característica:**
Nombre de la característica, es utilizado el código CAXX para diferenciar cada una de las características a definir.

- **Descripción:**
Se describirá la característica en cuestión.

CARACTERÍSTICA	DESCRIPCIÓN
CA01	Los grupos de los cuáles extraer la información deseada.
CA02	Estadística de la actividad del grupo.
CA03	Indicación de publicaciones que han impactado.
CA04	Participación de usuarios en el grupo (mediante la presencia de comentarios o publicaciones).
CA05	Opción de inicio y cierre de sesión, pudiendo entrar con otro usuario.
CA06	Navegación por la interfaz de la web, volviendo a secciones anteriores.
CA07	Análisis de tipo de perfil del grupo en FB.
CA08	Estadísticas a fechas actuales de los grupos de FB.
CA09	Disponibilidad Web sin necesidad de instalar software adicional.
CA10	Análisis de otros grupos en el sistema.

Tabla 3: Características del sistema.

2.5. Conclusión

En este apartado 2 se han analizado diferentes sistemas y plataformas que ofrecen un análisis de páginas de FB, muros y grupos. Se han visto diferentes ventajas como las publicaciones de las propias plataformas y desventajas como son las versiones *freemium*, sistemas muy orientados al mundo de marketing o visualización de resultados muy pobres.

Gracias a este análisis se han establecido las características para nuestro proyecto que aparecen en la Tabla 3. Algunas son básicas como son la navegación por la interfaz o el inicio de sesión en FB y otras son más complejas como las participaciones de usuarios en el grupo y el análisis del perfil del grupo en FB.

3. Tecnologías utilizadas

3.1. DJANGO

A continuación, se procede a describir en qué consiste Django y cómo funciona (la implementación y funcionamiento completo está explicado posteriormente en este documento).

“**Django** es un *framework* de alto nivel en **Python** Web que fomenta un desarrollo rápido y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de la molestia de desarrollo web, por lo que puede centrarse en escribir su aplicación. Es de código abierto y gratuito.”

(www.djangoproject.com , Django)

3.1.1. Frameworks

Existen muchos **Frameworks** para el entorno web (**Symfony, Ruby on Rails, AngularJS, Grails**). **Django** destaca sobre el resto gracias a las siguientes características:

- **Código abierto:**
Django fue diseñado para promover el acoplamiento débil y la estricta separación entre las piezas de una aplicación.
- **Restricciones hardware:**
El sistema no fue diseñado para un hardware concreto y tampoco tiene unos requisitos mínimos para poder utilizarlo.
- **Portabilidad:**
El producto final desarrollado es un conjunto de archivos y carpetas, por lo que es migrable a cualquier tipo de plataforma que pueda ejecutar Python y sus dependencias.
- **Componentes gráficos:**
La interfaz del usuario está basada en HTML (lenguaje de marcas de hipertexto), por lo que cualquier navegador puede visualizarlo.
- **Servicios y patrones:**
Por defecto no ofrece ningún servicio al respecto al usuario final. Sin embargo, nos ofrece un patrón de arquitectura de *software* Modelo-Vista-Controlado y también un patrón de diseño *Model-Template-View*.
- **Seguridad:**
Django dispone de librerías para protegernos de los ataques más comunes en la web.

3.1.2. Proyecto Django

A continuación, se exponen los elementos básicos que forman parte de un proyecto Django:

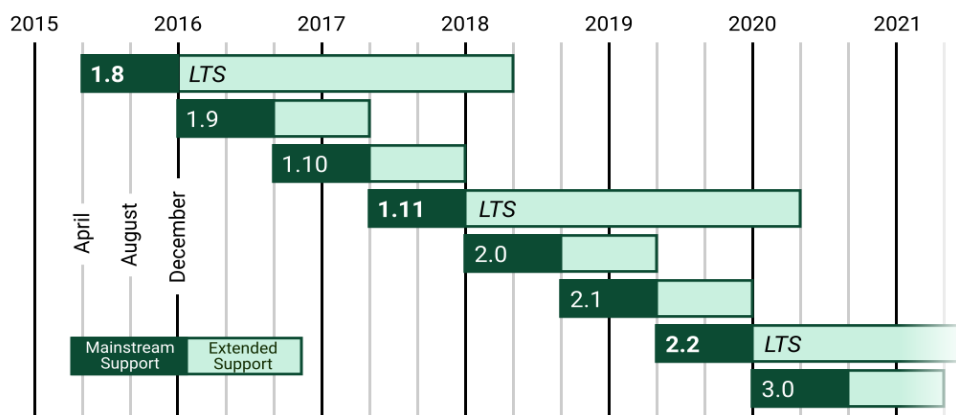


Ilustración 2: Versiones de Django.

- **Model (Modelo):**
La capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- **Template (Plantilla):**
La capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: cómo algunas cosas son mostradas sobre una página web u otro tipo de documento.
- **View (Vista):**
La capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: esto sería como un puente entre el modelo y la plantilla.
- **Url (Enlace):**
A diferencia de las anteriores, aquí solo especificaremos si accedemos a una ruta concreta que llamaremos a la vista, es decir, a la capa lógica de negocio.

3.1.3. Conceptos básicos

Desde su aparición, Django fue evolucionando, aparecieron distintas versiones. Con cada una de ellas se incorporaron mejoras de rendimiento y nuevas funcionalidades.

3.2. Base de datos no relacional

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los

documentos de una misma colección (concepto similar a una tabla de una base de datos relacional), pueden tener esquemas diferentes.

“MongoDB (que proviene de «*humongous*») es la base de datos **NoSQL** líder y permite a las empresas ser más ágiles y escalables. Organizaciones de todos los tamaños están usando MongoDB para crear nuevos tipos de aplicaciones, gracias a las aplicaciones es posible mejorar la experiencia del cliente, acelerar el tiempo de comercialización y reducir costes manteniendo la calidad.

Es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta.”

(<https://www.mongodb.com/es> , MongoDB)

Para entender mejor cómo es la base de datos que vamos a utilizar, primero comentaremos un concepto que nos ayudará a entenderlo mejor. Se trata de JSON:

“**JSON**, acrónimo de *JavaScript Object Notation*, es un formato de texto ligero para el intercambio de datos. **JSON** es un subconjunto de la notación literal de objetos de **JavaScript**, aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

Una de las supuestas ventajas de **JSON** sobre **XML** como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (*parser*) de **JSON**.”

(<https://es.wikipedia.org/wiki/JSON> ,JSON)

3.3. Diccionario

Para obtener el estado emocional de un grupo de FB, analizaremos las palabras de todo el grupo clasificando por criterio de si cumple un tipo de emoción o no.

Partimos de un diccionario de palabras, este es un fichero xls que tiene un mapa de emociones. Tiene un total de diez estados de emociones diferentes, y cuenta con 14.142 palabras diferentes con el valor de 1 si cumple la emoción y valor de 0 si no la cumple. De esta manera, no se obliga que una palabra pertenezca solo y exclusivamente una única emoción. El diccionario ha sido obtenido de “*NRC Word-Emotion Association Lexicon*” (Saif, 2016) (<http://saifmohammad.com/>)

En la Ilustración 3 podemos ver un ejemplo de la palabra **abandonado**.

	A	B	C	D	E	F	G	H	I	J	K
1	Palabras	Positivo	Negativo	Enfadado	Anticipacion	Disgustado	Miedo	Alegria	Triste	Sorprendido	Confiado
2	abandonado	0	1	1	0	0	1	0	1	1	0
3											

Ilustración 3: Estructura de diccionario.

3.4. Terminal (Consola de Comandos)

La interfaz de línea de comandos o interfaz de línea de órdenes (en inglés, *command-line* interface, CLI) es un método que permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple.

3.5. Github

Hay que comentar que se ha hecho uso del control de versiones **GitHub** guardado de forma periódica, ya que se hizo uso de la metodología espiral comentada en el apartado 1.3.

“**GitHub** es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones **Git**. Utiliza el framework **Ruby on Rails** por **GitHub, Inc.** (anteriormente conocida como **Logical Awesome**).”

(<https://github.com/> , GitHub)

3.6. Conclusión

En este apartado se han visto las tecnologías actuales de hoy en día y las ventajas de cada una de ellas. Comentar que MongoDB tiene problemas de compatibilidad, ya que depende de otras librerías para el buen funcionamiento junto con los modelos de Django. Por ello se realizó una conexión directa, como se explica con más detenimiento en el apartado de diseño.

4. Especificación de requisitos

4.1. Introducción

Una vez realizado el estudio de sistemas similares y el análisis de los mismos, la presente sección es una especificación de requisitos de software (ERS) para la aplicación que permite la obtención, procesamiento, análisis, visualización, cotejamiento y posible estimación del resultado. Se ha procedido al uso del estándar IEEE 830 "Especificación de requisitos de Software" para la definición formal de sistema.

4.2. Propósito

La presente especificación de requisitos detalla de manera completa las funciones básicas del proyecto y las especificaciones definen el sistema de desarrollado.

4.3. Resumen

El sistema se denomina ***Analisticsgroup***.

Analisticsgroup, es un sistema que proporciona análisis y estadísticas sobre los grupos de Facebook, tanto para la medición de la actividad como para la evaluación del aprendizaje de los alumnos. Dicho análisis se realiza en base a las publicaciones, comentarios y reacciones de los alumnos en los grupos.

4.3.1. Personal involucrado

Se ha determinado en este apartado las personas que están implicadas en el desarrollo del proyecto. Para simplificar la complejidad, se ha representado la información en tablas con una columna en la parte izquierda como título, con el siguiente significado:

- **Nombre:**
Nombre de la persona relacionada con el sistema.
- **Rol:**
Tarea que desempeña dicha persona en el proyecto.
- **Categoría profesional:**
El cargo que ocupa o qué titulación posee.
- **Responsabilidades:**
Qué es lo que se encarga de hacer.

- **Información del contacto:**

Correo u otra manera de contactar con la persona.

Nombre:	Andriy Yatsyk
Rol:	Desarrollo de Analisticsgroup
Categoría profesional:	Estudiante
Responsabilidades:	Desarrollar el sistema
Información del contacto:	anyat@etsinf.upv.es

Tabla 4: Miembro Andriy Yatsyk.

Nombre:	Jose Luis Poza Luján
Rol:	Supervisor
Categoría profesional:	Profesor Contratado Doctor
Responsabilidades:	Supervisión
Información del contacto:	jopolu@disca.upv.es

Tabla 5: Miembro Jose Luis Poza Luján.

Nombre:	Ángeles Calduch Losa
Rol:	Supervisor
Categoría profesional:	Profesora Colaboradora
Responsabilidades:	Supervisar el proyecto
Información del contacto:	mcalduch@eio.upv.es

Tabla 6: Miembro Ángeles Calduch Losa.

Nombre:	Miguel Rebollo Pedruelo
Rol:	Supervisor
Categoría profesional:	Profesor Titular de la UPV
Responsabilidades:	Supervisar el proyecto
Información del contacto:	mrebollo@upv.es

Tabla 7: Miembro Miguel Rebollo Pedruelo.

Nombre:	Alumnos de estadística
Rol:	Alumno
Categoría profesional:	Estudiantes
Responsabilidades:	Testers
Información del contacto:	A través de Ángeles Calduch Losa (profesora de estadística)

Tabla 8: Miembro Alumnos.

4.3.2. **Definiciones, abreviaturas y acrónimos**

Con el propósito aclarar y entender mejor los conceptos más relevantes y concretos del proyecto a continuación, se redacta una lista de los acrónimos y palabras con su definición.

Definición:

- **Sistema:**

Todo el desarrollo del proyecto "**Analisticsgroup**".

- **Usuario de Facebook:**
Persona registrada en la red social Facebook.
- **Grupo docente:**
Es un grupo compuesto por usuarios de Facebook, con el propósito de participar en temas de ámbito docente.
- **Administrador docente:**
Personal docente encargada de la gestión y supervisión de los grupos de docentes de Facebook.
- **Post:**
Hace referencia a una publicación de una noticia u comentario hecha por las personas que forman ese grupo.
- **Comentario:**
Texto que escribe una persona expresando su opinión, crítica o realizando una pregunta relacionada con el post.
- **Comentario Respuesta:**
Es el texto de respuesta frente a un comentario de un post.
- **Muro del grupo:**
Sección del grupo donde se publican todos los posts, comentarios y los comentarios de respuesta.
- **Perfil:**
Tipo de emoción en el cual se encuentra un grupo. Está basada en los comentarios de los usuarios del grupo.

Acrónimos:

- **BD:** (base de datos) Nos referiremos a la base de datos del sistema.
- **JSON:** (*Java Script Object Notation*). Notación de objetos en JavaScript.

4.4. Descripción general

4.4.1. Perspectiva del producto

El sistema desarrollado es un producto independiente, es decir, no forma parte de ningún proyecto colectivo ni en grupo. Está desarrollado únicamente por las personas descritas en las tablas del apartado 3.3.1.

4.4.2. Funcionalidad del producto

Con el objetivo de mostrar las funcionalidades del sistema sin entrar en un gran nivel de complejidad, a continuación se muestra un diagrama de casos de uso. De esta manera nos permite observar por una parte qué usuarios intervienen en el sistema y por otra parte, nos permite ver las acciones que desarrolla cada uno.

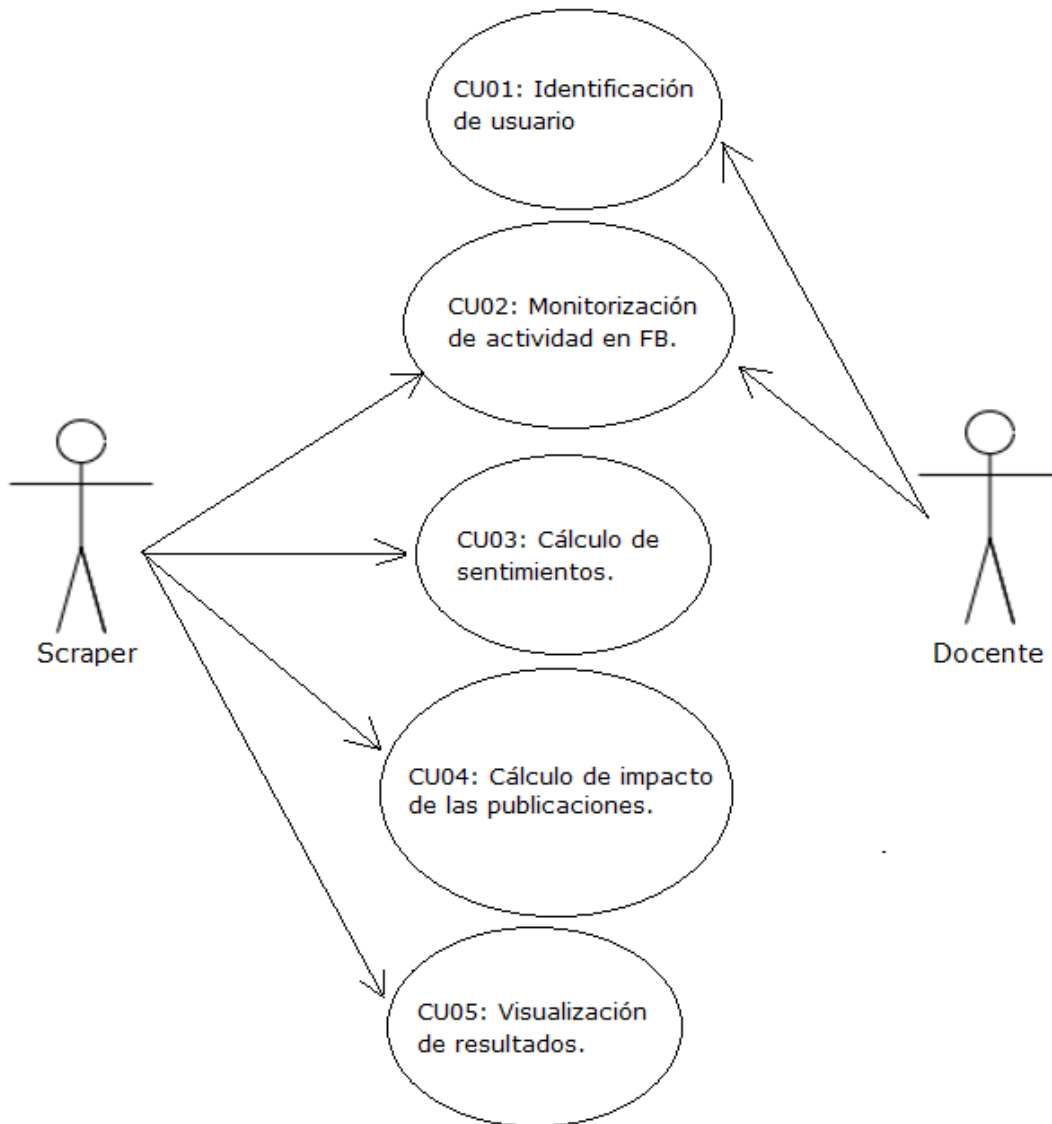


Ilustración 4: Casos de uso.

Para concretar el significado de la Ilustración 4 se ha hecho uso de la Tabla 9, cuyas columnas son descritas a continuación:

- **Caso de uso:**
Hace referencia al caso de uso, en concreto sigue un patrón de nombramiento CUXX.

- **Descripción:**
Se describe el caso de uso en cuestión.
- **Referencia:**
Determina de qué característica proviene. Estas han sido descritas en la Tabla 3.

Caso de uso	Descripción	Referencia
CU01	Un usuario hace uso del sistema para autenticarse y permitir la solicitud de permisos.	CA05
CU02	El acceso para la monitorización y navegación por el sistema sin necesidad de software adicional.	CA01 CA09 CA06
CU03	El correcto análisis de las palabras de cada grupo es calculado mediante Analisticsgroup .	CA07 CA10
CU04	Cálculo del impacto de las publicaciones y usuarios con mayor actividad en el grupo por su participación.	CA03 CA04
Cu05	Tablas y diagramas con la visualización de los resultados sobre el grupo de FB.	CA02 CA08

Tabla 9: Casos de uso.

4.4.3. Características de los usuarios

En el sistema desarrollado, se identifican principalmente dos tipos de usuarios. Tal como está mostrado en la Ilustración 4 son el scraper y el docente.

Se ha recogido la información de los dos tipos de usuarios en la siguiente tabla con las filas de los significados siguientes:

- **Tipo de usuario:**
Usuario que formará parte del sistema.
- **Actividad:**
Tipo de acción que desarrolla dentro del sistema.

Tipo de usuario:	Docente.
Actividad:	Visualizar los resultados

Tabla 10: Usuario profesor/profesora

Tipo de usuario:	<i>Scraper</i>
Actividad:	Extracción de actividad del grupo

Tabla 11: Usuario Scraper

La sencillez de las interfaces permite a los usuarios utilizar el sistema de manera fácil. Esto quiere decir que no necesitan conocimientos previos sobre sistemas similares.

4.4.4. Restricciones

El sistema tiene varios tipos de restricciones, a continuación se mencionarán algunos:

- El profesorado tiene que disponer de una cuenta de Facebook.
- El docente podrá acceder al sistema con cualquier tipo de dispositivo que disponga de un navegador.
- Tener acceso a Internet.
- El profesor/profesora debe ser el administrador del grupo.

4.4.5. Precondiciones y dependencias

Una precondición elemental para el buen funcionamiento del sistema, es que para obtener los resultados de un grupo necesita tener un contenido previo. El grupo debe tener actividad, es decir, tener *posts* comentados para ser analizados junto con comentarios respuestas en caso que los haya, ya que con una mayor cantidad de participaciones en el grupo se obtendrá un resultado más representativo. Es decir, a mayor cantidad de comentarios y acciones sobre los *posts*, mejor certeza a la hora de analizar el estado emocional del grupo.

4.4.6. Evolución previsible del sistema

Algunas de las posibles ampliaciones del sistema podrían ser las detalladas a continuación:

- Poder seleccionar una fecha concreta para que se muestren los resultados en ese rango.
- Realizar comparativas entre las gráficas de diferentes grupos de asignaturas parecidas.

- Notificación al profesor/profesora docente cuando haya cambios en el grupo.
- Lista de los alumnos más activos del grupo.
- Sugerencias de páginas con contenido similar al tratado en el grupo.
- Traza de la evolución de cada alumno en el grupo desde que se ha unido hasta la fecha dada.

4.5. Requisitos específicos

En este apartado hemos recogido los diferentes requisitos del sistema en forma de tabla para así obtener una visión más entendible a la hora de describir cada requisito. El significado de las filas se presenta a continuación:

- **Número de requisito:**
Referencia con la que se ha identificado el requisito, hemos usado la identificación RFXX.
- **Nombre del requisito:**
Nombre asignado al requisito.
- **Tipo:**
Se distingue entre requisito de interfaz, funciona u otros.
- **Descripción:**
Breve explicación del requisito.
- **Prioridad:**
Se clasifica en alta, media y baja.
- **Característica:**
En el caso de que referencie alguna de las características.

4.5.1. Requisitos comunes de las interfaces

Nuero de requisito	RF01		
Nombre del requisito	Mostrar los grupos.		
Tipo	<input checked="" type="checkbox"/> Requisito de interfaz	<input type="checkbox"/> Requisito funcional	<input type="checkbox"/> Otros requisitos
Descripción	Visualizar los diferentes grupos del usuario autenticado para analizarlos.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Características	CA01		

Tabla 12: Requisitos funcional RF01

Nuero de requisito	RF02		
Nombre del requisito	Visualización de navegación.		
Tipo	<input checked="" type="checkbox"/> Requisito de interfaz	<input type="checkbox"/> Requisito funcional	<input type="checkbox"/> Otros requisitos
Descripción	Vistas claras de la interfaz y operaciones a realizar a la vista del usuario.		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Características	CA06		

Tabla 13: Requisitos de interfaz RF02

4.5.2. Requisitos funcionales

Nuero de requisito	RF03		
Nombre del requisito	Inicio de sesión con Facebook.		
Tipo	<input type="checkbox"/> Requisito de interfaz	<input checked="" type="checkbox"/> Requisito funcional	<input type="checkbox"/> Otros requisitos
Descripción	El profesorado se autentifica en la web con su perfil de Facebook para poder visualizar los grupos que queremos analizar.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Características	CA05		

Tabla 14: Requisitos de interfaz RF03

Nuero de requisito	RF04		
Nombre del requisito	Recuperación de los datos para mostrar los resultados.		
Tipo	<input type="checkbox"/> Requisito de interfaz	<input checked="" type="checkbox"/> Requisito funcional	<input type="checkbox"/> Otros requisitos
Descripción	Mediante la API de Facebook extraer los datos del grupo que se ha seleccionado previamente y hacer el análisis correspondiente para mostrar los datos, guardando estos en la BD.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Características	CA02, CA08		

Tabla 15: Requisitos de interfaz RF04

4.5.3. Otros requisitos

Nuero de requisito	RF05		
Nombre del requisito	Cumplimiento de la normativa.		
Tipo	<input type="checkbox"/> Requisito de interfaz	<input type="checkbox"/> Requisito funcional	<input checked="" type="checkbox"/> Otros requisitos
Descripción	Para un correcto funcionamiento del sistema hay que respetar en todo momento las políticas de la red social Facebook.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Características	CA05		

Tabla 16: Requisitos de interfaz RF05

Casos de Uso	Característica	Requisito
CA01	CA05	RF03, RF05
CA02	CA01, CA09, CA06	RF02
CA03	CA07, CA10	RF01
CA04	CA03, CA04	RF04
CA05	CA02, CA08	RF04

Tabla 17: Tabla Requisitos Generales

4.6. Conclusión

En este apartado se ha realizado una especificación de requisitos de manera formal utilizando el estándar **IEEE 830**. Se ha identificado al personal involucrado en el proyecto que son los cotutores, los alumnos que participaron en el grupo de FB y el desarrollador del sistema. Los cinco casos de uso junto con las características asignadas del apartado síntesis 2.4 y junto con las restricciones son necesarios para la creación de **Analisticsgroup**.

Por último, se han asignado los requisitos específicos a las características y a su vez a los casos de uso correspondientes, tal como se puede observar en la tabla 17.

5. Diseño

5.1. Introducción

A partir de la especificación de requisitos del apartado anterior, se ha hecho el diseño del sistema usando descripciones técnicas estándares como UML, diagramas de secuencias o similares.

5.2. Problemas al recoger la información

Podríamos preguntar a los propios alumnos sobre el estado de ánimo en el que se encuentran, recolectar toda esa información y tratarla para averiguar el estado de ánimo de un grupo docente. Pero esta es una forma forzada de medición, ya que el alumno es consciente del estado de ánimo que va a indicar o solamente expresaría como esta su estado de ánimo en el momento concreto. De esta manera no sabríamos con certeza si realmente es el estado emocional que corresponde al grupo.

5.3. Especificación conceptual

Para mostrar una visión de cómo funciona el sistema de forma completa, la Ilustración 5 nos muestra un esquema de funcionamiento.

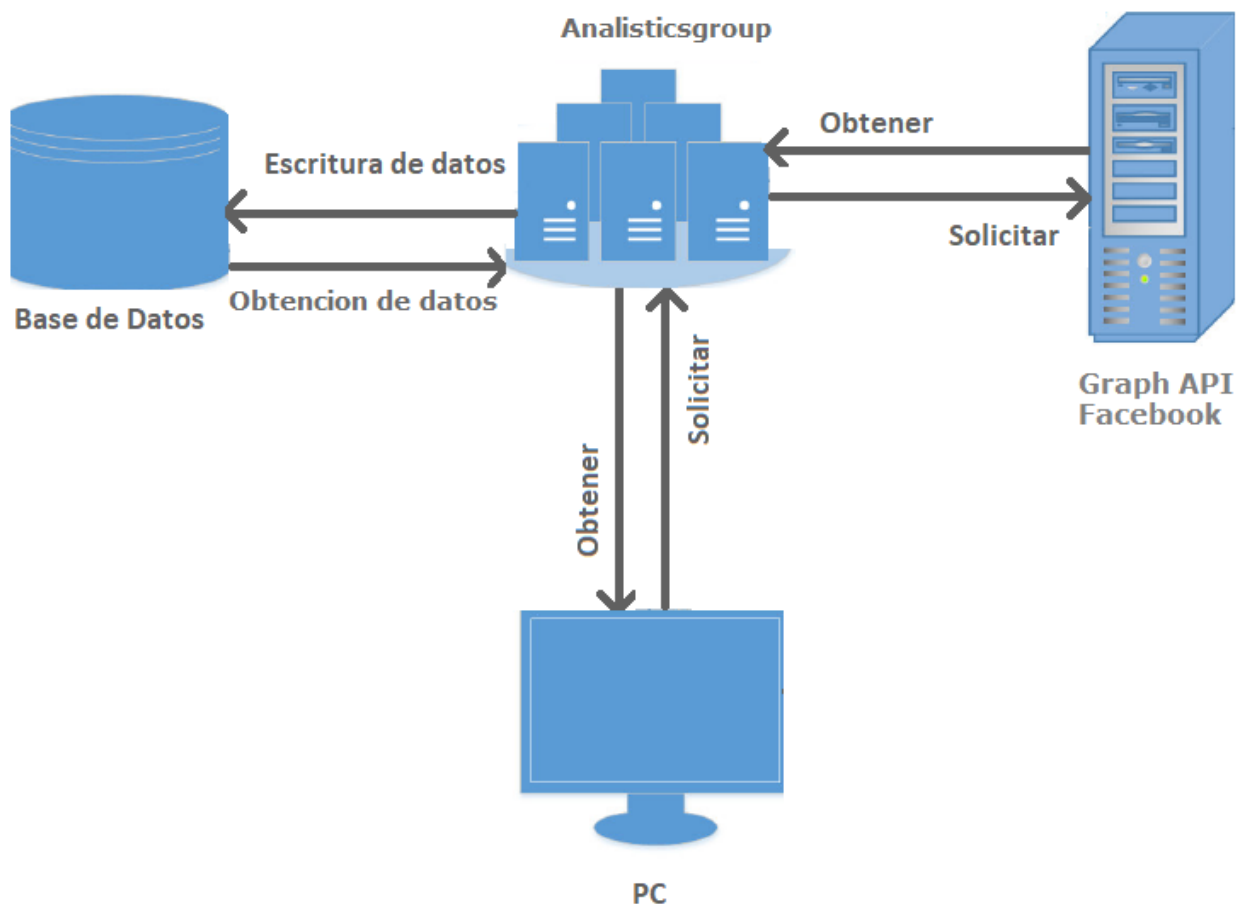


Ilustración 5: Especificación conceptual del proyecto.

El usuario, mediante el dispositivo "PC" solicita una autenticación al "Extractor de datos". Éste se encarga de consulta al "Graph API Facebook" si el usuario existe y le solicita los grupos de los cuales es administrador. Seguidamente, al obtener esta información, el "Extractor de datos" guarda en la "Base de datos" la información relevante. El "PC" obtiene la información de todos los grupos que puede analizar y así mismo consultar en cualquier momento de la base de datos.

5.4. Arquitectura de capas

Este apartado consta de tres subapartados, cada uno de los cuales describe cada capa implementada en el sistema. En primer lugar, se habla de la capa de persistencia o capa de datos, posteriormente de la capa de negocio o lógica y finalmente de la capa de presentación.

5.4.1. Capa de persistencia

Por lo que se refiere a la base de datos, como ya se ha descrito en el apartado 3.2., utilizaremos **MongoDB**, un tipo de base de datos no relacional. Es decir, la información de esta BD tiene una estructura de datos basada en documentos similares a **JSON** con un esquema dinámico.

A continuación, se representará con la Ilustración 6 el contenido de diseño de la BD de nuestro sistema.

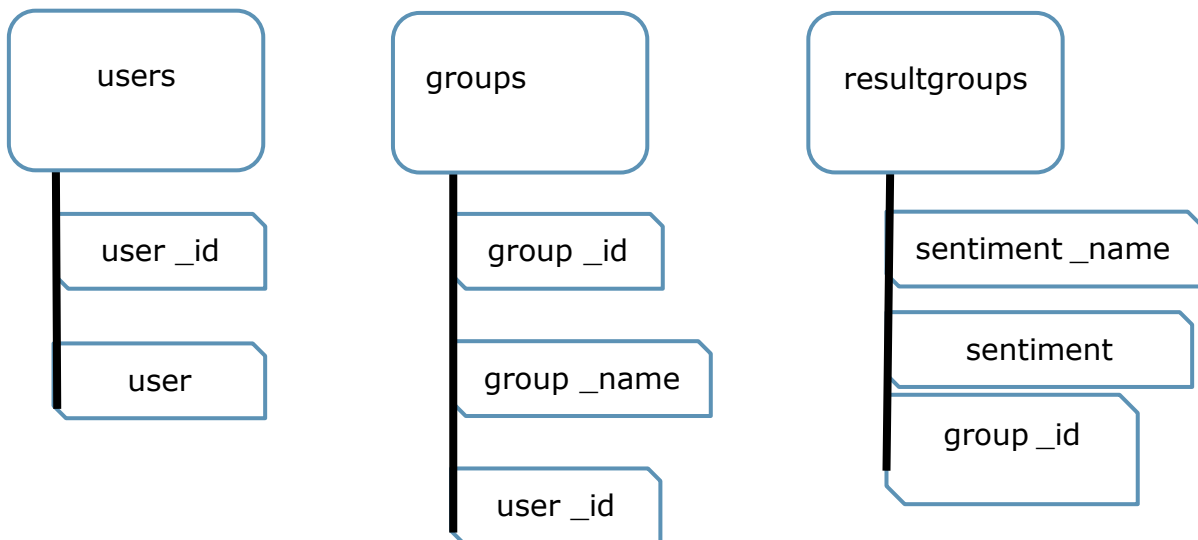


Ilustración 6: Diagrama BD.

- **Colecciones (Collections): [users, groups, dictionary]**
Son equivalentes a tablas en una base de datos relacional. Estas guardan información relacionada con la colección.

- **Clave – Valor: [user _id, user _name ...]**
Este funcionamiento podría interpretarse como equivalente a las columnas de una tabla relacional. Son las claves [user _id, user _name...], accediendo a ellas tenemos el contenido de esta clave. Sin embargo, en vez de guardar un único dato se pueden conservar varios. Como ya hemos comentado antes, tiene el funcionamiento de JSON dinámico.
- **Relaciones:**
Para relacionar diferentes colecciones entre ellas y así poder realizar diferentes consultas o modificaciones se usan identificadores, y un identificador de una colección puede estar dentro de otra. En el diagrama anterior podemos ver cómo [user _id] aparece en la colección [groups].

Una vez vistos los valores básicos, vamos a explicar el contenido de las colecciones. Pero antes de ello hay que comentar que al insertar cualquier contenido en una colección de MongoDB éste autogenera un id propio único y podríamos usarlo sin problema en nuestro sistema. Pero en nuestro caso se ha utilizado un id aparte, ya que nosotros extraemos ese id FB y es un identificador único.

- **(users) User _id:**
El id del usuario de Facebook, por lo que es único.
- **(users) User _name:**
Aquí se almacenará el nombre del usuario, que será el nombre del profesor autenticado en nuestro sistema.
- **(groups) Group _id:**
Este es el id del grupo de FB, ya que un usuario puede administrar varios grupos.
- **(groups) Group _name:**
Nombre del grupo extraído gracias a la API de Facebook.
- **(groups) user _id:**
Aquí indicaremos el grupo a qué usuario pertenece.
- **(resultgroup) sentiment _name:**
El nombre de la emoción que se mostrará al usuario. Se puede analizar el resultado de esta, tanto si es la más usada como no lo es.
- **(resultgroup) sentiment _result:**
Dentro hay un vector con datos. Si corresponde a un sentimiento tiene valor de 1 y si no valor de 0.
- **(resultgroup) group _id:**
Aquí se guardará el id del grupo al que pertenecen las palabras.

5.4.2. Capa lógica o de negocio

En este apartado, para ver gráficamente el desarrollo de una lógica de la aplicación, se mostrará un diagrama de secuencias para cada uno de los casos de uso del apartado 4.4.2. Así se mostrará en primer lugar el diagrama de secuencias correspondiente a la petición de inicio de sesión en FB y la solicitud de permisos en perfil del usuario.

A continuación, en la Ilustración 7 se observa cómo el usuario llama al extractor con los datos para iniciar sesión en Analyticsgroup. Ahora, el extractor realiza una llamada a Facebook pasando el usuario y la contraseña para poder autentificar al usuario, y la respuesta de la API de FB es la petición de permisos de la cuenta del usuario.

Al conceder los permisos al usuario la autenticación es aceptada y le informa entrando en el sistema. De lo contrario devolvería una respuesta negativa al extractor, el cual informaría al usuario de que algo ha ido mal.

Una vez autenticado el usuario puede cerrar sesión, en este caso el extractor informa a FB de que el usuario hizo *logout* y en el siguiente inicio de sesión, la API de FB ya tendrá guardada la configuración de los permisos del usuario, por lo que no se solicitarán de nuevo.

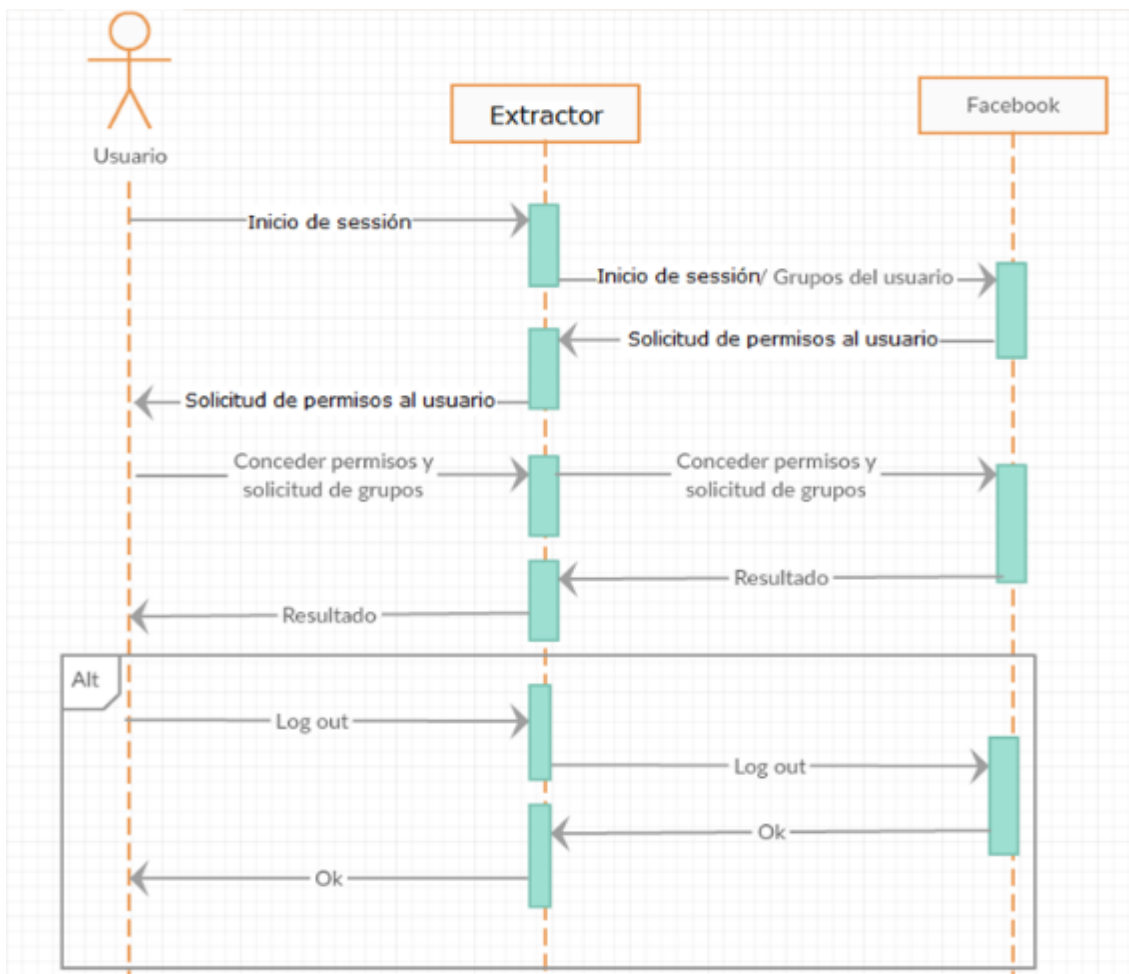


Ilustración 7: CU01 Identificación del usuario.

En la siguiente Ilustración 8 se observa el diagrama de secuencias cuando un usuario se ha autenticado y pasa a explorar los grupos disponibles para luego analizar uno de ellos.

Para ello, el usuario solicita al autenticarse los grupos que están disponibles y el extractor es el encargado de realizar la petición a FB y recoger la respuesta.

Una vez recibida la respuesta de FB, el extractor comprueba que los grupos no existen en la base de datos y guarda la información recibida, seguidamente le devuelve al usuario los resultados, es decir, los grupos disponibles para analizar.

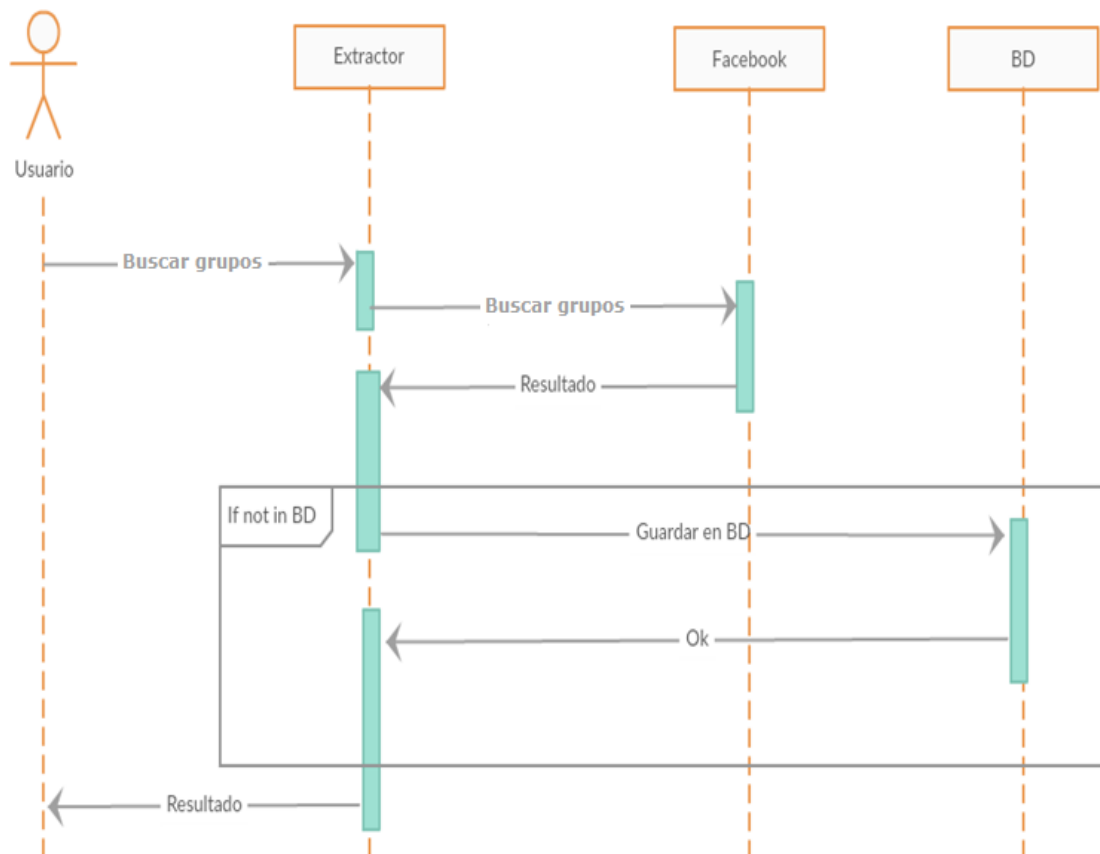


Ilustración 8: CU02 Monitorización de actividad en FB.

En la Ilustración 9 tenemos un grupo seleccionado para analizar por lo que manda una petición al extractor (buscar datos del grupo elegido). El extractor es encargado de recuperar los datos en la base de datos.

Con los datos (palabras) recogidos procede a conectarse al diccionario de palabras que incluye también la puntuación que se le asigna a cada palabra

según la emoción. Seguidamente, el extractor recibire las puntuaciones del diccionario y realiza el cálculo de las puntuaciones del contador global. Por último, devuelve los datos al usuario.

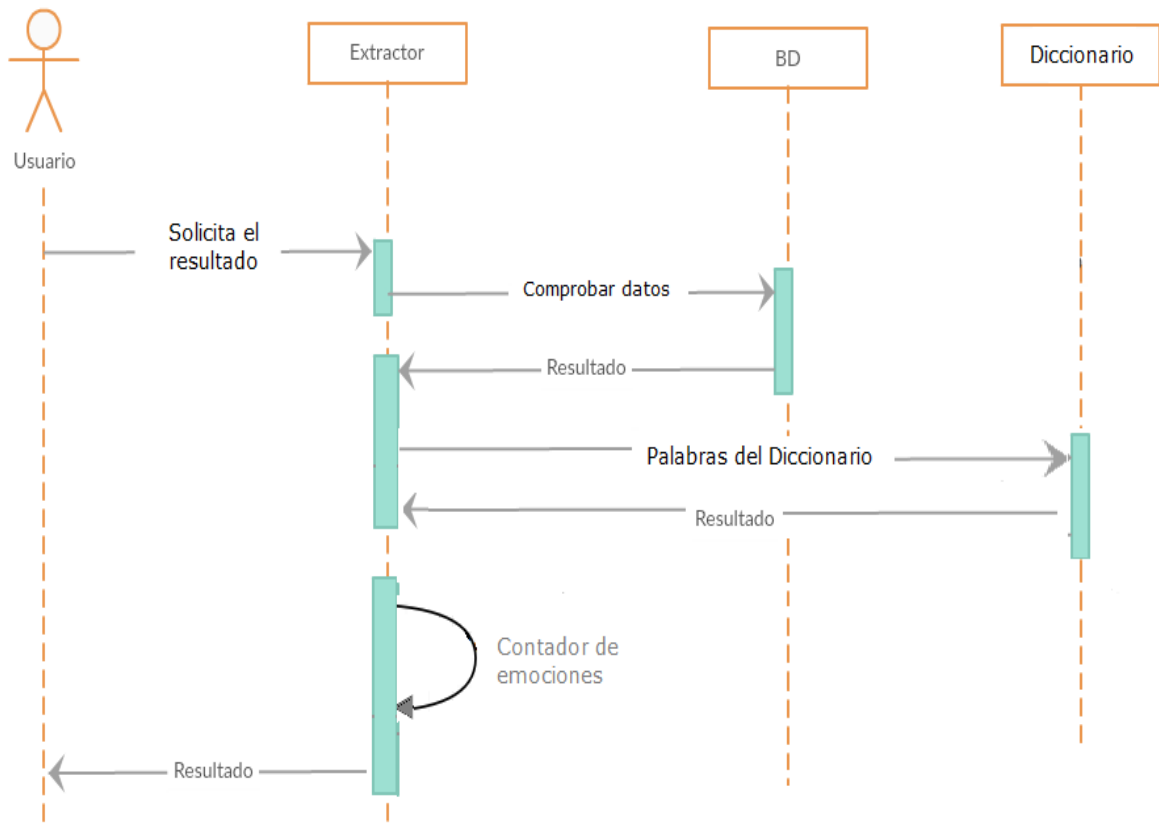


Ilustración 9: CU03 Cálculo de sentimientos.

Seguidamente, en la Ilustración 10 se describe el diagrama de secuencias del cálculo de impacto de las publicaciones. El usuario solicita al extractor los resultados del cálculo de impacto de las publicaciones.

El extractor consulta a la base de datos y al recibir el resultado realiza el cálculo del número de usuarios que han reaccionado a una publicación y se lo devuelve al usuario.

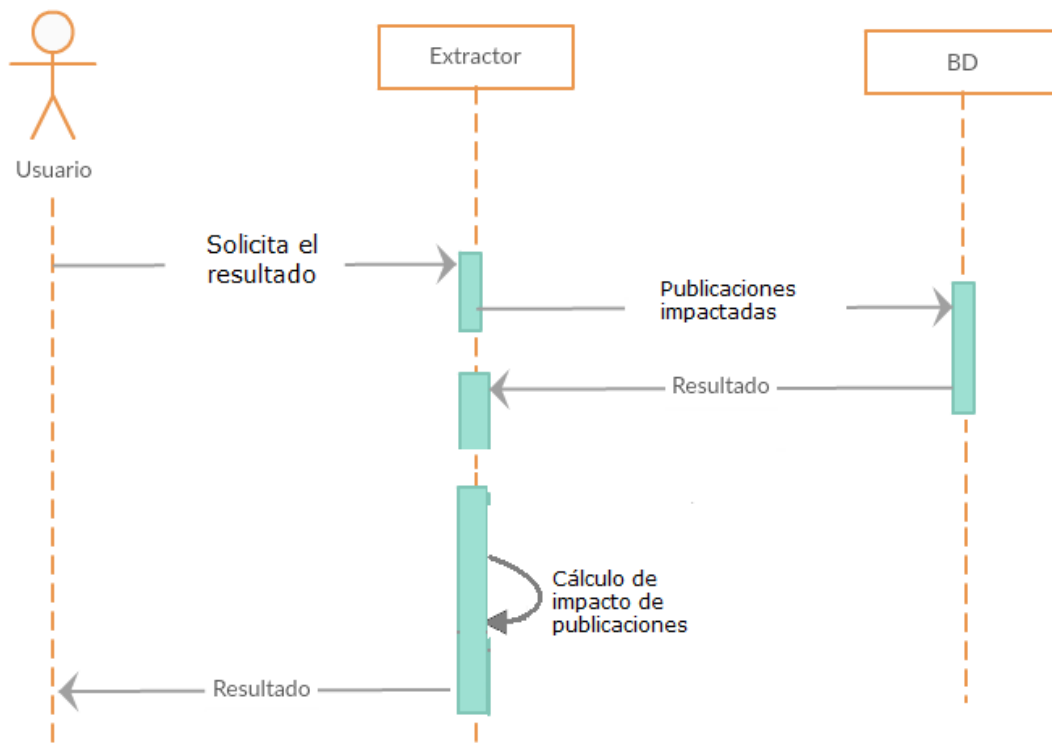


Ilustración 10: CU04 Cálculo de impacto de las publicaciones.

Por último, en la Ilustración 11 se ve cómo el extractor, al recibir la solicitud del usuario, consulta la base de datos para obtener la información disponible y seguidamente realiza el renderizado para formar los resultados totales y darselos al usuario.

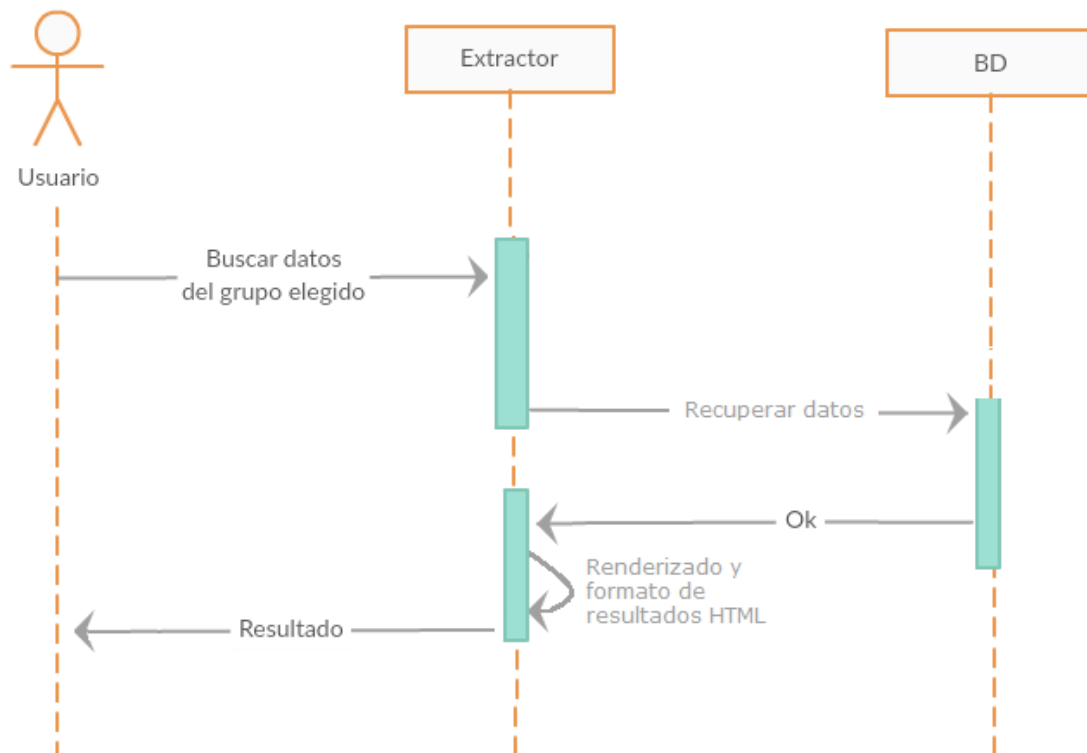


Ilustración 11: CU05 Visualización de resultados.

5.4.3. Capa de presentación

En este apartado se describe una serie de *mockups* acompañados de las correspondientes explicaciones con tal de determinar cómo se ha creado la capa de presentación. Se explica la aplicación web y se muestra el funcionamiento completo de las interfaces del usuario por pasos.

Se ha creado con la interfaz más simple e intuitiva. Para ello, en la Ilustración 12 se ha diseñado solamente un botón que tiene como objetivo el inicio de sesión con Facebook. La API de Facebook pide permisos a la cuenta del usuario para poder acceder a sus datos de los grupos. Al centrarnos en esta red social, únicamente hemos omitido las posibilidades de autenticación o registro manual. Esto se debe a que todos los datos a analizar se extraen de esta misma red social.

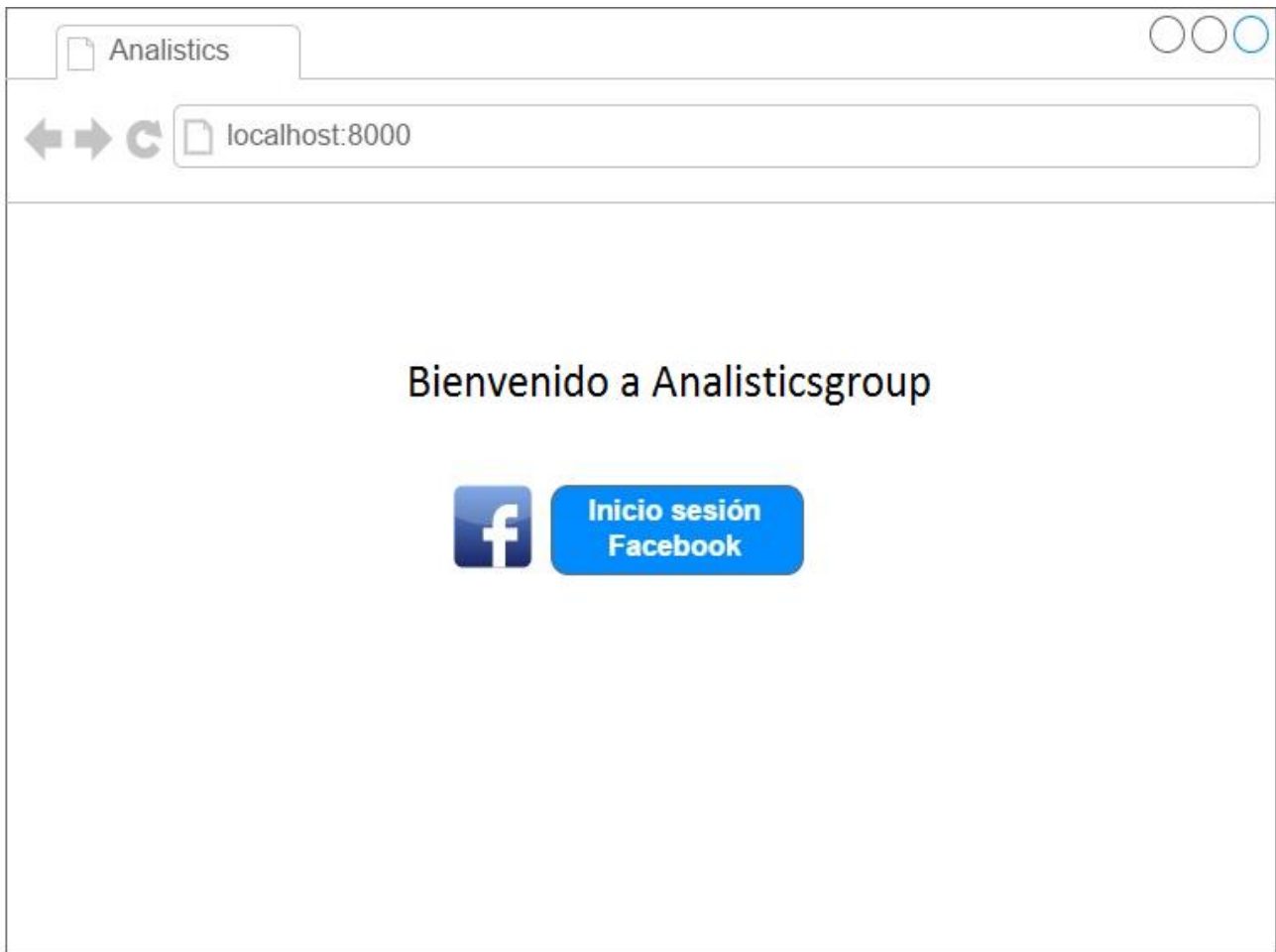


Ilustración 12: Mockup de inicio de sesión.

Una vez pasada la ventana anterior, se le mostrará al usuario todos los grupos de los cuales es administrador.

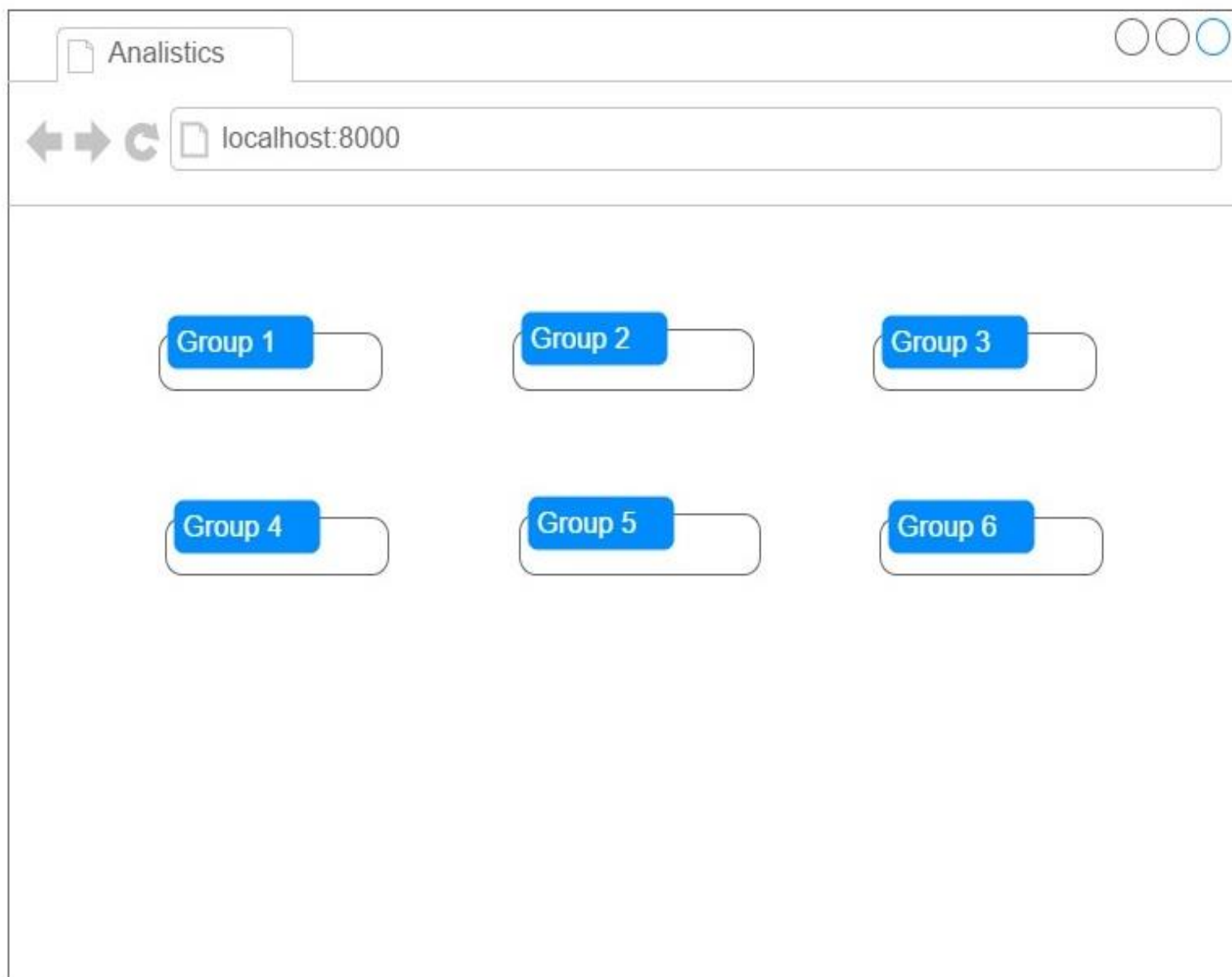


Ilustración 13: Grupos disponibles

A continuación, seleccionando un grupo cualquiera, el sistema entra en el modo de extracción de datos. Y pasa a procesar tal como se ha comentado en Ilustración 8.

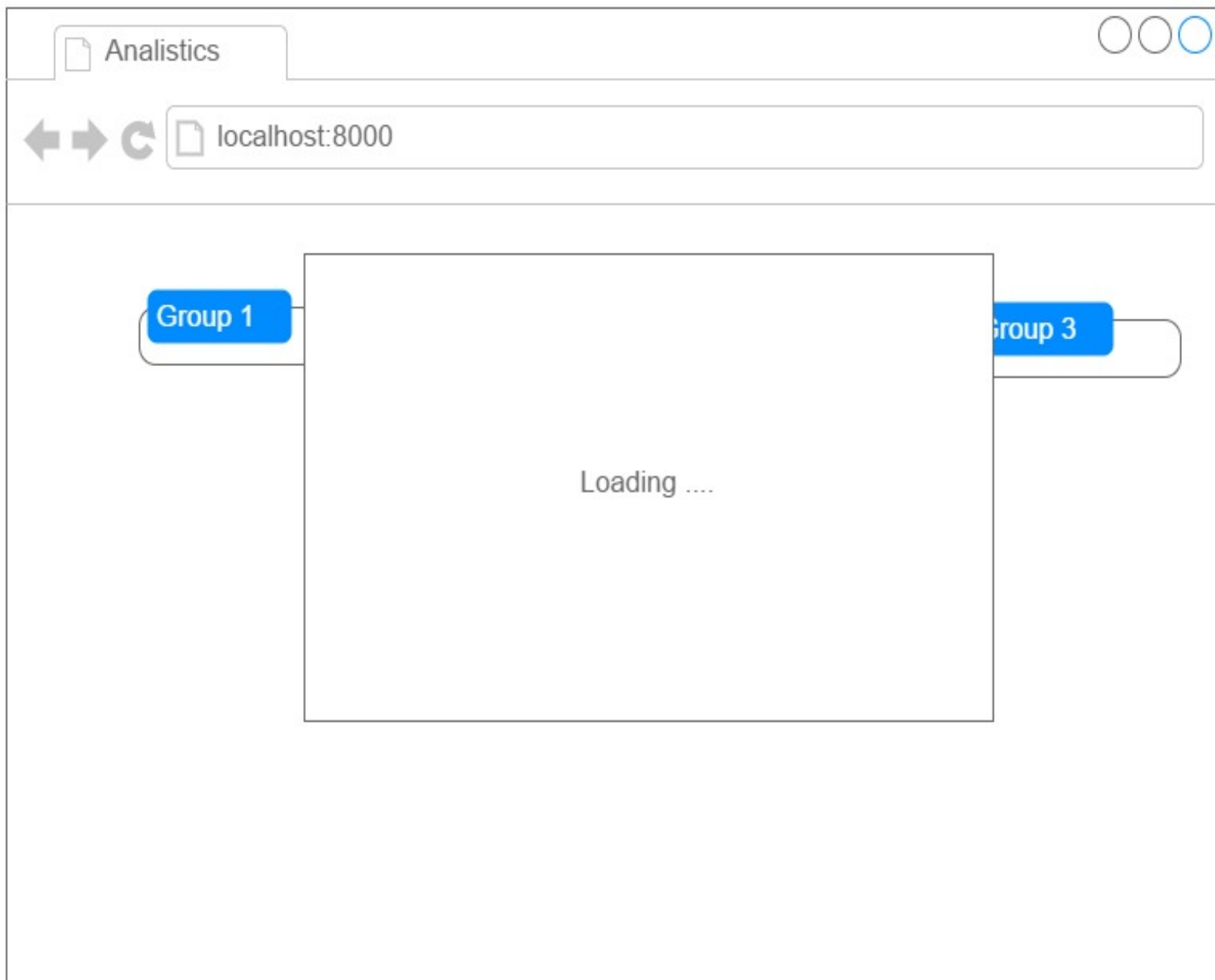


Ilustración 14: Pantalla de carga para extracción.

En la Ilustración 15 podemos observar las diferentes gráficas con los resultados. Es decir, en esta interfaz ya hemos recolectado todos los datos y hecho las comparaciones y evaluaciones necesarias. Esto quiere decir que el usuario, con solamente un vistazo, puede ver y saber cómo está evolucionado el grupo.

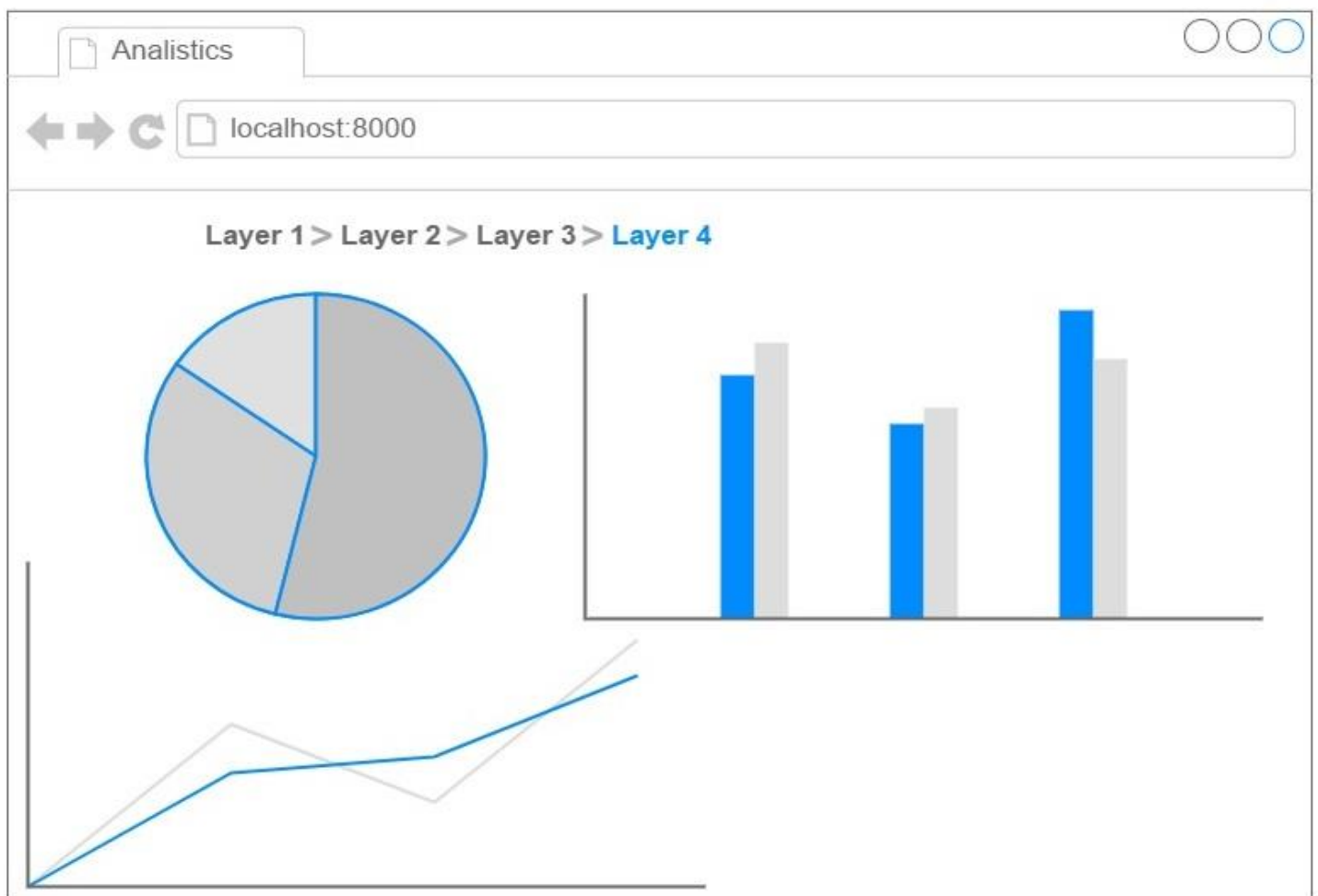


Ilustración 15: Resultados análisis grupo.

5.5. Conclusión

En este apartado hemos hablado del diseño que tiene nuestra aplicación, estableciendo un esquema general con la intención de mostrar la funcionalidad completa. Posteriormente, se ha realizado el diseño del sistema. Una vez determinado el diseño que tendrá el sistema se procederá a la implementación, teniendo en cuenta que el diseño anterior es totalmente aproximado. Es decir, no hay que ceñirse exactamente a los *mockups*.

6. Implementación y evaluación

6.1. Entorno de desarrollo

6.1.1. Entorno hardware

Para desarrollar la aplicación se ha utilizado un único ordenador. A continuación, describimos sus características.

PC portátil:

- Procesador Intel Core i7 2.20 GHz
- 8 GB de memoria RAM
- Sistema operativo Ubuntu 16.04 LTS

6.2. Instalación y configuración

Como ya se ha comentado en apartados anteriores como el 3.1., el “**framework de Django**” funciona con Python.

Por ello, a continuación se explicará la instalación de Django en el entorno hardware comentado anteriormente. El entorno gráfico donde se instala todo para la puesta en marcha del sistema será mediante el terminal (ventana de comandos) del sistema operativo.

6.2.1. Configuración de *settings*

```
INSTALLED_APPS = [  
    'social_django',  
    'mysite.core'.  
]  
ROOT_URLCONF = 'mysite.urls'
```

Código 1: Configuración de settings.

Se puede observar que hemos agregado una **APP** llamada **social_django** que es proporcionada por Facebook, nos servirá para la autenticación de usuarios en la red social.

```
LOGIN_URL = 'login'  
LOGOUT_URL = 'logout'  
LOGIN_REDIRECT_URL = 'home'
```

Código 2: Configuración URL base.

6.3. Comunicación entre Django y MongoDB

Esto se ha realizado gracias a la librería **Pymongo** comentada en 6.2.3. que hemos instalado. A continuación, con la librería en marcha, se ha importado la base de datos:

```
from pymongo import MongoClient
client = MongoClient('localhost', 7017)
db = client['analistics']
resultado = db.users.find_one ({condicion/clausula})
```

Código 3: Consulta a MongoDB.

Todos los resultados que devuelve **MongoDB** son documentos de JSON como ya se mencionó anteriormente en el apartado 3.2.2.

6.3.1. Configuración de URL's

En el fichero (**/mysite/urls.py**) se encuentran todas las rutas a las que vamos a acceder mediante el navegador y también nos indican qué función se debe ejecutar según la **URL**.

Primero se indica dónde se encuentran los ficheros con los métodos que queremos que se ejecuten, una vez se haya accedido a la URL.

Importamos las diferentes vistas que tengamos, en este caso solo se ha importado **dango.contrib.auth** y también el **mysite.core**.

```
from django.conf.urls import url, include
from django.contrib import admin
from django.contrib.auth import views as auth_views
from mysite.core import views as core_views
urlpatterns = [
    url(r'^$', core_views.home, name='home'),
    url(r'^login/$', auth_views.login, name='login'),
    url(r'^logout/$', auth_views.logout, name='logout'),
    url(r'^settings/$', core_views.settings, name='settings'),
    url(r'^datag/$', core_views.datag, name='datag'),
    url(r'^oauth/', include('social_django.urls', namespace='social')),
    url(r'^admin/', admin.site.urls),
]
```

Código 4: Redirección de accesos a URL's.

6.3.2. Configuración de Views en Django

En el archivo (`/mysite/core/views.py`) se encuentran las vistas y la extracción de datos que haremos sobre Facebook, es decir, el funcionamiento del programa cuando el usuario se autentifique y lo utilice.

En este fichero, lo primero que se realiza es la carga de las diferentes expresiones emocionales con su puntuación. Esto viene del diccionario que se ha comentado anteriormente. Lo primero que se ha realizado es cargar dicho diccionario en el sistema, concretamente en un **ArrayHash**. De esta manera, si queremos consultar cualquier palabra, el coste de la consulta será mínimo, ya que la palabra será el identificador de este vector.

```
#Read de xlsx file, save in arrayhash and the puntuation in the table.
for row in ws.iter_rows(min_row=2, max_col=11, max_row=14183):
    diccionario[row[0].value] = []
    for cell in row:
        if not cell.value in diccionario:
            diccionario[row[0].value].append(cell.value)
```

Código 5: Carga del diccionario.

Siguiendo el flujo de la aplicación, una vez el usuario se haya identificado, nos redireccionará a la página de todos los grupos disponibles. El usuario tiene que ser **Administrador**, ya que sólo se pueden gestionar este tipo de grupos. Esto se debe a los permisos de Facebook que tenga el usuario y las posibles restricciones y la política de seguridad.

En el Código 6 se puede observar el uso obligatorio de **login** en nuestro sistema `"@login_required"`. De esto se encarga **social_django** comentado anteriormente (**Settings**).

```
@login_required
def settings(request):
    user = request.user
    try:
        facebook_login = user.social_auth.get(provider='facebook')
    except UserSocialAuth.DoesNotExist:
        facebook_login = None
    #Get tokens and id necesry from facebook user
    graph = facebook.GraphAPI(access_token)
    user_id = graph.get_object(user_id)
```

Código 6: Configuración login.

En el Código 7 se comprueba en la base de datos si el usuario existe, si no lo inserta en la BD.

```
#Check if user are yet in DataBase
result_user = db.users.find_one({'id':user_id})
#Insert if not exist
if str(result_user) == 'None':
    db.users.insert(user_id)
groups = graph.get_connections(user_id, 'groups')
can_disconnect = (user.social_auth.count() > 1 or
user.has_usable_password())
```

Código 7: Comprobación del usuario en BD.

A continuación (ver Código 8) se ha añadido el grupo al usuario perteneciente a la BD.

```
groups_avaliables = {}
#Add user_id wich groups is had
for value in groups['data']:
    result = db.groups.find_one({'id':value['id']})
    groups_avaliables[value['id']] = value['name']
    if str(result) == 'None':
        value['user_id'] = user
        db.groups.insert(value)
```

Código 8: Almacenado el grupo del usuario.

Por último, antes de terminar, Django se encarga de renderizar un *template* de HTML pasándole todos los datos y muestra por la pantalla de un navegador el resultado.

```
return render(request, 'core/settings.html', {
    'groups_avaliables': groups_avaliables,
    'facebook_login': facebook_login,
    'can_disconnect': can_disconnect
})
```

Código 9: Renderizado de HTML.

Para la extracción del contenido de un grupo (ver Código 10) se llama a la **API** de FB pasando las peticiones por POST y recuperando por GET. Se recorren todas las páginas de información y según la estructura de JSON que devuelve FB extraemos el contenido (las palabras).

```
#Start to GET all comments from posts with diferents levels of Facebook groups.
datass = graph.get_connections(id_group,
                              '?fields=feed{comments{message,from,created_time}}')
datass = datass.get('feed')
while(True):
    try:
        for dat in datass.get('data'):
            datt = dat.get('comments')
            if datt:
                for allcoments in datt.get('data'):
                    id_user = allcoments.get('from').get('id')
                    created_date = allcoments['created_time']
                    user_message = allcoments.get('message').split()
                    if id_user in message:
                        d = message.get(id_user)[0] + user_message
                        message[id_user] = [list(set(d)), created_date]
                    else:
                        message[id_user] = [user_message, created_date]
            if datass:
                datass = requests.get(datass['paging']['next']).json()
    except KeyError:
        break
```

Código 10: Extracción de contenido.

En el Código 11 se recorren todas las palabras disponibles y se buscan en el diccionario las palabras existentes, todas aquellas encontradas se insertan en la base de datos con el identificador del usuario, el grupo al que pertenecen, la fecha de creación y la palabra.

En el vector **sentimientos** actualizaremos el contador de cada sentimiento, y así mantendremos la cuenta de apariciones de palabras del diccionario.

```
#FINALY ARRAYHASH WITH ALL COMMENTS AT USERS
for key, valu in mesage.iteritems():
for pal in valu[0]:
    if pal in diccionario:
        db.datagroup.update_one(
            {
                "id_user" : key,
                "id_group" : id_group,
                "date_created" : valu[1],
                "content" : pal
            }, {"$set" : {"content" : pal}},
            upsert=True
        )
        sentimientos = map(operator.add,
                            sentimientos,
                            diccionario.get(pal))
```

Código 11: Bucle que calcula sentimientos.

A continuación (ver Código 12) se calcula el porcentaje que corresponde a cada sentimiento y se guarda el resultado en la base de datos.

```
allfound = sum(sentimientos)
for typ, punt in zip(typeemoji,sentimientos):
    finaldata[typ] = round(punt * 100/float(allfound), 2)
db.resultsgroups.update_one(
    {
        "id_group" : id_group
    },
    {
        "$set" : {"id_group" : id_group,
                 "finaldata": finaldata,
                 "date": datetime.datetime.now().date().isoformat()}
    },upsert=True
)
```

Código 12: Almacenamiento de resultado.

Si la base de datos está actualizada con los datos de fecha del día actual se ejecuta el Código 13, es decir, el *e/se* que recupera la información de la base de datos y no se realiza de nuevo el Código 10.


```
else:
    resultdat = db.resultsgroups.find({'id_group': id_group})
    for res in resultdat:
        finaldata = res['finaldata']
```

Código 13: Recuperación de datos de BD.

Seguidamente en el Código 14 se realiza el cálculo de impacto de las publicaciones.

```
usersInGroup = db.datagroup.distinct("id_user");
for var1 in usersInGroup:
    nameUser = graph.get_object(id=var1)
    adat = db.datagroup.count({"id_user": var1,
                              "id_group": id_group})
    usersPart[adat] = nameUser['name']
usersPart =
    collections.OrderedDict(reversed(sorted(usersPart.items())))
return render(request, 'core/datag.html', {
'ladate': finaldata,
'usersPart' : usersPart
})
```

Código 14: Cálculo de impacto de las publicaciones.

6.3.3. Funcionamiento de *Templates* en Django

En las funciones anteriores, si nos fijamos al final de código, en los dos casos teníamos un **return** en el apartado 6.3.2.

Esto se debe que en la ruta (**/mysite/core/templates/core/**) se encuentran los ficheros **HTML** que van a ser renderizados y creados por Django, agregando los datos que se pasan por el **"return"**.

A continuación, se explica cómo se muestran los datos que son recibidos por Django. Un ejemplo claro es a la hora de extraer los grupos disponibles de un usuario, para elegir cuál de ellos analizar.

```

{% if facebook_login %}
{% for key, value in groups_avaliables.items %}
<div class="col-lg-3 col-md-6">
  <div class="panel panel-primary">
    <div class="panel-heading">
      <div class="row">
        <div class="col-xs-3">
          <i class="fa fa-comments fa-5x"></i>
        </div>
        <div class="col-xs-9 text-right">
          <div>{{ value }}</div>
        </div>
      </div>
    </div>
    <form action="{% url 'datag' %}"
          method="post"
          style="margin:0;"> {% csrf_token %}
    <input hidden id="id_group"
           type="number"
           name="numero"
           value="{{ key }}" >
    <button style="width:100%"
           type="submit"
           data-toggle="modal"
           data-target="#myModal">
    <div class="panel-footer">
      <span class="pull-left">Ver Detalles</span>
      <span class="pull-right">
        <i class="fa fa-arrow-circle-right"></i>
      </span><div class="clearfix"></div>
    </div>
    </button>
  </form>
</div>
{% endfor %}
{% endif %}

```

Código 15: Visualización de los resultados.

Se puede observar que mediante las etiquetas **{% etiquetas %}** los datos son recogidos del **"return"** del apartado anterior (6.3.2.) para ser procesados y mostrados al usuario. Como se observa al principio del Código 14, se puede insertar condiciones, bucles, etc...

Otro detalle a comentar es la etiqueta **{% csrf_token %}**, que es el token de protección de formularios para evitar que un usuario malintencionado intente hacer que los usuarios envíen información que no quieran enviar sin saberlo.

6.4. **Conclusión**

Se han presentado las características de la maquina física en la cual se ha desarrollado el sistema.

En el apartado Implementación y evaluación se ha explicado la configuración de Django en el fichero principal **Settings**. Seguido de ello se ha mostrado la implementación de ficheros como **URL**, **Views** y **Templates**, siendo estos últimos para mostrar el resultado de la implementación del sistema.

7. Pruebas

Tal como se ha indicado en el apartado 1.3., la metodología utilizada es el modelo en espiral. Por lo que con cada una de las cinco versiones (mediante el uso de GitHub) de la aplicación se hacían pruebas comprobando que se cumplía cada requisito funcional implementado.

7.1. Behat

“Behat es un *framework* de desarrollo basado en comportamiento. Es una herramienta para apoyarle en la entrega de **software de calidad** a través de la comunicación continua, el descubrimiento deliberado y la automatización de pruebas.”

(<http://behat.org/> , Behat)

Mink es un controlador / emulador de navegador de código abierto para aplicaciones web, escrito en PHP 5.3. Éste, junto a **Behat** proporcionan la simulación de ejecutar nuestro sistema y realizar las pruebas de test. Y mediante **Selenium** podremos ver cómo pasan los test automáticamente ejecutando el navegador y comprobando las funcionalidades.



Ilustración 16: Automatización Behat.

Estas pruebas se organizan por escenarios, dentro de los cuales asignamos diferentes funciones para que se realice el test automatizado.

El código se organiza por tabulaciones, y entre comillas indicamos los parámetros que se pasaran para analizar. Gracias a **Selenium** se buscan mediante **Xpath** los posibles id de los botones o textos que aparecen en la web para confirmar que ha llegado a dicho punto de la web.

```
Feature: Comprobar si se cumplen los requisitos funcionales.
```

```
Scenario: Autenticación and result of group.
```

```
Given I am on "www.localhost:8000/"
And I press "Continuar con Facebook"
When I fill in "#idUserinFb" with "andriyocio@gmail.com"
When I fill in "#idPassFb" with "123456789a"
And I press "#idAutenticación"
Then I should see "Grupos"
And I press "#idgroup1"
Then I should see "Estadísticas"
```

```
Scenario: Autenticación, navegación and logout.
```

```
Given I am on "www.localhost:8000/"
And I press "Continuar con Facebook"
When I fill in "#idUserinFb" with "andriyocio@gmail.com"
When I fill in "#idPassFb" with "123456789a"
And I press "#idAutenticación"
Then I should see "Grupos"
And I press "#idgroup2"
Then I should see "Estadísticas"
And I press "#allGroups"
Then I should see "Grupos"
And I press "#idgroup1"
Then I should see "Estadísticas"
And I press "#userMenu"
And I press "#logout"
Then I should see "Muchas gracias por usar Analisticsgroup"
```

```
2 escenarios (2 passed)
```

```
23 steps (23 passed)
```

Código 16: Resultado del test.

Los **Scenarios** (scenario: Autenticación and result of group) son ciclos de pruebas que están compuestas por **steps** es decir, etapas (Given I am on "www.localhost:8000/").

En el Código 16 se puede ver que entramos en la página (www.localhost:8000/), a continuación hacemos clic en el botón (Continuar con Facebook). Behat busca automáticamente por id el campo con el identificador (#idUserinFB) y asigna el texto del correo (andriyocio@gmail.com). Realiza el mismo procedimiento con (#idPassFB) con el texto de la contraseña (123456789a). Ahora se pulsa el botón (#idAutenticación) y estaremos logueados automáticamente. Comprobamos si podemos ver el título "Grupos" en la etapa (Then I should see "Grupos").

Con la línea (And I press "#idgroup1") entraremos en el primer grupo de los disponibles del usuario (si no tuviera grupos nos daría un fallo y dejaría de ejecutarse Behat). Si vemos el texto (Estadísticas) quiere decir que estamos en la pantalla de los resultados y el test ha finalizado correctamente.

En el segundo escenario podemos observar que realiza los mismos pasos solo que primero entra en (#idgroup2) un grupo diferente al anterior y comprueba que está en la pantalla de resultados.

A partir de ahora se realiza el test de la navegación del sistema mediante la etapa (And I press "#allGroups"). Si volvemos a la pantalla de los grupos comprobamos que realmente es la pantalla correcta (Then I should see "Estadísticas"). Ahora entramos en otro grupo para comprobar que no haya fallos en la navegación (And I press "#idgroup1") y observamos si estamos en la pantalla de resultados.

Ahora comprobaremos el cierre de sesión. Primero desplegaremos el menú del usuario (And I press "#userMenu"), seguidamente se hará clic en la opción cerrar sesión (And I press "#logout") y para comprobar si se ha realizado correctamente se comprueba que aparece el mensaje "Muchas gracias por usar Analyticsgroup" con la etapa (Then I should see "Muchas...").

Con esto habría concluido el test de los requisitos funcionales establecidos. Y si todo funcionó correctamente nos saldrá el mensaje de cuántos escenarios y las etapas de estos que han pasado el test. En el caso contrario, nos indicaría en qué etapa de qué escenario falla el test.

7.2. Conclusión

En este apartado se ha observado la gran ventaja de utilizar la tecnología **Behat**, ya que los test quedan automatizados. Cada vez que se ha añadido contenido nuevo, se ha comprobado la funcionalidad y accesibilidad con una ejecución de **Behat**, devolviendo el sencillo resultado de si se han pasado correctamente todos los "**Scenario**" o no.

8. Conclusiones

8.1. Conclusiones generales

Durante el desarrollo de nuestro sistema se han analizado diferentes sistemas y plataformas que ofrecen un análisis de páginas de FB, muros y grupos. Las ventajas como las publicaciones en Facebook de las propias plataformas y las desventajas de ser sistemas muy orientado al mundo de marketing fueron las que nos dieron una primera idea de cómo debería ser nuestro **Analisticsgroup**.

Mediante el análisis se han establecido las características principales tales como conocer el perfil de nuestro grupo en FB. Esto hace que de nuestro sistema destaque frente otros.

Una vez planteada una solución a los problemas que presentan los sistemas actuales, se ha tomado la decisión de crear un sistema propio y para ello se han utilizado nuevas tecnologías como como **Django** y **MongoDB**. Se han elegido bajo el criterio de autoaprendizaje y con la idea de que nos facilitara la creación del sistema por sus ventajas principales de rapidez, respuesta y seguridad.

Al tener claro la tecnología en la que se desarrollaría **Analisticsgroup**, se ha planteado como paso siguiente la estructura a seguir: una especificación de requisitos según el estándar IEEE 830.

También se identificaron a las personas que han influido durante la evolución del proyecto, como son los alumnos y el usuario docente. Seguidamente se han definido las funciones de nuestro sistema, como son los casos de uso, características y requisitos funcionales.

Uno de los apartados a destacar son las restricciones, que es importante conocer para el correcto funcionamiento del **Analisticsgroup**.

Teniendo claro el problema a resolver y la tecnología a utilizar, se ha realizado la puesta en marcha con la creación del diseño para el sistema, estableciendo un esquema general presentado con prototipados en **mockups**.

Estos **mockups** cogieron color y forma en la implementación y se ha desarrollado **Analisticsgroup**. Y para comprobar su correcto funcionamiento de forma automatizada, cada vez que teníamos una evolución o algún elemento ya creado, se hizo uso de **Behat**.

8.2. Desafíos

Al inicio del proyecto, en la realización del estudio de los diferentes sistemas similares, uno de los primeros problemas que surgió fue el de la gran cantidad de sistemas existentes. Con esto como primer obstáculo se tuvieron que filtrar las aplicaciones que tenían objetivos en común.

Determinar una estructura de los campos en la Base de datos ha sido una tarea costosa, ya que se había creado una estructura inicial que se ha ido modificando durante la evolución de proyecto, sea por la necesidad de guardar nuevos datos o el almacenamiento de estos por aplicaciones.

Todos estos aspectos han sido modificados respecto al planteamiento inicial del proyecto. Esto no quiere decir que son problemas, sino dificultades que ha influido en el aprendizaje y el desafío del desarrollo de éste.

Uno de los desafíos más grandes ha sido la clasificación de palabras, ya que la evaluación de cada palabra puede tener varios significados, dependiendo del contexto. Es así que una palabra puede tener diferentes sentimientos expresados a los que aparenta.

8.3. Aportaciones tecnológicas

La principal aportación es el sistema compuesto por Python, ya que la mayoría de las aplicaciones están creadas con los lenguajes que proporciona Facebook oficialmente.

La autenticación en **Analisticsgroup** es la autenticación en Facebook y no hay necesidad de crear cuentas adicionales.

Ninguno de los sistemas similares analiza palabras. Esto se debe a que están orientadas más a métricas más generales como el número de usuarios, número de comentarios etc...

Hay muy pocos sistemas que analizan los grupos exhaustivamente, ya que todos se centran en el marketing de las páginas de Facebook y no en los grupos.

Analisticsgroup ha sido creado con un propósito educativo para tener una aproximación del estado emocional de un grupo de Facebook. El sistema devuelve como resultado final una tabla con los porcentajes de cada emoción.

Otro enfoque que se ha realizado es la automatización de los test con **Behat**, **Mink** y **Selenium**.

8.4. **Posibles mejoras**

El siguiente nivel de este proyecto sería un análisis de las frases, es decir, asignar una puntuación positiva o negativa, y realizar el cálculo de los puntos para evaluar si corresponde a una emoción concreta.

Otra posible mejora es devolver por fecha los resultados, es decir, en un rango de fechas seleccionadas por el usuario docente mostrar los resultados. De esta forma se podría analizar el impacto que implican las fechas de exámenes, por ejemplo.

Agregar funcionalidad de gestión, para la publicación de contenido en el grupo de Facebook desde la plataforma de ***Analisticsgroup***.

9. Bibliografía

1. Poza Luján, Jose Luis, Calduch-Losa, Ángeles, Albors, Ana y otros. Propuesta de parámetros y caracterización de los grupos de las redes sociales orientados a la docencia universitaria: experiencia y resultados. *RED – Revista de educación a distancia*. Número 44. 2014-11-15. Grupo RED, Departamento de Psicología Evolutiva y de la Educación de la Universidad de Murcia. [Consulta: Marzo 2017]
<https://riunet.upv.es/bitstream/id/225494/Cabrera%3BD%C3%ADez-Somavilla%20-%20Proposal%20of%20Parameters%20and%20Characterization%20of%20Social%20Network%20Groups.pdf>
2. Wikipedia. *JSON*. [Consulta: Julio 2017]
<https://es.wikipedia.org/wiki/JSON>
3. Wikipedia. *PIP*. [Consulta: Julio 2017]
[https://es.wikipedia.org/wiki/Pip_\(administrador_de_paquetes\)](https://es.wikipedia.org/wiki/Pip_(administrador_de_paquetes))
4. MongoDB. *PyMongo*. [Consulta: Abril-Mayo 2017]
<https://api.mongodb.com/python/current/>
5. Django Software Foundation. *Django*. [Consulta: Mayo-Junio 2017]
<https://docs.djangoproject.com/en/1.11/topics/auth/default/>
6. Saif M. Mohammad. *Diccionario de emociones*. [Consulta: Mayo 2017]
<http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>
7. Konstantin Kudryashov. *Behat*. [Consulta: Julio 2017]
<http://behat.org/en/latest/>
8. Konstantin Kudryashov. *Mink*. [Consulta: Julio 2017]
<http://mink.behat.org/en/latest/>
9. Apache License. *Selenium*. [Consulta: Julio 2017]
<http://www.seleniumhq.org/>
10. Facebook. *Facebook Insights*. [Consulta: Enero-Febrero 2017]
<https://www.facebook.com/help/search/?q=insights>
11. Emeric Ernout. *Agorapulse*. [Consulta: Enero-Febrero 2017]
<https://www.agorapulse.com/es/>

12. Dennis Yu. *Blitzmetrics*. [Consulta: Enero-Febrero 2017]
<https://blitzmetrics.com/>
13. Rasmus Møller Nielsen. *Komfo*. [Consulta: Enero-Febrero 2017]
<https://komfo.com>
14. *PostAcumen*. [Consulta: Febrero-Marzo 2017]
<http://postacumen.com>
15. Alexander Peiniger. *Quintly*. [Consulta: Marzo-Abril 2017]
<https://www.quintly.com/>
16. Jan Rezab. *Socialbakers*. [Consulta: Marzo-Abril 2017]
<https://www.socialbakers.com>
17. Dane Atkinson. *Sumall*. [Consulta: Marzo-Abril 2017]
<https://sumall.com/>
18. Ricky Yean. *Crowdbooster*. [Consulta: Marzo-Abril 2017]
<http://crowdbooster.com/>
19. Stéphane Baudin. *Grytics*. [Consulta: Abril-Mayo 2017]
<https://grytics.com/es/>
20. GNU AGPL v3.0. *MongoDB*. [Consulta: Abril 2017]
<https://www.mongodb.com/es>
21. Tom Preston Werner, Chris Wanstrath, PJ Hyett. *GitHub*. [Consulta: Julio 2017]
<https://github.com/>
22. Python Software Foundation License. *Python*. [Consulta: Julio 2017]
<https://www.python.org/>

Anexo I PIP

PIP es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python.

Muchos paquetes pueden ser encontrados en el **Python Package Index (PyPI)**. **Python 2.7.9** y posteriores (en la serie **Python2**), **Python 3.4** y posteriores incluyen **PIP** (**pip3** para **Python3**) por defecto.

Una ventaja importante de **PIP** es la facilidad de su interfaz de línea de comandos, la cual permite instalar paquetes de software. Fácilmente desde solo una orden:

```
pip install nombre-paquete
```

Los usuarios también pueden fácilmente desinstalar algún paquete:

```
pip uninstall nombre-paquete
```

Anexo II Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la **Python Software Foundation**. Posee una licencia de código abierto, denominada **Python Software Foundation License**.

Instalacion de Python

```
$ sudo apt-get install -y python3-venv
```

Dependencias

```
$ sudo apt-get install python-pip
```

Anexo III Django

Como ya se ha explicado en el apartado 3.1., **Django** es un *framework* de **Python** e instalarlo es tan sencillo como ejecutar el siguiente comando.

```
pip install django
```

Para arrancar el servidor de **Django** ejecutamos el siguiente comando.

```
python manage.py runserver
```

Para conectar con **MongoDB** también necesitamos la librería correspondiente para poder navegar y realizar por este y realizar todas las operaciones sobre la base de datos. Se instalara con el siguiente comando:

```
python -m pip install pymongo
```

Anexo IV MongoDB

A niveles teóricos, en el apartado 3.2.2. se ha explicado qué es **MongoDB**. Ahora se procede a explicar la instalación de **MongoDB** sobre **Ubuntu**. Para ello se han que seguir los siguientes pasos:

Paso 1: Importar la clave pública de **MongoDB**.

```
Sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

Paso 2: Generar el fichero con el repositorio URL de MongoDB.

```
Echo `deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist  
10gen`
```

Paso 3: Actualizar el repositorio local.

```
Sudo apt-get update
```

Paso 4: Instalar la última versión estable de MongoDB.

```
Sudo apt-get install mongodb-org  
#OR  
sudo apt-get install -y mongodb
```

Paso 5: Para arrancar el servicio de **MongoDB**:

```
sudo service mongod start
```