



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València

Obtención, procesamiento, análisis y
visualización de datos de Facebook aplicado a
la docencia.

Proyecto Final de Carrera

Ingeniería Técnica en Informática de Sistemas

Autor: Óscar Ortiz Rodríguez

Directores: José Luis Poza Luján

Miguel Rebollo Pedruelo

21/10/2015

Tabla de contenidos

| | | |
|----------|--|----|
| 1 | Introducción | 4 |
| 1.1 | Contexto | 4 |
| 1.2 | Objetivos | 4 |
| 1.3 | Descripción del documento | 6 |
| 2 | Estudio Estratégico | 7 |
| 2.1 | Introducción | 7 |
| 2.2 | Sistemas Similares | 9 |
| | (S1) NodeXL | 9 |
| | (S2) Wolfram Alpha | 9 |
| | (S3) Netvizz | 9 |
| 2.3 | Análisis de sistemas | 9 |
| | Aspectos Objetivos | 10 |
| | Aspectos subjetivos | 11 |
| 2.4 | Características | 11 |
| | NodeXL | 11 |
| | Wolfram Alpha | 13 |
| | Netvizz | 14 |
| 2.5 | Conclusiones | 16 |
| 3 | Especificaciones de Requisitos | 19 |
| 3.1 | Introducción | 19 |
| 3.1.1 | Propósito | 19 |
| 3.1.2 | Ámbito del Sistema | 19 |
| 3.1.3 | Definiciones, Acrónimos y Abreviaturas | 20 |
| 3.1.4 | Referencias | 21 |
| 3.1.5 | Visión General del Documento | 21 |
| 3.2 | Descripción General | 22 |
| 3.2.1 | Perspectiva del Producto | 22 |

| | | |
|----------|---|-----------|
| 3.2.2 | Funciones del Producto | 23 |
| 3.2.3 | Características de los Usuarios | 24 |
| 3.2.4 | Restricciones..... | 24 |
| 3.2.5 | Suposiciones y Dependencias | 26 |
| 3.2.6 | Requisitos Futuros..... | 26 |
| 3.3 | Requisitos Específicos | 27 |
| 3.3.1 | Interfaces | 27 |
| 3.3.2 | Funciones..... | 30 |
| 3.3.3 | Requisitos de Rendimiento | 43 |
| 3.3.4 | Restricciones de Diseño | 43 |
| 3.3.5 | Atributos del Sistema | 43 |
| 3.3.6 | Otros Requisitos..... | 43 |
| 3.4 | Apéndices..... | 43 |
| 4 | Diseño..... | 44 |
| 4.1 | Introducción | 44 |
| 4.2 | Especificación conceptual | 44 |
| 4.3 | Conclusiones | 53 |
| 5 | Implementación e Implantación..... | 54 |
| 5.1 | Introducción | 54 |
| 5.2 | Sistema Desarrollado | 54 |
| 5.3 | Instalación | 63 |
| 5.4 | Pruebas..... | 70 |
| 5.5 | Conclusiones | 81 |
| 6 | Conclusiones | 83 |
| 6.1 | Trabajo Desarrollado | 83 |
| 6.2 | Aportaciones..... | 85 |
| 6.3 | Ampliaciones Futuras | 86 |
| 7 | Referencias | 90 |



1 Introducción

1.1 Contexto

La vida evoluciona. Avanza a tantos niveles, y tan sumamente rápido, que resulta casi una irresponsabilidad evolutiva no pararse a estudiar esos cambios e intentar anexarse a ellos, nutrirse de sus implicaciones, de su imbricación en todos y cada uno de los aspectos cotidianos y, en la medida de lo posible, aprovecharse de ellos con el fin de mejorar nuestras delimitadas parcelas cognitivas, educacionales y laborales, como en el caso que atañe de forma directa al presente proyecto: la docencia universitaria.

Las relaciones interpersonales siempre han sido un rasgo evolutivo de las especies animales, y más concretamente de la humana. Pero ha sido en estas últimas décadas con el asentamiento de las nuevas tecnologías, con el desarrollo de conceptos como “Internet” y su globalización, cuando se ha posibilitado la ruptura de barreras naturales como la distancia física, o la falta de tiempo. Las redes sociales se han convertido en un medio de comunicación directo entre conocidos y desconocidos, una manera fácil y rápida de crear rutas para la difusión de mensajes y conocimientos, lo cual hace que tenga mucho sentido el plantearse la utilización de todo ese potencial en la elaboración de nuevas estrategias de enseñanza.

Una educación de calidad requiere del uso adecuada de todas las herramientas posibles para mejorar, agilizar y llegar de una forma más directa al alumno, de incluirlo y hacerlo partícipe del propio proceso educativo, de hacer que se sienta parte del mismo no sólo como un mero espectador a tiempo completo, sino como una entidad activa del mismo. Las redes sociales son un inmejorable medio para tal fin.

Hoy en día es muy difícil imaginar a un alumno en edad universitaria que nunca haya interactuado de alguna forma con una red social. Y no sólo eso, es casi un meta lógica el conseguir que las nuevas generaciones se sientan cómodas con su uso, y que las vean no sólo como una herramienta personal o lúdica, sino que sepan apreciar el futuro educacional y laboral de las mismas.

1.2 Objetivos

Focalizando el tema sobre el objetivo concreto que persigue este proyecto se puede resumir en el siguiente: analizar una red social concreta, en este caso Facebook, desarrollando un plan de integración con la docencia de una asignatura (simulada o real) que permita la gestión de la misma a través de entidades propias de la red social.

Para alcanzar este objetivo se afrontarán una serie de tareas concretas, como definir una serie de modelos necesarios para representar la información que se va a obtener:

- **Grupo:** Se utilizarán los grupos de Facebook como grupo docente de la asignatura, invitando a los alumnos a incluirse en el mismo y fomentando el uso cotidiano del mismo, a través de entradas, vídeos, documentos, comentarios y cualquier otro elemento que tenga una utilidad docente.
- **Eventos:** Se utilizarán eventos para definir unidades lectivas específicas, para confeccionar grupos de trabajo (públicos o privados) con un subgrupo de alumnos concreto (grupos de trabajo), o para difundir eventos reales como conferencias, ponencias, prácticas, exámenes etc.

Por otro lado, se necesitará no sólo desarrollar un sistema de seguimiento que permita la correcta gestión del mismo, sino la lógica condición de potenciar a nivel docente el uso del mismo. Es decir, de poco servirá crear un sofisticado método para la obtención y el procesamiento de datos, si no hay datos que procesar. No obstante este no deja de ser un objetivo que escapa al presente proyecto.

Así pues una primera tarea, en cuanto a la red social se refiere, será el estudio de todo lo concerniente a la comunicación con la misma a través de sus librerías públicas (Graph API), la gestión de permisos de acceso y licencias de la aplicación, siendo necesario darse de alta como desarrollador de Facebook (paso que se explicará en profundidad en el capítulo 5. implementación e implantación)

Una segunda tarea será el volcado de los datos a un sistema de gestión de base de datos, eligiendo para este caso un sistema NoSQL que permitirá un mejor análisis y procesamiento de los datos obtenidos, así como su visualización posterior. De esta forma será necesario el estudio de un sistema de gestión de base de datos específico, su puesta en funcionamiento y el dominio de su propio lenguaje de consultas.

Por último será necesario el análisis de los datos obtenidos haciendo uso de las teorías correspondientes y de las herramientas de visualización y modelado específicas para tal fin. Presentando los resultados adaptados a las necesidades docentes concretas, con el único fin de mejorar las actuales técnicas docentes y la calidad del aprendizaje. Para ello se definirán una serie de índices que permitan evaluar el uso de los grupos, teniendo en cuenta tanto la actividad del profesor como la de los alumnos que lo integren.

1.3 Descripción del documento

El presente documento tiene como propósito la definición conceptual y funcional de un sistema de obtención, procesamiento, análisis y visualización de datos de un grupo (o grupos) de Facebook creado y gestionado como complemento docente a la impartición de una asignatura.

La motivación principal es la de facilitar herramientas para el análisis de dichos grupos como elemento activo de la docencia, automatizando los procesos de análisis y la elaboración de resultados que ayuden a la posterior confección de conclusiones prácticas y medibles.

Para ello se realizará un estudio estratégico, comparando el impacto actual de aplicaciones que, de una manera más o menos genérica, interactúe con la misma red social (Facebook), analizando sus debilidades, fortalezas e intentando establecer un paralelismo con las decisiones tomadas a la hora de implementar los hitos de este proyecto.

Una vez analizados sistemas similares se procederá a la especificación funcional de requisitos, detallando el alcance de la aplicación haciendo uso de los estándares documentales adecuados para tal fin.

Posteriormente se entrarán en aspectos más técnicos tales como el diseño a nivel conceptual y formal, así como la implementación e implantación del sistema final para su uso.

Finalmente se presentará una visión a posteriori de todo el proceso a modo de conclusión, dónde se propondrán posibles mejoras futuras.

2 Estudio Estratégico

2.1 Introducción

El Social Media está de moda pero, al igual que muchos otros avances en materia tecnológica, quizás no su nacimiento pero sí su crecimiento y desarrollo están íntimamente ligados a intereses de índole mercantilistas, paralelizando en no pocas ocasiones su ciclo de vida a un plan estratégico de explotación económica: al fin y al cabo vivimos en un mundo de liberalismo económico en el que la promesa de rentabilidad es lo que seduce a la inversión de tiempo y capital. Y no sugiero que sea un planteamiento incorrecto, simplemente pongo de manifiesto uno de los motivos por los cuales es complicado encontrar herramientas de utilización de redes sociales que no estén enfocadas a la obtención de datos con fines empresariales.

Sólo hay que realizar una búsqueda superficial sobre herramientas de Networking (o análisis de redes sociales), para entender que la gran mayoría son aplicaciones de terceros, de empresas orientadas a la gestión de grandes cuentas y posicionamiento de marcas en internet. Y no sólo eso, estas diferencias resultan aún más marcadas dependiendo de la red social de la cual hagan su objetivo.

De esta forma, si hablamos por ejemplo de Twitter, una red social muy dinámica donde la inmediatez de los contenidos es su leitmotiv, su razón de ser, la cantidad de aplicaciones que analizan su uso se dispara exponencialmente (las empresas buscan un impacto directo e inmediato). Mientras que una red social como Facebook, cuyo contenido es más estático y perdurable en el tiempo, requiere de una estrategia de análisis diferente, enfocando la casi totalidad del proceso al análisis de utilización de perfiles personales o páginas de Facebook, que no son más que el desarrollo de un escaparate público, una forma de mostrar y vender un producto, bien sea cosas tan heterogéneas como un grupo de música, un libro o un sello político.

Así pues nos encontramos ante un escenario propicio para cierto grado de innovación. Por un lado la explotación de Facebook como red social a nivel docente es cuanto menos escasa; y por otro lado la utilización de Grupos y Eventos como herramientas lectivas, y no comerciales, es un terreno por descubrir y propicio para abonar. Esto supone un primer escollo en lo referente al diagnóstico de la situación actual, en cuanto a la aproximación lógica y deseable de poder “repetir” las virtudes ya demostradas de ciertas tendencias de uso; aunque también abre un gran abanico de posibilidades en cuanto a la demostración de las fortalezas particulares del producto

que vamos a desarrollar. Si habláramos a nivel mercantilista podríamos afirmar que no serán muchos los competidores que vamos a encontrar para nuestro producto final.

Sin embargo no es esta la finalidad del presente proyecto. Pero ¿cuál es el objetivo? ¿Dónde queremos llegar?

Como hemos comentado en la introducción del presente documento el objetivo es puramente educacional. La idea no es otra que el utilizar una red social de amplia difusión como herramienta adicional para difundir contenido lectivo. A grandes rasgos, y salvando las distancias, se pretende utilizar los grupos de Facebook como una especie de aula virtual desde donde dar contenido específico del temario, donde incluir elementos audiovisuales (vídeos, documentos) que ayuden y enriquezcan de forma práctica las sesiones teóricas y, sobre todo, donde motivar una participación activa del alumno más allá del estar sentado escuchando y tomando apuntes.

El conseguir un grupo de trabajo impersonal en el que el alumno se vea desligado de restricciones sociales y personales que le imposibiliten su participación (siempre es más fácil escribir que hablar), un entorno que propicie la implicación de los mismos a través de comentarios y eventos que fomenten su curiosidad, que los mantengan informados en tiempo casi inmediato y les permita interactuar más allá de los horarios establecidos para tal fin (clases, tutorías, etc.), hacen del proyecto una posibilidad manifiesta de mejora de los métodos de enseñanza.

Es más, desde un punto de vista institucional, al profesor le da un canal de retroalimentación con los alumnos que le permitirá una mejor evaluación de los mismos, un mayor y más cercano seguimiento, así como la obtención de índices de análisis de flaquezas y fortalezas de su propio trabajo, permitiéndole tomar decisiones para mejorar la docencia.

En definitiva, el objetivo no es otro que el de implementar una herramienta que automatice procesos que permitan al alumno aprender más y mejor, y al profesor enseñar más y mejor. Todos ganan.

En los siguientes puntos analizaremos una serie de herramientas de análisis de redes que conectan y extraen información de Facebook. Si bien estas no están orientadas de forma específica a la gestión de grupos, ni tienen una relación directa con el ámbito educacional per se, si ayudaran a entender como conectan con la red social y como analizan la información obtenida.

2.2 Sistemas Similares

Procederemos a analizar tres herramientas que, de una u otra forma, interactúan o basan su operativa en la conexión con Facebook. Introduciremos brevemente en qué consisten estos sistemas para, posteriormente, analizarlos y compararlos entre sí buscando sus puntos fuertes y sus carencias.

(S1) NodeXL

NodeXL [NETWORK Graphs. The Social Media Research Foundation.] es una herramienta orientada a la exploración de redes utilizando la teoría de grafos. Consiste en una plantilla libre y open-source para Microsoft Excel 2007, 2010 y 2013. Utiliza SNAP (Stanford Network Analysis Platform <http://snap.stanford.edu/>), una librería de análisis de redes y minería gráfica escrita en C++ y fácilmente escalable. Obtiene y almacena en una hoja de Excel una lista de Nodos y Aristas.

(S2) Wolfram | Alpha

Wolfram|Alpha [WOLFRAM Research Company. *Wolfram*] es un buscador de conocimientos y contenidos que, además, tiene una herramienta de análisis de perfiles personales de Facebook que genera un informe cerrado sobre el propio perfil del usuario. Es decir, no es una herramienta para nada configurable, y al ser un servicio prestado por una empresa privada la mayoría de apartados de los informes tienen una serie de opciones de pago que son las que realmente dotarían de cierto interés al análisis obtenido.

(S3) Netvizz

Netvizz [ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.] es una herramienta desarrollada para uso académico y personal, no comercial, que permite la extracción de datos de grupos y páginas de Facebook. Dispone de tres módulos integrados como aplicación de Facebook:

- **Group data:** centrado en el análisis de grupos de Facebook.
- **Page data:** permite obtener información sobre páginas de Facebook.
- **Page link network:** crea una red de páginas conectadas a través de los “me gusta” entre ellas.

2.3 Análisis de sistemas

Analizaremos los sistemas mencionados desde una perspectiva tanto objetiva (más adelante, más adelante), como subjetiva (más adelante), proponiendo como elementos de comparación los siguientes:

- Características Objetivas:

Obtención, procesamiento y visualización de datos de Facebook.

- C1 – Precio: Valor del producto y de las herramientas necesarias para su utilización.
 - C2 - Sistema Operativo: Sistema Operativo que requiere para su utilización.
 - C3 – Instalación: pasos requeridos para su instalación.
 - C4 – Dependencia: si requiere de otras herramientas o aplicaciones para su correcto funcionamiento.
 - C5 - Sistema Almacenamiento: en qué soporte se almacenan los datos.
 - C6 – Rapidez: velocidad de ejecución.
 - C7 - Posibilidades de configuración: si es un sistema cerrado o ampliamente configurable.
- Características Subjetivas:
- C8 - Facilidad Uso: determinará si la interfaz es amigable e intuitiva.
 - C9 - Facilidad Aprendizaje: curva de aprendizaje para poder utilizar la aplicación.
 - C10 - Relevancia de la información: la información obtenida es útil o no.
 - C11 – Objetivo: intención final de la aplicación.

Aspectos Objetivos

| | Precio | S.O | Instalación | Dependencia |
|----------------------|---|--|--|--|
| NodeXL | Licencia MS Excel. | Windows | Instalador ejecutable que personaliza office. | MS. Excel. Complemento para Facebook. |
| Wolfram Alpha | Suscripción PRO mensual (3'75 a 5'49 dólares) | No requiere, es una herramienta WEB | No requiere | Navegador WEB con acceso a internet. |
| Netvizz | Gratis | No requiere, es una aplicación de Facebook | Instalar en Facebook y aceptar permisos de acceso. | Depende de Facebook, necesita perfil de usuario. |

Tabla 1. Análisis de Aspectos Objetivos C1, C2, C3 y C4

| | Almacenamiento | Rapidez | Configuración |
|----------------------|--|--|--|
| NodeXL | Libro de Excel. | La obtención de datos es lenta | Muchas opciones de análisis de grafos y redes. |
| Wolfram Alpha | No requiere. | Rápida. Sólo extrae información de perfil personal de Facebook | No permite. |
| Netvizz | Ficheros gdf (bigrafo) y tsv (tabulados) | Depende del tamaño del grupo | Permite filtrado muy básico por fechas o número de entradas. |

Tabla 2. Análisis de Aspectos Objetivos C5, C6 y C7

Aspectos subjetivos

| | Uso | Aprendizaje | Relevancia | Objetivo |
|----------------------|--|---|--|---|
| NodeXL | Complejo, requiere conocimientos teoría de redes y análisis de grafos. | Curva de aprendizaje media, alta. | Datos útiles para análisis de redes. | Análisis de redes |
| Wolfram Alpha | Inexistente. Darse de alta y darle a un link. | No requiere. | Datos informativos poco útiles. | Saciar curiosidad del usuario y obtener dinero con versión de pago. |
| Netvizz | Fácil. Interfaz sencilla de Facebook sin apenas opciones. | No requiere, la documentación es muy escueta debido a su falta de complejidad | Datos útiles, sin procesar. Buena opción como importador de datos, no como analizador. | Educacional y de uso personal para estudio de grupos y páginas. |

Tabla 3. Análisis de Aspectos Subjetivos

2.4 Características

Analizando de forma más detallada las aplicaciones mencionadas podemos destacar las siguientes características para cada una de las mismas.

NodeXL

- Permite la manipulación de gráficos, cálculo de propiedades estructurales, genera grafos regulares y aleatorios, soporta atributos en nodos y aristas. Tiene una versión SNAP for Python que funciona sobre win64.
- Permite el cálculo de centralidad de grado, cercanía, intermediación (betweenness), centralidad de vector propio (eigenvector centrality), pagerank, coeficiente de clustering, densidad de grafo y más.
- Importa y exporta grafos a GraphML, Pajek, UCINet y matrix formats.
- Conexión directa con redes sociales, importando directamente de Twitter, Youtube, Flickr e email, o haciendo uso de uno de los pluggins disponibles para obtener información de Facebook, Exchange, Wikis, Surveys, WWW hyperlinks y nubes.
- Zoom y escalado de vértices para reducir el clutter.
- Layout flexible, usando algoritmos específicos para dibujar el grafo, o arrastrar los vértices con el ratón.
- Permite ajustar la apariencia: establecer el color, forma, tamaño, etiquetas y opacidad de cada vértice individual o permitir que NodeXL lo haga en base a atributos de vértice como el grado, la intermediación, la centralidad o el PageRank.
- Permite filtrado dinámico; ocultación instantánea de vértices y aristas usando una serie de filtros, ocultando todos los vértices menores a grado cinco, por ejemplo.

- Permite agrupación de Vértices: Agrupar los vértices del grafo por atributos comunes, o dejar que NodeXML analice su conectividad y automáticamente los agrupe en clusters. Hacer los grupos distinguibles usando formas, colores, colapsándolas con un click, o poniendo cada grupo en su propia caja dentro del grafo.
- Permite la automatización de tareas con un simple click.
- Utiliza un control Windows Presentation Foundation (WPF) que puede ser reutilizado en aplicaciones. De hecho la plantilla Excel no es más que un wrapper que trabaja sobre una serie de librerías precompiladas descargables. Se puede crear un ensamblado .NET que importe datos a la plantilla NodeXML. Generar una .DLL que se puede integrar simplemente copiándola en una carpeta específica, apareciendo posteriormente en el menú importación de la aplicación.

Para el caso concreto que nos atañe, la importación de Facebook, existe un proyecto denominado [Social Network Importer for NodeXML](#).

Este es un complemento que permite la descarga directa de Grupos y Páginas de Facebook:

- Tiene su propia interfaz gráfica de importación (más adelante).
- Puede extraer conexiones entre usuarios y entradas relacionadas por acciones como “me gusta” o comentarios. Relaciones usuario-usuario, entradas-entradas y usuario-entrada están disponibles.
- Puede importar datos sobre el perfil personal de un usuario de Facebook específico (ego network data importer), visualizando la cantidad de conexiones entre él y sus amigos, y sus amigos con otros usuarios.
- Dispone de una serie de algoritmos para dibujar el grafo: Fruchterman-Reingold (Predeterminado), Harel-Koren Fast Multiscale, Circle, Spiral, Horizontal Sine Wave, Vertical Sine Wave, Grid, Polar, Polar Absolute, Sugiyama, Random.

Podemos decir, por tanto, que NodeXL es una muy buena herramienta de análisis de redes, muy completa y no demasiado compleja en su utilización, aunque adolece de una dependencia absoluta de una plataforma que, si bien está muy extendida, no deja de ser una aplicación propietaria y de pago.

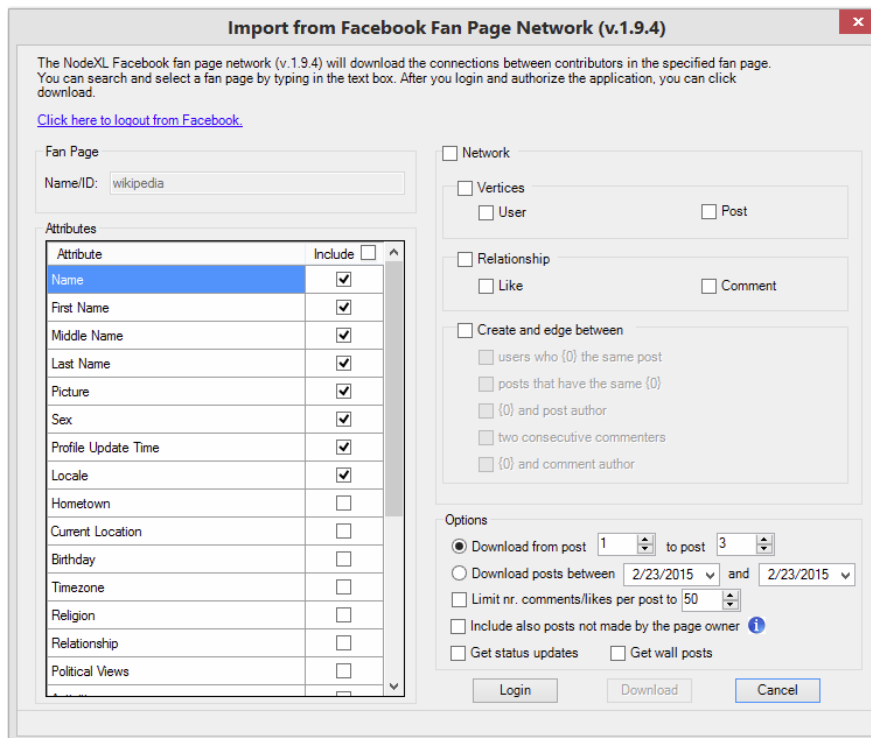


Fig. 1 - Interfaz Complemento Facebook NodeXL

Wolfram | Alpha

- El uso avanzado de las vistas interactivas requiere de una versión PRO de pago. Realizar zoom, descargar información, personalizar y descargar las gráficas, exportar en texto plano, activar la interactividad.
- Muestra datos personales del usuario.
- Muestra datos de actividad reciente con filtros por estado, links, videos, fotos y check-ins. Y filtro temporal por mes, semana o día.
- Muestra estadísticas de Entradas. Total de “me gusta”, total comentarios, medias, análisis de longitud contando caracteres, frecuencia de palabras, nube de palabras, entradas con más “me gusta”, entrada más comentada, usuarios más comentarores, usuarios que más comparten.
- Muestra estadísticas de check-in
- Muestra estadísticas de fotos. Álbumes totales, fotos propias, media de gente etiquetada en las fotos. Foto con más “me gusta”, foto con más comentarios.
- Muestra un análisis de amigos. Por género, por estado sentimental, por edades, ranking de personas más viejas y más jóvenes, Localización geográfica por mapa y por tabla país-número amigos, extremos geográficos, horarios locales de los amigos. Cumpleaños de los amigos (hoy, mañana, próximos 7 días). Nombres más comunes entre los amigos, apellidos más comunes. Amigos que comparten más amigos.

Amigos con más amigos. Friend Network (mapa de grafos con usuarios por nodos pero no utilizable si no tienes modo Pro). Red 3D interactiva de amigos (requiere interactividad, lo cual requiere modo Pro)

- Muestra tablas de amigos influyentes:
 - Top “social insiders”: amigo que comparte muchos amigos contigo. Un gran grado de Centralidad.
 - Top “Social Outsiders”: amigo que al menos comparte un amigo contigo. Centralidad de grado 0 o 1.
 - Top “social connectors”: usuario que conecta grupos de amigos que de otra manera estarían desconectados entre sí. Tienen un alto valor de intermediación (betweenness)
 - Top “social neighbors”: Usuario con un número muy bajo de amigos que tú no tienes. Tienen un bajo grado de non-ego vertex
 - Top “Social gateways”: Amigos que tienen muchos amigos que tú no compartes. Alto grado de non-ego vertex.

En general es una herramienta bastante limitada en cuanto a la falta de interactividad, la imposibilidad de personalización o de elección de datos a analizar, la poca utilidad de las estadísticas devueltas, y la obligatoriedad de dos procesos de registro: uno previo para otorgar permisos en Facebook, y uno posterior para darse de alta en su propio sistema vía web, con el añadido de que la página de registro tiene problemas de visualización dependiendo del navegador empleado.

Netvizz

- **Group data:** crea las redes y ficheros tabulados para amistades e interacciones dentro de grupos.
 - Extrae dos tipos de redes (más adelante) para grupos a los que perteneces, ofuscando los nombres de usuarios reales.
 - Conexiones de amistad. Puede tardar horas en calcular la red en función del número de integrantes del grupo.
 - Interacciones. Si un usuario comenta o le da a “me gusta” en la entrada de otro usuario se crea un link directo. Se trabaja con los 200 últimas entradas.
 - Se requiere el id del grupo. Siendo necesario el uso de herramientas externas para su obtención. (<http://lookup-id.com/>)
 - La interfaz de utilización está integrada en facebook y es muy básica (más adelante)

netvizz v1.05 (pages & groups only)

groups:

Netvizz can extract two types of social networks from **groups** (*groups are not pages!*) you are a member of (both create gdf files, users are **anonymized**):

- friendship connections (API limits for group data are changing regularly, current version should be able to get up to 5000 group members. This may take a very long time, i.e. hours).
- interactions (if a user comments or likes another user's post, a directed link is created - currently the last 200 posts are take into account)

group id (find group ids [here](#))

get **friendship connections** or **interactions**

file fields (friendship connections network - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

groupid: unique identifier of the group

file fields (interactions network - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

posts: number of posts a user posted to the group

file fields (interactions posts - tsv format - rows are posts):

message: the message of the post

created_time: when the post was made

comments: the number of comments the post received

likes: the number of likes the post received

commentsandlikes: the upper two summed

Fig. 2 - Interfaz Netvizz

file fields (friendship connections network - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

groupid: unique identifier of the group

file fields (interactions network - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

posts: number of posts a user posted to the group

file fields (interactions posts - tsv format - rows are posts):

message: the message of the post

created_time: when the post was made

comments: the number of comments the post received

likes: the number of likes the post received

commentsandlikes: the upper two summed

Fig. 3 – Estructura de ficheros devueltos por Group Data

- **Page data**: crea la red y los ficheros tabulados para la actividad de usuarios en entradas dentro de una página.

- Obtiene las últimas entradas de la página y crea (más adelante):
 - Un fichero con la información de un bigrafo en formato gdf, con las entradas, los usuarios y las conexiones entre ambos. Un usuario está conectado con una entrada si lo ha comentado o le ha dado a “me gusta”.
 - Un fichero tabulado (tsv) que lista diferentes métricas para cada entrada.
 - Un fichero tabulado (tsv) que contiene el texto de los comentarios de cada usuario (anonimizado)
 - El tiempo de proceso depende del tamaño de la página, siendo posible que se quede sin memoria para página grandes (páginas de más de un millón de comentarios y “me gusta”)
 - Los parámetros de entrada del script son:
 - Número de entradas (máximo 999)
 - O entradas entre dos fechas.
 - Permite establecer una búsqueda más rápida obteniendo sólo entradas estáticas.
 - Requiere el ID de la página.
- **Page link network:** crea una red de páginas conectadas a través de los “me gusta” entre ellas.

Debido a su marcado carácter educacional tiene la peculiaridad de enmascarar los nombres de los usuarios, eliminándolos de los datos obtenidos para evitar usos comerciales. De la misma forma, y siendo coherente con dicha política de uso, el código fuente no está liberado para evitar ingeniería inversa sobre el algoritmo de ofuscación de los nombres de usuario.

2.5 Conclusiones

Hemos analizado, por tanto, tres aplicaciones diferentes cada una de las cuales atiende a una ideología concreta. Desde un punto de vista analítico la intención ha sido evaluar una aplicación de código libre, con las ventajas inherentes a ello, una aplicación con orientación claramente comercial y una aplicación con motivación educacional.

Si nos paramos a repasar las características de cada una de ellas podemos concluir que es difícil encontrar una aplicación con orientación puramente educacional, y más concretamente que se centre en la obtención de datos de grupos de Facebook como herramienta de apoyo a la docencia.

file fields (network file - gdf format - nodes are either posts or users):

type: either "user" (if node is user) or "post_page_pageid" (post by page) or "post_user_pageid" (post by user)
type_post: Facebook's post classification (e.g. photo, status, etc.)
post_published: publishing date
post_published_unix: publishing date as Unix timestamp (for easy conversion and ranking)
user_locale: user selected interface language (empty if node is post)
user_sex: user specified sex (empty if node is post)
likes: number of actually retrieved likes a post received or a user made
likes_count_fb: Facebook provided like count for posts (can be higher than actually retrieved likes)
comments_all: number of comments made on a post or by a user
comments_base: number of base level comments (in threaded conversations)
comments_replies: number of reply level comments (in threaded conversations)
comment_likes: number of likes on comments (posts only)
shares: number of shares (posts only)
engagement: likes, comments_all and shares summed
post_id: id of the post (empty for users)
post_link: link of the post (empty for users)
edge weight encodes the number of times a user commented or liked a post

file fields (stat file - tsv format - rows are posts):

type: Facebook's post classification (e.g. photo, status, etc.)
by: either "post_page_pageid" (post by page) or "post_user_pageid" (post by user)
post_message: text of the post
picture: picture URL (if a picture is attached to the post)
link: link URL (if the post points to external content)
link_domain: domain name of link
post_published: publishing date
post_published_unix: publishing date as Unix timestamp (for easy conversion and ranking)
likes: number of actually retrieved likes a post received or a user made
likes_count_fb: Facebook provided like count for posts (may be higher than actually retrieved likes due to privacy or deletion)
comments_count_fb: Facebook provided comment count for posts (may be higher than actually retrieved likes due to privacy or deletion)
comments_all: retrieved number of comments made on a post or by a user
comments_base: number of base level comments (in threaded conversations)
comments_replies: number of reply level comments (in threaded conversations)
shares: number of shares
comment_likes: number of likes on comments
engagement: likes, comments_all and shares summed
post_id: id of the post
post_link: link of the post

file fields (comments file - tsv format - rows are comments):

post_id: id of the post
post_by: author of the post
post_text: text of the post
post_published: publishing date of the post
comment_id: id of the comment
comment_by: author of the comment
is_reply: whether the comment is a reply to another comment (in threaded conversations)
comment_message: text of the comment
comment_published: publishing date of the comment
comment_like_count: number of likes on the comment

Fig. 4 – Estructura de ficheros devueltos por Page Data

La primera aplicación (NodeXL) es, sin duda, la más completa de las tres. Una amplia funcionalidad, numerosos complementos que facilitan su uso y su orientación a diversos escenarios y de código libre, pero dependiente de un sistema propietario (y de pago) como es Microsoft Excel.

La segunda aplicación es una muestra de lo que el mercado ofrece: una forma de ganar dinero inundando con información muchas veces irrelevante. Es cierto que habrá infinidad de herramientas mejor concebidas y preparadas, pero lo que es seguro es que todas ellas se orientan a la venta de productos o al posicionamiento de marcas, no al estudio docente.

La tercera aplicación es la que más se acerca a nuestras necesidades, el estudio de un grupo de Facebook, pero no deja de ser un acercamiento muy tímido, muy escaso para lo que realmente necesitamos, con pocas opciones de configuración y sin abordar el apartado de análisis de los datos.

Si tenemos en cuenta que el objetivo final de nuestro sistema no sólo es el de extraer y almacenar datos de Facebook, sino también el de darles una forma legible y útil para analizar el impacto de los mismos a nivel educacional como parte del seguimiento directo de una determinada asignatura, es necesario poder definir una serie de índices específicos que no se encuentran en aplicaciones de corte generalista como las estudiadas.

3 Especificaciones de Requisitos

3.1 Introducción

La presente sección es una especificación de requisitos de software (ERS) para una aplicación que permita la obtención, procesamiento, análisis y visualización de datos de Facebook aplicado a la docencia. Esta subsección de la memoria del proyecto se ha desarrollado siguiendo el estándar IEEE Prácticas Recomendadas para Especificaciones de Requisitos Software ANSI/IEEE 830-1998.

3.1.1 Propósito

El propósito del presente documento es detallar las especificaciones que definen el desarrollo de una aplicación web que permitirá la importación de datos de grupos de Facebook en un sistema de gestión de bases de datos, así como la posterior generación de estadísticas que permitan analizar el impacto de los mismos a nivel académico.

Este documento va dirigido a los integrantes del equipo de diseño y desarrollo de la aplicación, así como al administrador final de la misma representado en la figura del administrador del grupo de Facebook que, en líneas generales, coincidirá con el profesor o profesores que impartan la asignatura (o módulo docente) asociada al grupo de Facebook.

3.1.2 Ámbito del Sistema

El presente sistema es una aplicación web que posibilitará al impartidor de una determinada materia educativa (asignatura, módulo, master), extraer información de utilización de un grupo de Facebook.

Será por tanto necesaria la implicación activa tanto de los alumnos como integrantes del grupo, cómo del profesor en calidad de administrador del mismo. La aplicación no creará en ningún caso contenido en el grupo, siendo esto tarea de los miembros. La principal función de la aplicación será la de obtener todos los datos del grupo para importarlos a un sistema de gestión de base de datos que permita su posterior consulta y manipulación.

Las dos principales funciones de la aplicación serán la de importación, y la de generación de estadísticas para evaluar la participación activa del grupo creado para tal fin.

Esta aplicación está ideada como herramienta de apoyo al proceso educativo. Actualmente hay mucha información sobre cómo incluir las nuevas tecnologías (Referencias), y en concreto las redes sociales, en el plan lectivo de institutos y universidades, no obstante la cantidad de aplicaciones que permitan la gestión de la información de estas redes sociales desde un punto de

vista académico es muy escasa. Esto supone el incremento de tiempo y trabajo extra para los docentes, así como la necesidad de formación sobre determinados aspectos tecnológicos que serían deseables para el correcto aprovechamiento del proceso (uso de hojas de cálculo, aplicación de fórmulas matemáticas para obtener estadísticas útiles, etc...)

El objetivo final de la aplicación es la de enmascarar toda la complejidad pareja a la obtención y procesamiento de datos, los cuales pueden llegar a incluir periodos lectivos muy prolongados en el tiempo y qué, por tanto, pueden suponer grandes cantidades de información a tener en cuenta.

La aplicación devolverá gráficas con determinados índices que se consideran de utilidad para el estudio del impacto académico, así como la posibilidad de exportar dichos datos para su posterior gestión.

3.1.3 Definiciones, Acrónimos y Abreviaturas

Definiciones:

- **Recurso:** Entidad almacenada en el Sistema de Gestión de Base de Datos que corresponderá con un elemento concreto del sistema a estudiar.
- **Operación:** Acción que se lleva a cabo sobre las entidades. Pueden ser desencadenadas por la interacción con un usuario o por la ejecución de tareas programadas.
- **Usuario WEB:** usuario que utilizará sus credenciales como usuario Facebook para la correcta utilización de los recursos y operaciones de extracción de datos.
- **API de Facebook:** La API de acceso a Facebook contiene una determinada nomenclatura que conviene conocer:
 - **Acces Token (Token de Acceso):** Elemento necesario para la validación y obtención de permisos de acceso sobre Facebook. Es un elemento de la API de Facebook
 - **Node (Nodo):** Elementos de Facebook sobre los que se puede realizar consultas, como por ejemplo Usuario, Grupos, Fotos, Comentarios.
 - **Arista (Edge):** Conexiones entre los nodos, como por ejemplo las entradas de un Grupo.
 - **Group (Grupo).** Entidad que representa un Grupo de Facebook.
 - **Post (Entrada):** cada una de las entradas de un grupo o evento, creadas por el administrador o cualquier otro usuario.
 - **Comment (Comentario):** Respuesta de un usuario a un nodo concreto (entrada, evento)

- Event (Evento): Es un tipo de entrada especial que permite generar contenido propio y agrupado en torno a un tema específico que decide el usuario que lo creó.
- User (Usuario): Usuario de Facebook.

Acrónimos:

- SGBD: Sistema de Gestión de Base de Datos.
- ERS: Especificación de Requisitos de Software.
- API: Application Programming Interface, o Interfaz de programación de aplicaciones. Es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción
- BD: Base de datos.

3.1.4 Referencias

- Especificación de Requisitos según el estándar de IEEE 830 (IEEE Std. 830-1998)
- Referencia Graph API de facebook [[Link](#)]
- 99 Ways You Should Be Using Facebook in Your Classroom [[Link](#)]
- Facebook for Learning Communities: Groups vs. Pages [[Link](#)]
- Social network analysis [[Link](#)]
- Un estudio de la literatura de Facebook en el ámbito académico [[Link](#)]

3.1.5 Visión General del Documento

Esta subsección de la memoria del proyecto se encuentra dividida en cuatro partes. La primera es una somera introducción al mismo, estableciendo la motivación y el ámbito de aplicación en rasgos generales.

La segunda parte es la descripción general del proyecto, con especial énfasis en una definición funcional del mismo, así como en la enumeración de las características y restricciones que contemplará.

La tercera parte es una detallada descripción de los requisitos concretos que deberán tener en cuenta tanto los desarrolladores de la aplicación, como los usuarios de la misma. El grado de detalle deberá ser suficiente para la correcta comprensión de la problemática y funcionalidad requerida por la aplicación, plasmando fielmente los casos de uso y las necesidades observables tanto por los desarrolladores como por los usuarios finales de la misma.

La última sección será un compendio de toda esa información relevante para la ERS pero que, propiamente, no forme parte de la ERS.

3.2 Descripción General

A continuación se procederá a la descripción del contexto de la aplicación, poniendo de manifiesto todos aquellos factores que intervengan de forma directa o indirecta en el desarrollo del producto, así como en el posterior uso del mismo.

Para ello se abordará la perspectiva del producto, o relación del producto con otros productos ya existentes que requerirá para su correcto funcionamiento. Posteriormente se resumirán las funciones del producto, las características de los usuarios que interactuarán con la aplicación, así como las restricciones o limitaciones impuestas por y para el proyecto. Se nombrarán también los casos que podrían hacer variar los requisitos de la aplicación, y se dará una propuesta de futuras mejoras del sistema.

3.2.1 Perspectiva del Producto

La aplicación será desarrollada para su utilización desde entornos WEB, permitiendo al usuario interactuar desde cualquier plataforma con un navegador WEB estándar. No obstante, la aplicación tiene una dependencia con una serie de sistemas externos detallados a continuación.

Plataforma WEB.

La aplicación requiere de un Servidor WEB instalado con complementos específicos dependientes del lenguaje de programación elegido para su desarrollo. Este servidor debe de tener acceso a internet y poder ser objeto de consultas HTTP (conexiones entrantes) por los usuarios que utilizarán la aplicación. Así mismo debe de permitir conexiones directas con los servidores WEB de Facebook.

Se utilizará a este nivel, por tanto, el Protocolo HTTP y el protocolo HTTPS como interfaces de comunicación.

- Protocolo HTTP (Hypertext Transfer Protocol): protocolo que utilizan los elementos de software de arquitectura web para comunicarse.
- Protocolo HTTPS (Hypertext Transfer Protocol Secure): protocolo basado en el HTTP pero destinado a la transferencia segura de datos.

API de Facebook.

Dado que la aplicación debe trabajar directamente sobre un producto propietario (Facebook), se hace necesaria la utilización de su API (Graph API), tanto para la validación a través de Tokens de acceso, como para las posteriores consultas sobre sus nodos y aristas.

Graph API es una API basada en HTTP, con lo que las interfaces de comunicación serán las mismas anteriormente reseñadas para la plataforma WEB.

Sistema de Gestión de Base de Datos.

El SGBD deberá estar instalado, configurado y en ejecución para el correcto funcionamiento de la aplicación. En este caso se ha optado por dos sistemas diferentes:

- Un sistema NoSQL que permitirá almacenar información primando las relaciones entre los datos y facilitará la posterior aplicación de conceptos inherentes a la teoría de grafos.
- Un sistema SQL para los datos de carácter más estático que no requieren de un análisis posterior, como por ejemplo la información sobre los días lectivos de cada año académico.

La interfaz de acceso o consulta es dependiente del SGBD elegido. Para el primer SGBD NoSQL se empleará el lenguaje de consulta Cypher, desarrollado para trabajar sobre Neo4j. Para el segundo SGBD se utilizará SQL estándar.

3.2.2 Funciones del Producto

El sistema proporciona las funciones representadas a continuación con diagramas de casos de uso, organizados por tipo de usuario y por tipo de funcionalidad.

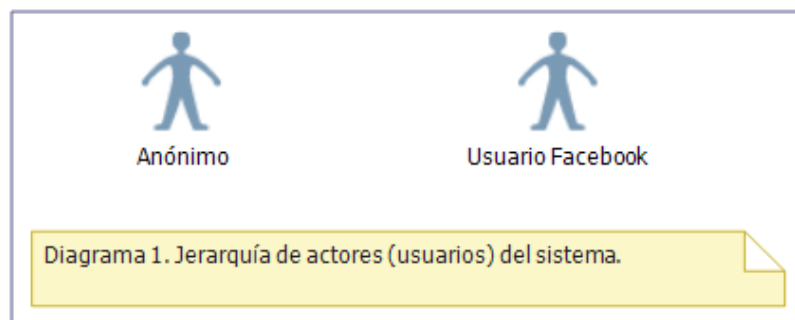


Diagrama 1. Jerarquía de actores (usuarios) del sistema.

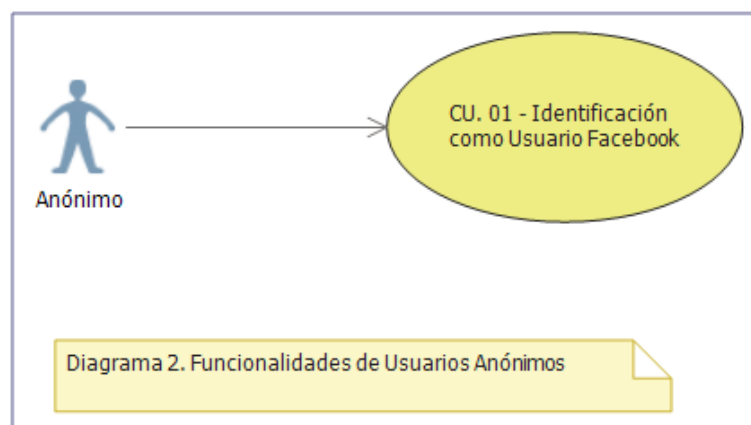


Diagrama 2. Funcionalidades de Usuarios Anónimos

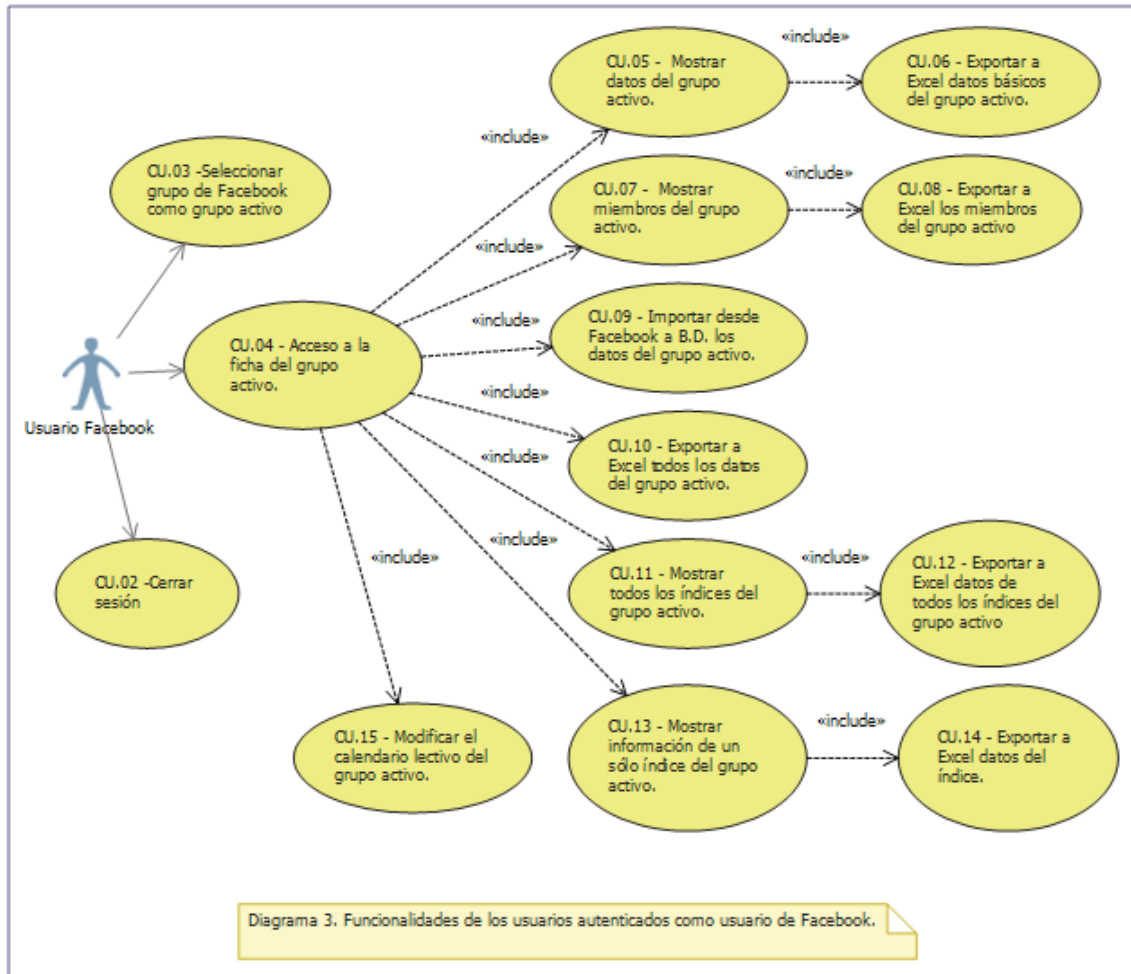


Diagrama 3. Funcionalidades de usuarios Facebook

3.2.3 Características de los Usuarios

Existen dos tipos de Usuarios o Roles diferentes (Diagrama 1):

- Anónimo (Diagrama 2): Usuarios no autenticados en el sistema, su funcionalidad está limitada a la propia autenticación
- Usuario Facebook (Diagrama 3): Usuarios que se validarán utilizando el sistema de validación propietario de Facebook, otorgando a la aplicación permisos especiales para interactuar con su propio perfil de Facebook. Esta es una condición necesaria para poder acceder a los datos de los grupos de los que es miembro.

3.2.4 Restricciones

Se deberán contemplar y cumplir una serie de condiciones de carácter obligatorio:

- Los desarrolladores de la aplicación deberán de tener una cuenta de Facebook para realizar pruebas sobre el mismo.

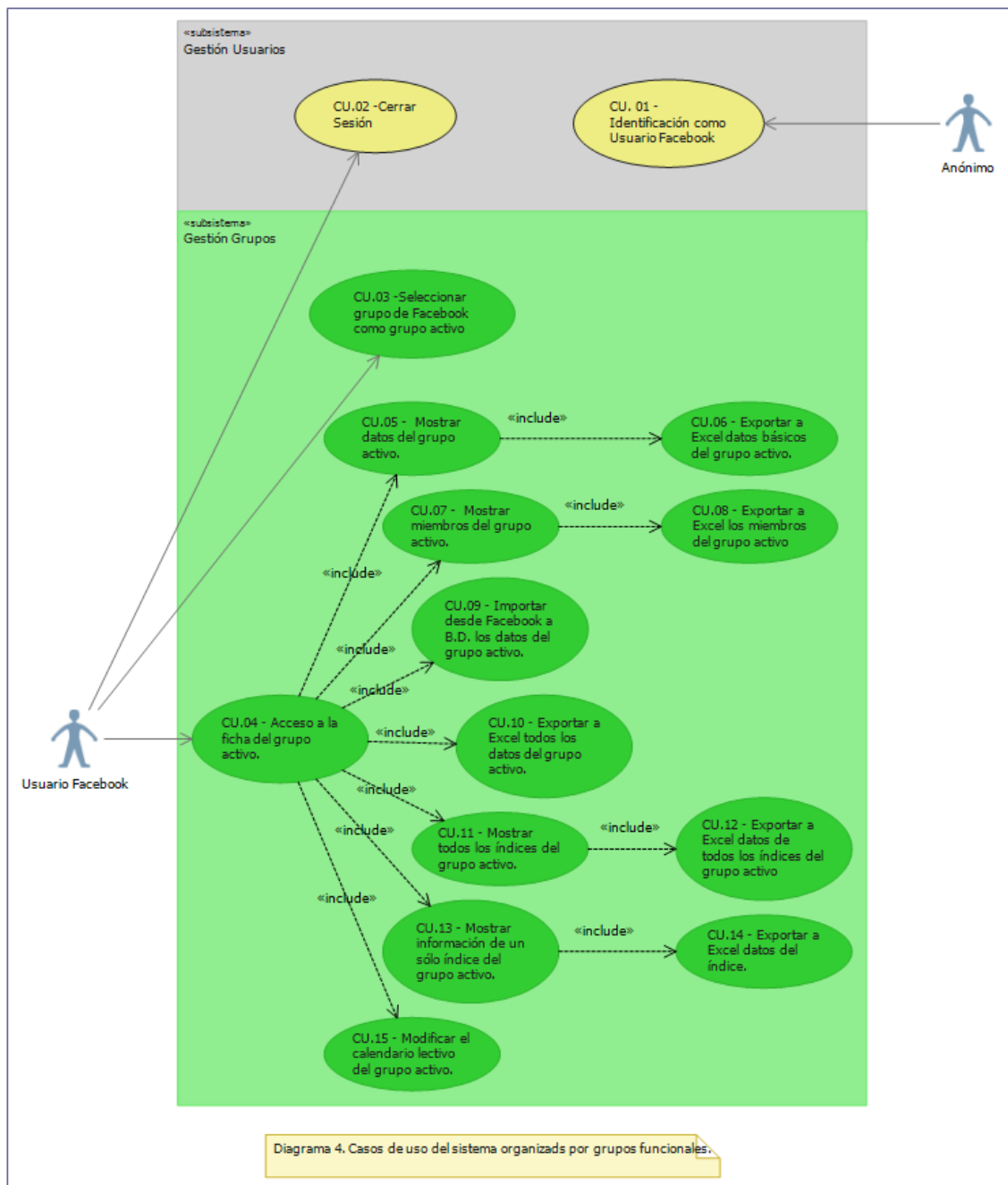


Diagrama 4. Casos de uso del sistema organizados por grupos funcionales

- Es necesario la creación de una Aplicación de Facebook, definiendo un dominio de uso para la misma, así como los parámetros necesarios para que la aplicación web pueda comunicarse con los servidores de Facebook. Para ello se debe de dar de alta como “Desarrollador de Facebook”, paso obligatorio impuesto y que supone la validación de un perfil personal como tal (validación vía SMS por teléfono móvil) y la utilización de la herramienta de creación y configuración de aplicaciones de Facebook.

- Al ser una aplicación WEB es imprescindible la existencia de un servidor WEB accesible por los usuarios y un dominio específico publicado de forma que sea “alcanzable” por los servidores de Facebook.
- El lenguaje de programación debe de ser Python, y el framework usado para su desarrollo Django. Por tanto, el servidor WEB debe de ser compatible con estas elecciones.
- El SGBD debe de ser Neo4j para la parte de importación de datos del grupo de Facebook, sistema NoSQL que permitirá un estudio posterior aplicando teoría de grafos y análisis de redes como posible mejora futura. Para ello será necesario conocer su lenguaje de consulta Cypher.
- Para los datos de carácter más estático (gestión de cursos académicos) se utilizará SQLite, integrada en la solución y que, por tanto, no requerirá de ninguna tarea de administración extra.
- El Usuario Facebook deberá gestionar, de forma paralela, un grupo de Facebook del cual deberá ser administrador. En ningún caso incidirá la presente documentación en la política de gestión de dicho grupo, siendo una elección totalmente personal e inherente al proceso y las técnicas educativas del profesor que la imparte.

3.2.5 Suposiciones y Dependencias

Los siguientes factores podrían derivar, en caso de que se incumplan o modifiquen, en una variación de los requisitos tratados hasta ahora:

- Pérdida o cambio del dominio WEB.
- Hardware necesario para la implantación del sistema.
- Servidor WEB y SGBD adecuados para el correcto desempeño de la aplicación.
- Cambio en las especificaciones de la API de Facebook o de su forma de autenticación y obtención de permisos sobre su plataforma.
- Conocimientos a nivel de usuario de navegación WEB de los usuarios de la aplicación.

3.2.6 Requisitos Futuros

Algunas posibles mejoras para el sistema que podrían analizarse e implementarse en un futuro:

- Utilización de las consultas agrupadas en las llamadas a la API de Facebook, con el consiguiente aumento de rendimiento y velocidad en el proceso de importación.
- Nuevos índices para el análisis de los datos obtenidos de los grupos de Facebook.

- Generación de grafos y estudio de los mismos aplicando teoría de grafos y análisis de redes.
- Implementar la lógica asociada a un nuevo Rol de usuario que permita gestionar la información de todos los grupos almacenados en el SGBD, independientemente de si es o no miembro del mismo.
- Diseño de una interfaz Responsive para poder acceder desde cualquier dispositivo tecnológico con un navegador Web (Smartphone, Table)
- Automatización de la extracción de datos de un grupo con tareas programadas sin necesidad de la acción directa de un usuario.

3.3 Requisitos Específicos

En esta sección se procede a detallar los requisitos que debe reunir y satisfacer la aplicación, extrapolados de los comportamientos observables desde todos los posibles puntos de vista, tanto de los usuarios implicados, como de sistemas externos con los que deba interactuar.

El análisis obtenido debe ser condición necesaria y suficiente para desarrollar de forma consistente y correcta la aplicación final, abarcando tanto la funcionalidad de la misma, como las restricciones inherentes y derivadas de la utilización de plataformas externas como Facebook.

3.3.1 Interfaces

Interfaces de Usuario

| | |
|--------------------|---|
| NOMBRE | FORMULARIOS WEB |
| CÓDIGO | RI.001 |
| ACTORES | Anónimo, Usuario Facebook |
| FUNCIÓN | Introducción de datos necesarios para la realización de una determinada función de la aplicación |
| DESCRIPCIÓN | Siempre que una determinada función u operación de la aplicación requiera de datos por parte del usuario, se utilizará un formulario WEB con tantos campos como datos se requieran. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|--|
| NOMBRE | ERRORES DE VALIDACIÓN |
| CÓDIGO | RI.002 |
| ACTORES | Anónimo, Usuario Facebook |
| FUNCIÓN | Comunicación al usuario de un error de validación de formulario WEB. |
| DESCRIPCIÓN | En el caso de producirse un error de validación en un formulario WEB se mostrará el error en la parte superior del formulario, de forma que capte la atención inmediata del usuario. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|---|
| NOMBRE | LISTADO DE RECURSOS |
| CÓDIGO | RI.003 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Listar el contenido de un recurso determinado |
| DESCRIPCIÓN | El usuario podrá listar los diferentes registros de un recurso concreto, mostrándose una estructura visual (lista o tabla), que contenga el identificador del mismo y, de forma opcional, una descripción visual y un acceso directo a las posibles operaciones que se puedan llevar a cabo sobre el mismo. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|---|
| NOMBRE | DETALLE DE RECURSOS |
| CÓDIGO | RI.004 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Mostrar la información detallada de un recurso |
| DESCRIPCIÓN | El usuario podrá consultar los detalles de un recurso en concreto siempre que disponga de permisos para ello. Se mostrarán con una página WEB y opcionalmente se podrán incluir elementos específicos para realizar acciones y operaciones sobre el recurso mostrado. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|--|
| NOMBRE | Exportación |
| CÓDIGO | RI.005 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar datos a Excel |
| DESCRIPCIÓN | El sistema permitirá la exportación a ficheros Excel de los datos almacenados en el SGBD NoSQL referente a los grupos de Facebook. |
| COMENTARIOS | |
| REFERENCIAS | |

Interfaces de Hardware.

| | |
|--------------------|---|
| NOMBRE | PUERTOS DE COMUNICACIÓN |
| CÓDIGO | RI.006 |
| ACTORES | |
| FUNCIÓN | Restricción de Hardware |
| DESCRIPCIÓN | El sistema debe implantarse en un servidor configurado con acceso a internet, tanto para conexiones entrantes como salientes. El puerto que debe encontrarse abierto para la comunicación e intercambio de datos vía WEB (protocolo HTTP) será el 80. |
| COMENTARIOS | |
| REFERENCIAS | |

Interfaces Software Externas.

| | |
|--------------------|--|
| NOMBRE | SERVIDOR WEB |
| CÓDIGO | RI.007 |
| ACTORES | |
| FUNCIÓN | Restricción de Software sobre el Servidor WEB |
| DESCRIPCIÓN | El servidor WEB debe ser compatible con el lenguaje de programación utilizado (Python) y con el framework de desarrollo empleado (Django). Debe ser capaz tanto de aceptar peticiones HTTP por el puerto 80, como de permitir peticiones HTTP salientes con los servidores de Facebook para la obtención de datos. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|--|
| NOMBRE | BASES DE DATOS |
| CÓDIGO | RI.008 |
| ACTORES | |
| FUNCIÓN | Restricción de Software sobre el SGBD |
| DESCRIPCIÓN | <p>El sistema empleará dos SGBD diferentes. Uno para el almacenamiento de la información sobre los grupos de Facebook, y otro para guardar datos de carácter más estático como la configuración de los cursos académicos o las sesiones y los tokens de acceso de los usuarios validados.</p> <p>El primer SGBD será Neo4j, y debe estar instalado y configurado de forma correcta para aceptar conexiones desde el Servidor WEB. Para ello debe cumplir las siguientes características:</p> <ul style="list-style-type: none"> ▪ Tener definido un usuario y una contraseña para poder conectarse de forma segura y utilizar sus recursos. ▪ Tener definido el puerto de comunicación específico. ▪ Estar siempre en ejecución. <p>El segundo SGBD será SQLite, no siendo necesaria su instalación en el servidor puesto que forma parte de la solución desarrollada y se despliega de forma automática con la misma. Debe cumplir las siguientes características:</p> <ul style="list-style-type: none"> ▪ Tener creada la estructura de tablas necesaria para su uso. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|---|
| NOMBRE | AUTENTICACIÓN EXTERNA |
| CÓDIGO | RI.009 |
| ACTORES | |
| FUNCIÓN | Restricción de Software sobre proceso de Autenticación Externo |
| DESCRIPCIÓN | Para que el Usuario Anónimo pueda alcanzar el Rol de Usuario Facebook requiere de un proceso de validación como usuario real de Facebook, la aceptación de permisos específicos de acceso a datos del perfil personal y de los grupos de los que es miembro. Este proceso utiliza una interfaz propietaria de Facebook consistente en un pop-up cuya ejecución debe posibilitarse y permitirse para el correcto proceso de autenticación. |
| COMENTARIOS | |
| REFERENCIAS | |

Interfaces de Comunicación.

| | |
|--------------------|---|
| NOMBRE | Comunicación Usuario-Sistema |
| CÓDIGO | RI.010 |
| ACTORES | |
| FUNCIÓN | Requisito de comunicación entre el usuario y el sistema |
| DESCRIPCIÓN | La comunicación entre el usuario y el sistema será mediante el protocolo HTTP y su visualización mediante páginas HTML. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|--|
| NOMBRE | Comunicación Sistema-Base de Datos Relacional |
| CÓDIGO | RI.011 |
| ACTORES | |
| FUNCIÓN | Requisito de comunicación entre el sistema y la base de datos relacional |
| DESCRIPCIÓN | La comunicación entre el sistema y el SGBD relacional se hará mediante el lenguaje de consultas SQL. |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|---|
| NOMBRE | Comunicación Sistema-Base de Datos NoSQL |
| CÓDIGO | RI.012 |
| ACTORES | |
| FUNCIÓN | Requisito de comunicación entre el sistema y la base de datos NoSQL |
| DESCRIPCIÓN | La comunicación entre el sistema y el SGBD NoSQL se hará mediante el lenguaje de consultas Cypher |
| COMENTARIOS | |
| REFERENCIAS | |

| | |
|--------------------|---|
| NOMBRE | Comunicación Sistema-Red Social |
| CÓDIGO | RI.013 |
| ACTORES | |
| FUNCIÓN | Requisito de comunicación entre el sistema y la red social. |
| DESCRIPCIÓN | La comunicación entre el sistema y la red Social Facebook se llevará a cabo haciendo uso de Graph API, la API que Facebook publica para tal fin. Las llamadas sobre la API son llamadas HTTP básicas. |
| COMENTARIOS | |
| REFERENCIAS | |

3.3.2 Funciones

Requisitos funcionales del sistema extrapolados a partir de la agrupación de casos de uso en los grupos funcionales recogidos en el Diagrama 4.

Gestión de Usuarios

| | |
|----------------|---|
| NOMBRE | Identificación como Usuario Facebook |
| CÓDIGO | RF.001 |
| ACTORES | Usuario Anónimo |

| | |
|--------------------|--|
| FUNCIÓN | Inicio de sesión de usuario como usuario Facebook |
| DESCRIPCIÓN | El sistema debe de permitir a un usuario anónimo acceder al diálogo de autenticación facilitado por la API de Facebook, desde el cual introducirá su usuario de Facebook y su contraseña de Facebook. El diálogo de Facebook pedirá que se otorguen determinados permisos de utilización de Facebook con el fin de generar el Token necesario para llevar a cabo las acciones específicas sobre los grupos de Facebook sobre los que se van a extraer o mostrar datos y estadísticas. Los posibles errores devueltos en el proceso de autenticación son errores gestionados por Facebook, no por el sistema. |
| COMENTARIOS | Una vez validado, los permisos aceptados se almacenan en el SGBD relacional como variables de sesión para futuros accesos. |
| REFERENCIAS | CU. 01 |

Gestión de Grupos de Facebook Asociados a la Docencia.

Bienvenido

Es necesario realizar el Log in con su cuenta de facebook para poder acceder a la aplicación.



© 2015 - PFC Facebook

Fig. 5 - Pantalla de bienvenida de la aplicación



Fig. 6 – Inicio de sesión de Facebook

| | |
|--------------------|--|
| NOMBRE | Cerrar Sesión |
| CÓDIGO | RF.002 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Cerrar Sesión de Usuario |
| DESCRIPCIÓN | Se habilitará un botón para cerrar la sesión en la parte superior derecha de la aplicación, junto al nombre del usuario Facebook validado. Al cerrar la sesión se está cerrando la sesión son Facebook, con lo cual no se podrá realizar ninguna acción que utilice la API de Facebook y, por coherencia, tampoco se permitirá consultar la información asociada a los grupos del propio usuario, aunque dicha información ya esté almacenada en el SGBD. También se borrarán las variables de sesión que contengan datos como el grupo de Facebook activo, o grupo sobre el que se realizarán las diferentes acciones de la aplicación. |
| COMENTARIOS | |
| REFERENCIAS | CU. 02 |

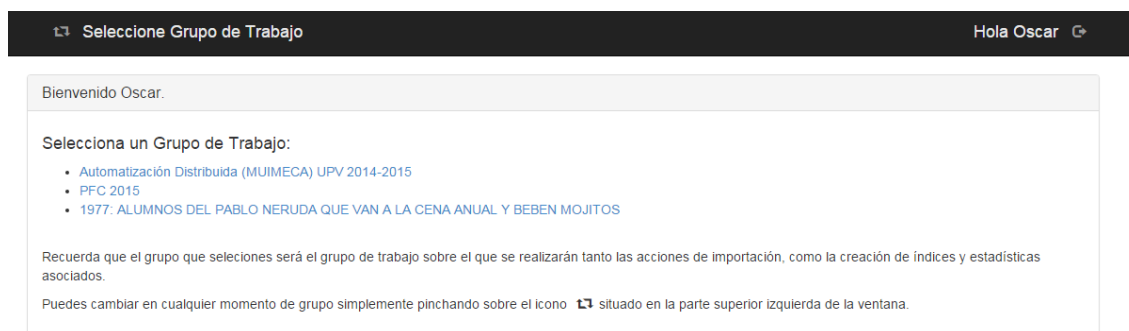


Fig. 7 – Link de cierre de sesión

Gestión de Grupos.

| | |
|--------------------|---|
| NOMBRE | Seleccionar grupo de Facebook como grupo activo |
| CÓDIGO | RF.003 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Seleccionar grupo activo |
| DESCRIPCIÓN | El usuario podrá seleccionar un grupo de trabajo que persistirá durante toda la sesión o hasta que decida cambiarlo. El sistema mostrará un listado de los grupos en los que participa el usuario, para ello se realizará una petición a través de Graph API para recuperar dichos grupos. Al seleccionar el grupo se mostrará el nombre del mismo en la parte superior izquierda de la aplicación, como indicativo visual del grupo que se tiene activo. Si intentara realizar alguna acción sobre grupos sin tener un grupo activo, el sistema lo redirigirá de forma automática a la ventana de selección de grupo. |
| COMENTARIOS | |
| REFERENCIAS | CU. 03 |

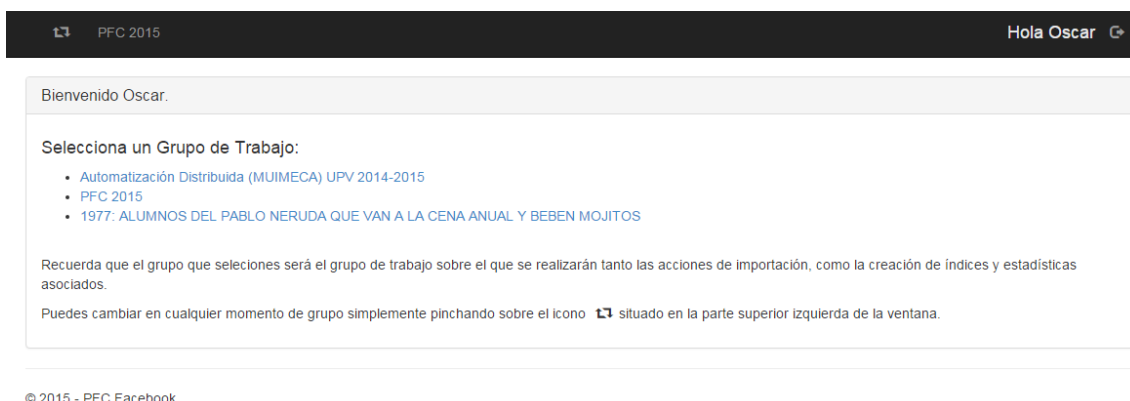


Fig. 8 – Pantalla de selección de grupo activo

| | |
|--------------------|--|
| NOMBRE | Acceso a la ficha del grupo activo |
| CÓDIGO | RF.004 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Acceso a la ficha del grupo activo |
| DESCRIPCIÓN | En la parte superior de la aplicación se mostrará (junto al botón que permitirá seleccionar un grupo activo) el nombre del grupo activo actual. En caso de no haber un grupo seleccionado se mostrará un mensaje indicando que debe seleccionar uno en lugar del nombre del mismo. Este nombre de grupo es un enlace directo a la ficha del grupo, con acceso directo a los datos del mismo (RF. 005), y a las acciones que el usuario podrá realizar |
| COMENTARIOS | |
| REFERENCIAS | CU. 04 |

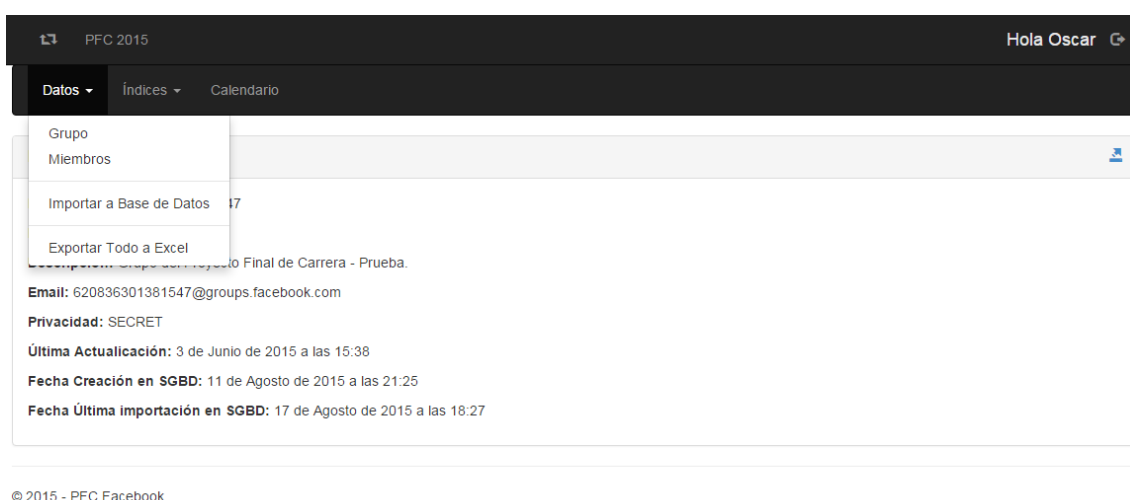


Fig. 9 – Menú de grupo y nombre de grupo activo

| | |
|--------------------|--|
| NOMBRE | Mostrar datos del grupo activo |
| CÓDIGO | RF.005 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Visualiza datos del grupo activo |
| DESCRIPCIÓN | <p>A esta página se accederá tanto desde la función CU.04 (Acceso a la ficha del grupo activo), como des la función CU. 05 (Mostrar datos del grupo activo)</p> <p>Se visualizan datos específicos del grupo activo:</p> <ul style="list-style-type: none"> • Identificador • Nombre • Descripción • Email • Privacidad • Última Actualización • Fecha de creación en el SGBD • Fecha de última importación en el SGBD |
| COMENTARIOS | |
| REFERENCIAS | CU. 05 |

| Datos del Grupo | |
|--|---|
| Identificador: | 620836301381547 |
| Nombre: | PFC 2015 |
| Descripción: | Grupo del Proyecto Final de Carrera - Prueba. |
| Email: | 620836301381547@groups.facebook.com |
| Privacidad: | SECRET |
| Última Actualización: | 3 de Junio de 2015 a las 15:38 |
| Fecha Creación en SGBD: | 11 de Agosto de 2015 a las 21:25 |
| Fecha Última importación en SGBD: | 17 de Agosto de 2015 a las 18:27 |

Fig. 10 – Datos del grupo activo

| | |
|--------------------|--|
| NOMBRE | Exportar a Excel datos básicos del grupo activo. |
| CÓDIGO | RF.006 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar a Excel |
| DESCRIPCIÓN | <p>En la cabecera de la ficha de los datos del grupo, situado en el extremo derecho, se incluirá un botón para exportar a Excel.</p> <p>El sistema realizará una consulta al SGBD recuperando toda la información referente al grupo como entidad individual, sin sus relaciones con miembros, entradas, comentarios u otra información.</p> <p>Se generará una hoja de Excel con los datos obtenidos.</p> <p>El usuario deberá descargar el fichero resultante para poder usarlo.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 06 |

| ID | Nombre | Descripción | Email | ID Propietar | Privacidad | Fecha Últim | Fecha Prim | Fecha Última Importacion |
|------------|--------|---|---|--------------|------------|---------------------|-----------------------------|-----------------------------|
| 6208363013 | | Grupo del Proyecto Final de Carrera - Prueba. | 6208363013 81547@grou ps.facebook.com | 1020558558 | SECRET | 2015-06-03 15:38:22 | 2015-08-11 21:25:05.018 392 | 2015-08-17 18:27:31.280 329 |

Fig. 11 – Excel con datos del grupo activo.

| | |
|--------------------|---|
| NOMBRE | Mostrar miembros del grupo activo |
| CÓDIGO | RF.007 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Visualiza miembros del grupo activo |
| DESCRIPCIÓN | Se recuperarán los miembros del grupo activo almacenados en la base de datos. Es necesario tener en cuenta que estos pueden diferir respecto a los usuarios del grupo de Facebook si hace tiempo que no se ha realizado el proceso de importación al SGBD. Se listan los nombres de los miembros y sus ID's. |
| COMENTARIOS | |
| REFERENCIAS | CU. 07 |

Hola Oscar

Datos - Índices - Calendario

Miembros del Grupo

Actualmente el grupo tiene 6 miembros:

1. **ADMINISTRADOR:** Oscar Ortiz (10205585586346584)
2. Elena Del Val Noguera (10152680742945951)
3. Jose Luis Poza Luján (10205497322284727)
4. Malleus Hildeson (680819395383888)
5. Miguel Rebollo (10152799446507483)
6. Ruben Honrubia Pineda (624427524358041)

© 2015 - PFC Facebook

Fig. 12 - Página de los miembros del grupo

| | |
|--------------------|--|
| NOMBRE | Exportar a Excel los miembros del grupo activo |
| CÓDIGO | RF.008 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar a Excel |
| DESCRIPCIÓN | En la cabecera de la ficha de los miembros del grupo, situado en el extremo derecho, se incluirá un botón para exportar a Excel. El sistema realizará una consulta al SGBD recuperando toda la información referente a los miembros del grupo activo, generando la correspondiente hoja |

| | |
|--------------------|---|
| | de Excel. El usuario deberá descargar el fichero resultante para poder usarlo. |
| COMENTARIOS | |
| REFERENCIAS | CU. 08 |

| ID | Name | Profesor | Fecha Prim | Fecha Ultima Importacion |
|------------|-----------------------|----------|-------------------------|--------------------------|
| 1020558558 | Oscar Ortiz | True | 2015-08-17 18:13:17.898 | 2015-08-17 18:13:17.898 |
| 6346584 | Oscar Ortiz | True | 555 | 555 |
| 1015268074 | Elena Del Val Noguera | False | 2015-08-17 18:13:18.328 | 2015-08-17 18:13:18.328 |
| 2945951 | Elena Del Val Noguera | False | 579 | 579 |
| 1020549732 | Jose Luis Poza Luján | False | 2015-08-17 18:13:17.300 | 2015-08-17 18:13:17.300 |
| 2284727 | Jose Luis Poza Luján | False | 520 | 520 |
| 6808193953 | Malleus Hildeson | False | 2015-08-17 18:13:17.746 | 2015-08-17 18:13:17.746 |
| 83888 | Malleus Hildeson | False | 546 | 546 |
| 1015279944 | Miguel Rebollo | False | 2015-08-17 18:13:18.096 | 2015-08-17 18:13:18.097 |
| 6507483 | Miguel Rebollo | False | 566 | 567 |
| 6244275243 | Ruben Hornubia | False | 2015-08-17 18:13:18.533 | 2015-08-17 18:13:18.533 |
| 58041 | Ruben Hornubia | False | 591 | 591 |

Fig. 13 – Excel con datos de los miembros del grupo activo

| | |
|--------------------|--|
| NOMBRE | Importar desde Facebook a la base de datos los datos del grupo activo |
| CÓDIGO | RF.009 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Importa datos del Grupo |
| DESCRIPCIÓN | <p>El sistema se comunica con Facebook haciendo uso de su api Graph Api y recupera toda l información necesaria sobre el grupo, volcando los datos en el SGBD NoSQL para su posterior consulta.</p> <p>Los datos obtenidos son:</p> <ul style="list-style-type: none"> • Datos del grupo • Miembros del grupo y usuarios que interactúan aunque no pertenezcan al grupo. • Entradas del grupo • Comentarios sobre las entradas. • Eventos del grupo • Relaciones entre las entidades anteriores: <ul style="list-style-type: none"> ○ Usuario es Miembro de grupo ○ Usuario crea entrada ○ Usuario le gusta entrada ○ Usuario comparte entrada ○ Usuario crea comentario ○ Usuario crea evento ○ Entrada pertenece a grupo ○ Entrada pertenece a evento ○ Comentario comenta entrada ○ Evento pertenece a grupo <p>En sistema será capaz de crear nuevos registros y relaciones, así como de detectar modificaciones en las mismas y plasmarlas en el SGBD.</p> <p>Una vez finalizado el proceso se mostrará la ventana de datos del grupo activo</p> |

| | |
|-------------|----------|
| | (CU. 05) |
| COMENTARIOS | |
| REFERENCIAS | CU. 09 |

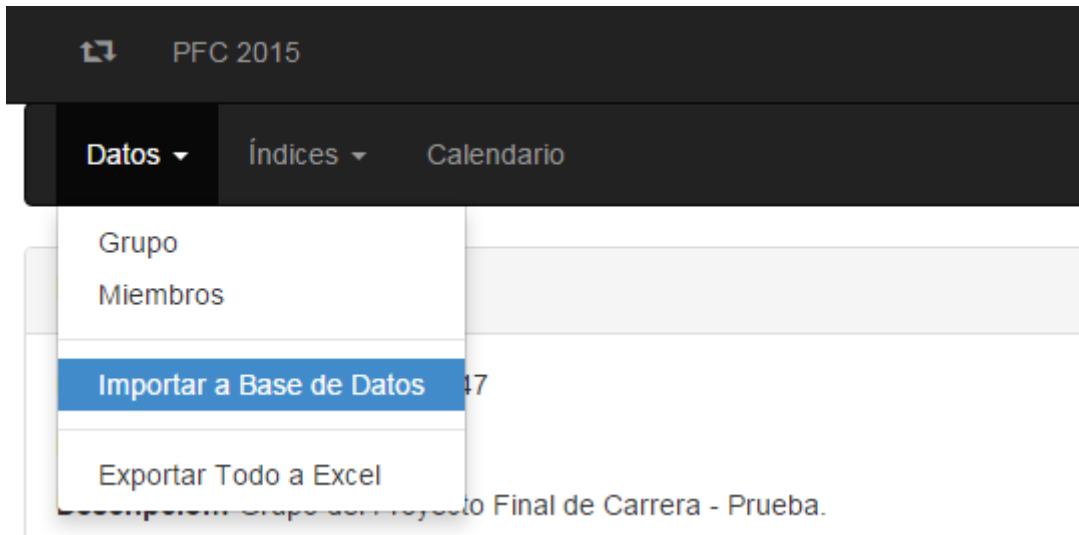


Fig. 14 – Opción de menú Importar a Base de Datos

| | |
|-------------|---|
| NOMBRE | Exportar a Excel Todos los datos del grupo activo |
| CÓDIGO | RF.010 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar a Excel |
| DESCRIPCIÓN | <p>Se recuperarán todos los datos referentes al grupo activo y se generará una hoja de cálculo dentro de un mismo fichero Excel por cada una de las entidades relevantes del mismo:</p> <ul style="list-style-type: none"> • Datos del grupo • Miembros del grupo • Entradas (Posts) • Eventos • Comentarios • Entradas Compartidas • “Me gusta” (Likes) <p>El usuario deberá descargar el fichero resultante para poder usarlo.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 10 |

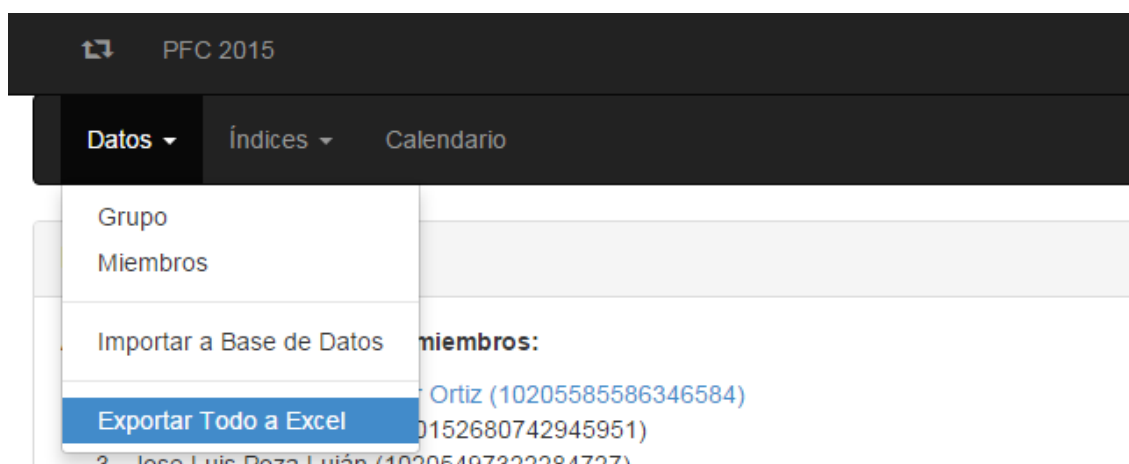


Fig. 15 – Opción de Menú Exportar Todo a Excel

| | |
|--------------------|---|
| NOMBRE | Mostrar todos los índices del grupo activo |
| CÓDIGO | RF.011 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Mostrar índices |
| DESCRIPCIÓN | <p>Se recuperará la información de la Base de Datos para calcular y generar todos los índices implementados en la aplicación, mostrándolos todos en una misma página, sin texto descriptivo ni la literatura asociada a los mismos necesaria para su comprensión e interpretación.</p> <p>Inicialmente la aplicación constará de tres índices:</p> <ul style="list-style-type: none"> • Índice de actividad del grupo (IAGr) • Índice de actividad de los alumnos (IAAl) • Índice de impacto de las publicaciones (IIPb) |
| COMENTARIOS | |
| REFERENCIAS | CU. 11 |

| | |
|--------------------|---|
| NOMBRE | Exportar a Excel datos de todos los índices del grupo activo |
| CÓDIGO | RF.012 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar a Excel |
| DESCRIPCIÓN | <p>En la cabecera de la ficha de los índices, situado en el extremo derecho, se incluirá un botón para exportar a Excel.</p> <p>El sistema realizará una consulta al SGBD recuperando toda la información referente a los índices, los calculará, y generará dos hojas de cálculo, dentro de un mismo fichero Excel, para cada uno de los índices implementados:</p> <ul style="list-style-type: none"> • Una hoja con los datos calculados que se muestran en la gráfica. • Una hoja con los datos utilizados para calcular los índices. (Entradas, comentarios, “me gusta”, etc...) <p>El usuario deberá descargar el fichero resultante para poder usarlo.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 12 |

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|---------------------|----------|---|---|-----------------------|------------|------------------------|-----------------------------------|-----------------------------------|-------------|---|---|
| 1 | ID | Nombre | Descripcion | Email | ID Propietar | Privacidad | Fecha Ultim | Fecha Prime | Fecha Ultima | Importacion | | |
| 2 | 6208363013 81547 | PFC 2015 | Grupo del Proyecto Final de Carrera - Prueba. | 6208363013 81547@grou ps facebook. com | 1020558558 6346584 | SECRET | 2015-06-03 15:38:22 | 2015-08-11 21:25:05.018 392 | 2015-08-17 18:27:31.280 329 | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | |
| 33 | | | | | | | | | | | | |
| 34 | | | | | | | | | | | | |

Fig. 16 – Excel con todos los datos del grupo

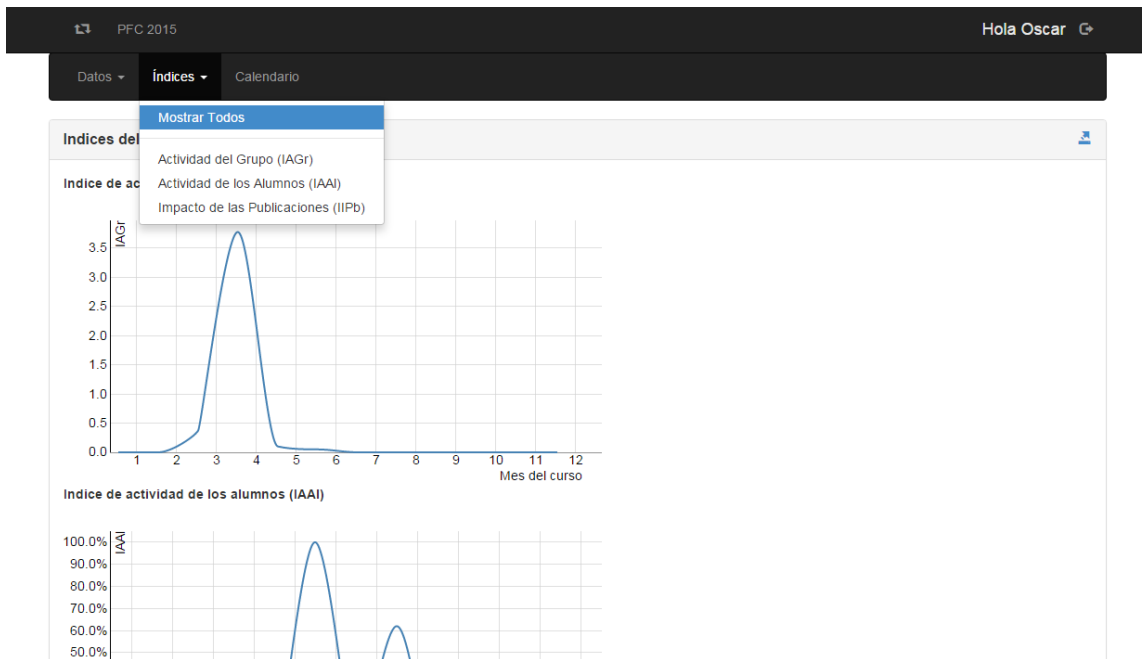


Fig. 17 – Opción de menú Mostrar Todos los índices y página resultante

| | A | B | C | D | E | F | G | H | I | J | K |
|----|-----|-------|---|---|---|---|---|---|---|---|---|
| 1 | Mes | Valor | | | | | | | | | |
| 2 | 1 | 0.0 | | | | | | | | | |
| 3 | 2 | 0.0 | | | | | | | | | |
| 4 | 3 | 0.37 | | | | | | | | | |
| 5 | 4 | 3.77 | | | | | | | | | |
| 6 | 5 | 0.1 | | | | | | | | | |
| 7 | 6 | 0.05 | | | | | | | | | |
| 8 | 7 | 0.0 | | | | | | | | | |
| 9 | 8 | 0.0 | | | | | | | | | |
| 10 | 9 | 0.0 | | | | | | | | | |
| 11 | 10 | 0.0 | | | | | | | | | |
| 12 | 11 | 0.0 | | | | | | | | | |
| 13 | 12 | 0.0 | | | | | | | | | |
| 14 | | | | | | | | | | | |
| 15 | | | | | | | | | | | |
| 16 | | | | | | | | | | | |
| 17 | | | | | | | | | | | |
| 18 | | | | | | | | | | | |
| 19 | | | | | | | | | | | |
| 20 | | | | | | | | | | | |
| 21 | | | | | | | | | | | |
| 22 | | | | | | | | | | | |
| 23 | | | | | | | | | | | |
| 24 | | | | | | | | | | | |
| 25 | | | | | | | | | | | |
| 26 | | | | | | | | | | | |
| 27 | | | | | | | | | | | |
| 28 | | | | | | | | | | | |
| 29 | | | | | | | | | | | |
| 30 | | | | | | | | | | | |
| 31 | | | | | | | | | | | |
| 32 | | | | | | | | | | | |
| 33 | | | | | | | | | | | |
| 34 | | | | | | | | | | | |
| 35 | | | | | | | | | | | |
| 36 | | | | | | | | | | | |
| 37 | | | | | | | | | | | |
| 38 | | | | | | | | | | | |

Fig. 18 – Excel con todos los índices.

| | |
|--------------------|---|
| NOMBRE | Mostrar información de un solo índice del grupo activo |
| CÓDIGO | RF.013 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Mostrar índices |
| DESCRIPCIÓN | <p>Se mostrará un submenú para mostrar la información de cada uno de los índices implementados:</p> <ul style="list-style-type: none"> • Índice de actividad del grupo (IAGr) • Índice de actividad de los alumnos (IAAl) • Índice de impacto de las publicaciones (IIPb) <p>Dentro de la ficha de cada índice se incluirá la información necesaria para entender la lógica del mismo, facilitando al usuario la interpretación de los datos mostrados en la gráfica generada.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 13 |

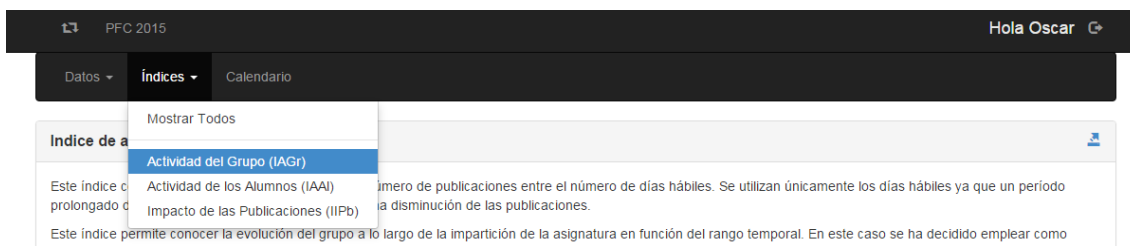


Fig. 19 – Opción de menú por índice individual.



Fig. 20 – Página de índice individual.

| | |
|--------------------|--|
| NOMBRE | Exportar a Excel datos de un índice |
| CÓDIGO | RF.014 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Exportar a Excel |
| DESCRIPCIÓN | <p>En la cabecera de la ficha del índice, situado en el extremo derecho, se incluirá un botón para exportar a Excel.</p> <p>El sistema realizará una consulta al SGBD recuperando toda la información referente al índice, lo calculará, y generará dos hojas de cálculo, dentro de un mismo fichero Excel:</p> <ul style="list-style-type: none"> • Una hoja con los datos calculados que se muestran en la gráfica. • Una hoja con los datos utilizados para calcular los índices. (Entradas, comentarios, “me gusta”, etc...) <p>El usuario deberá descargar el fichero resultante para poder usarlo.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 14 |

Obtención, procesamiento y visualización de datos de Facebook.

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|------------|---------------------|-----------------------|---|-------------|------------------------|-------------|---|---|---|---|---|---|
| 1 | Tipo | ID | Usuario | Texto | Es Profesor | Fecha Creac | Total Likes | | | | | | |
| 2 | Comentario | 6356265065 69193 | 6244275243 58041 | Se intenta. Ahora que he acabado la formación a ver si puedo enseñar el informe que tengo. | False | 2015-04-24 19:37:09 | 1 | | | | | | |
| 3 | Comentario | 6355848965 73354 | 1020558558 6346584 | No pasa nada, otro día será. Iré poniendo las dudas por aquí, así me sirve también como datos para el proyecto. | True | 2015-04-24 17:40:19 | 0 | | | | | | |
| 4 | Comentario | 6356379399 01383 | 1020558558 6346584 | Pon aquí el pantallazo, más datos. La participación activa es bien recibida. | True | 2015-04-24 20:04:41 | 1 | | | | | | |
| 5 | Comentario | 6319526669 36577 | 1020558558 6346584 | No se tiene en cuenta la actividad de los eventos. | True | 2015-04-17 11:04:14 | 2 | | | | | | |
| 6 | Comentario | 6350735832 91152 | 1020558558 6346584 | *Debería actualizar los comentarios. | True | 2015-04-23 13:22:38 | 1 | | | | | | |

Fig. 21 – Excel de índice individual

| | |
|--------------------|---|
| NOMBRE | Modificar el calendario lectivo del grupo activo. |
| CÓDIGO | RF.015 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Modificar calendario lectivo |
| DESCRIPCIÓN | <p>Si bien los calendarios lectivos se almacenan de forma independiente al grupo (en la base de datos relacional), se permitirá modificar únicamente los días lectivos referentes al curso académico que abarque el periodo lectivo del grupo.</p> <p>Esta modificación se realizará para un curso académico, no para el grupo específico, con lo que modificaciones por parte de usuarios de grupos diferentes que compartan el mismo curso académico podrían modificar los días del propio grupo de trabajo.</p> <p>No se considera crítica la modificación de los días lectivos, de ahí que no se considere relevante el limitar dicha acción con permisos de acceso especiales. El sistema mostrará la información de los días introducidos previamente por algún usuario. En caso de no existir esta introducción previa, el sistema creará un registro inicializando los días lectivos de cada mes a un valor predeterminado de 20 (una media aceptable).</p> <p>En cualquier caso, esta creación automática también será aplicada por cualquier índice que necesite dichos días para calcularse.</p> |
| COMENTARIOS | |
| REFERENCIAS | CU. 15 |

PFC 2015 Hola Oscar

Datos ▾ Índices ▾ **Calendario**

Calendario 2014-2015

Introduzca los días lectivos del calendario escolar.

Estos días se utilizan para realizar cálculos de índices y estadísticas siendo, por tanto, un dato necesario y muy relevante que puede llegar a desvirtuar los resultados finales en caso de no ser correctos.

Los meses se muestran en orden natural (de Enero a Diciembre), aunque debe tener en cuenta que los primeros meses del año (de Enero a Agosto) corresponden al tramo final del año lectivo, y el resto al tramo inicial del mismo (de Septiembre a Diciembre)

| | | | | | | | |
|------------|----|---------|----|-----------|----|-----------|----|
| Enero | 18 | Febrero | 20 | Marzo | 19 | Abril | 13 |
| Mayo | 20 | Junio | 22 | Julio | 23 | Agosto | 0 |
| Septiembre | 22 | Octubre | 21 | Noviembre | 20 | Diciembre | 15 |

Guardar

© 2015 - PFC Facebook

Fig. 22 – Página de modificación de días lectivos del calendario del grupo

3.3.3 Requisitos de Rendimiento

| | |
|--------------------|---|
| NOMBRE | Uso Concurrente |
| CÓDIGO | RNF.001 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Uso concurrente de la aplicación |
| DESCRIPCIÓN | El sistema debe garantizar un uso concurrente de la aplicación y del acceso en modo lectura y escritura de los datos del grupo. |
| COMENTARIOS | |
| REFERENCIAS | |

3.3.4 Restricciones de Diseño

No existen requisitos o restricciones del sistema aplicables a esta categoría.

3.3.5 Atributos del Sistema

No existen requisitos de fiabilidad, mantenibilidad o portabilidad aplicables al sistema.

| | |
|--------------------|---|
| NOMBRE | Contraseñas |
| CÓDIGO | RNF.002 |
| ACTORES | Usuario Facebook |
| FUNCIÓN | Contraseñas del Usuario Facebook |
| DESCRIPCIÓN | De la conexión con Facebook sólo se guarda un token de acceso que incluye todos los privilegios y permisos que requiere la aplicación, no almacenándose en ningún caso los datos de conexión. |
| COMENTARIOS | |
| REFERENCIAS | |

3.3.6 Otros Requisitos

No existen requisitos o restricciones del sistema aplicables a esta categoría.

3.4 Apéndices

No hay información relevante para ser incluida en esta sección.

4 Diseño

4.1 Introducción

Una vez realizada la especificación de requisitos, y tomando como referencia los requisitos funcionales detectados, podemos entrar a definir, tanto conceptual como formalmente, los diferentes elementos necesarios para llevar a cabo la implementación de la aplicación.

Es esta sección se presentarán los diagramas de clases necesarios para entender con suficiente claridad la lógica a implementar, así como los diagramas de secuencias para las funcionalidades de carácter más dinámico, centrándose con especial detalle en los algoritmos más complejos y de mayor peso dentro del desarrollo global de la aplicación.

4.2 Especificación conceptual

Analizada la aplicación lo primero que encontramos es un grupo de clases bien diferenciado, y un pequeño subconjunto de clases que tienen más preponderancia en el apartado visual o en la interfaz de usuario.

Puesto que el desarrollo planteado consiste, en gran medida, en la obtención de los datos de grupos de Facebook y su posterior inclusión en una base de datos NoSQL, el algoritmo o proceso encargado de esta importación resulta de una complejidad tal, y relaciona tantas entidades diferentes, que creemos requiere su propio diagrama de clases.

Frente a esto encontramos una segunda aproximación desde el punto de vista del usuario de la aplicación, usuario que si bien debe pertenecer al grupo y, por tanto, estará incluido como tal en el diagrama relativo al proceso de importación, no tiene una relación directa con determinadas clases a la hora de ejecutar las diferentes acciones de menú, de forma que aunque se podrían incluir todas las clases en un mismo diagrama, consideramos que queda mucho más claro si se separan en dos: por un lado el proceso de importación, con el grupo como centro de acción; y por otro el de la interfaz de usuario con este como protagonista.

De esta forma, y a modo informativo, encontraremos una serie de clases (Fig. 50) que introducimos a continuación:

- **Group:** representa el grupo de Facebook y es, por tanto, el núcleo principal del proceso de importación de datos, de ahí su relevancia y que contenga la mayoría de operaciones requeridas.
- **User:** representa a los usuarios Facebook (y por extensión a los usuarios de la aplicación) que pueden pertenecer a un grupo, o haber generado algún tipo de

actividad como usuario invitado sin llegar a pertenecer a él. Son los generadores de contenido por excelencia.

- **Post:** cada una de las entradas generadas en un grupo, bien sea texto, fotos, vídeos o cualquier otro tipo de contenido que un usuario cree en el grupo.
- **Comment:** es el contenido creado por un usuario como respuesta a una entrada o evento. Este puede ser, al igual que las entradas, tanto texto como fotos, vídeos, etc...
- **Event:** representa los eventos de Facebook. Los eventos son tratados prácticamente como un subgrupo dentro del grupo, conteniendo sus propios participantes, sus entradas y sus comentarios.
- **Index:** este es la primera clase con referencia más visual que de lógica de importación. Un usuario podrá genera un índice determinado y visualizarlo por pantalla.
- **Calendar:** representa el calendario lectivo de un determinado año académico, se asocia a un grupo en tanto en cuanto dicho grupo está incluido dentro del periodo lectivo del calendario.

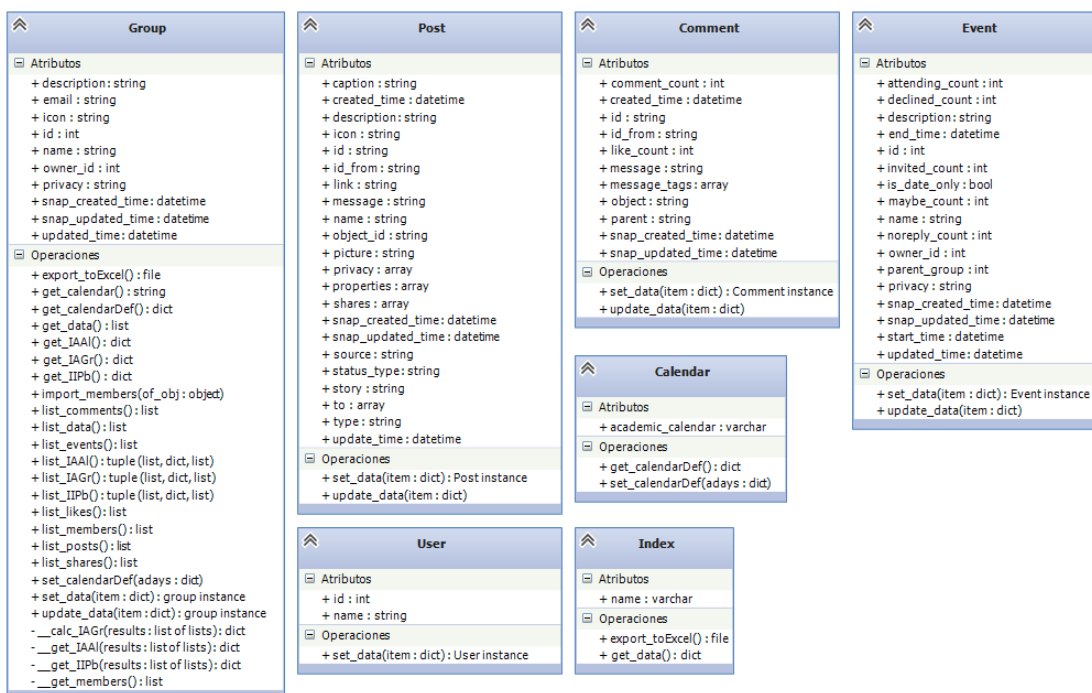


Fig. 23 – Clases con sus atributos y operaciones.

A partir de la definición de las clases podemos generar los correspondientes diagramas (Fig. 24, Fig. 25), mostrando únicamente los nombres de clases para economizar espacio y que sea más legible:



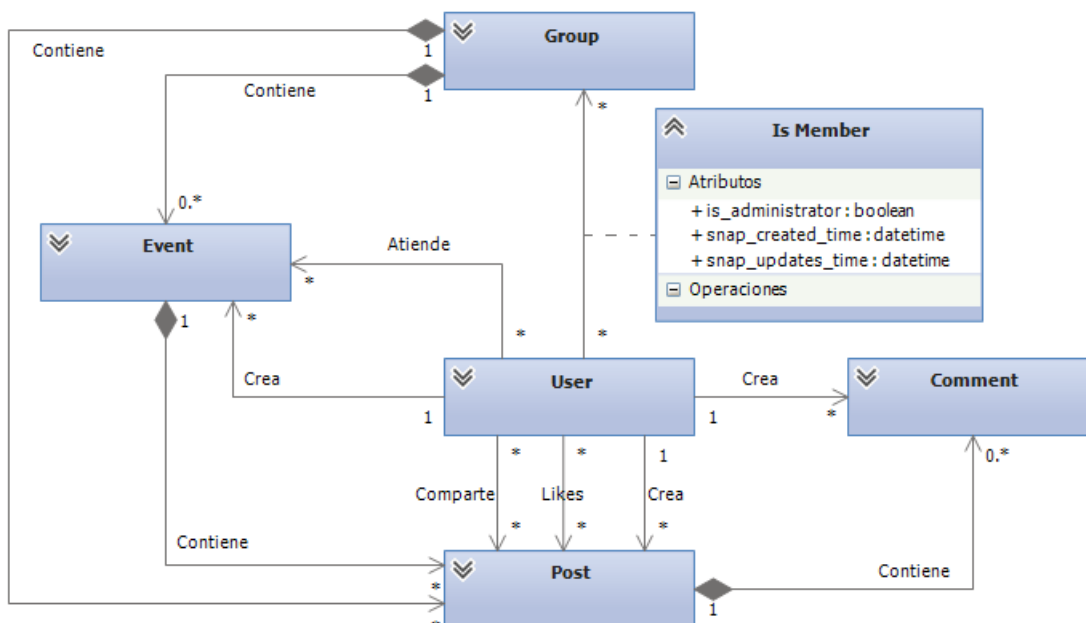


Fig. 24 – Diagrama de clases de los datos estáticos del grupo de Facebook.

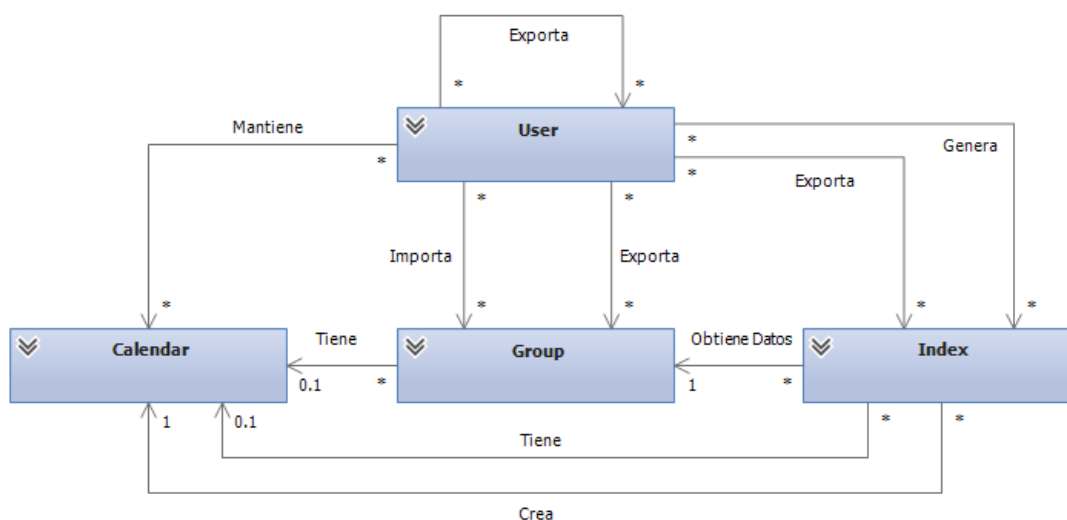


Fig. 25 – Diagrama de clases Interfaz Usuario.

Una vez presentada la especificación conceptual sería también conveniente incluir los diagramas entidad-relación de las bases de datos que se van a utilizar en la implementación, sin embargo, y dado que uno de los Sistemas de Gestión de Base de Datos no sigue el modelo relacional común, es complicado plasmar en un diagrama el contenido de la misma. De todas formas, el propio diagrama de clases que hace referencia al proceso de importación podría servir como modelo conceptual de la base de datos, teniendo en cuenta que la mayoría de relaciones entre las entidades (user, group, comment, event y post) tienen características asociadas, como, por ejemplo, la fecha de creación de la propia relación.

De esta forma, y siguiendo el diagrama propuesto, entre User y Post existirían tres relaciones directas Comparte, Likes y Crea, cada una de las cuales guardaría la fecha en la que se creó esta relación entre las entidades.

A modo de ejemplo se muestra la forma de construir la entidad Post desde Python utilizando el módulo Neomodel para trabajar con Neo4j (Fig. 26). Se pueden observar tanto los campos específicos de la entidad, como las relaciones definidas con otras entidades (RelationshipTo).

```
class Post(StructuredNode):
    id = StringProperty(unique_index=True, required=True)
    type = StringProperty()
    name = StringProperty()
    caption = StringProperty()
    description = StringProperty()
    message = StringProperty()
    story = StringProperty()
    id_from = StringProperty()
    to = ArrayProperty()
    created_time = DateTimeProperty()
    update_time = DateTimeProperty()
    icon = StringProperty()
    link = StringProperty()
    object_id = StringProperty()
    picture = StringProperty()
    privacy = ArrayProperty()
    properties = ArrayProperty()
    shares = ArrayProperty()
    source = StringProperty()
    status_type = StringProperty()
    snap_created_time = DateTimeProperty()
    snap_updated_time = DateTimeProperty(default=lambda: datetime.now(pytz.utc))
    pertains_group = RelationshipTo('Group', 'PERTAINS', model=PertainsRel)
    pertains_event = RelationshipTo('Event', 'PERTAINS', model=PertainsRel)
```

Fig. 26 – Definición de una Entidad para la base de datos creada en Neo4j.

Por otro lado sí se requiere el uso de una base de datos relacional para almacenar datos de carácter más estático como es el caso de los calendarios lectivos. Para ello se propone el uso de dos tablas muy simples con una relación de clave ajena entre ambas (Fig. 27).

El modelo en este caso es tremendamente simple puesto que lo único que se requiere es guardar los días naturales y los días lectivos para cada mes de un periodo lectivo, de esta forma para un curso “2014-2015” existirán 12 registros en la correspondiente tabla CalendarDef.

Continuaremos presentando el diagrama de secuencia del proceso más complejo de la aplicación, el proceso de importación que recupera información del Facebook haciendo uso de su API (Graph API), y la crea o actualiza en la base de datos creada para tal fin.

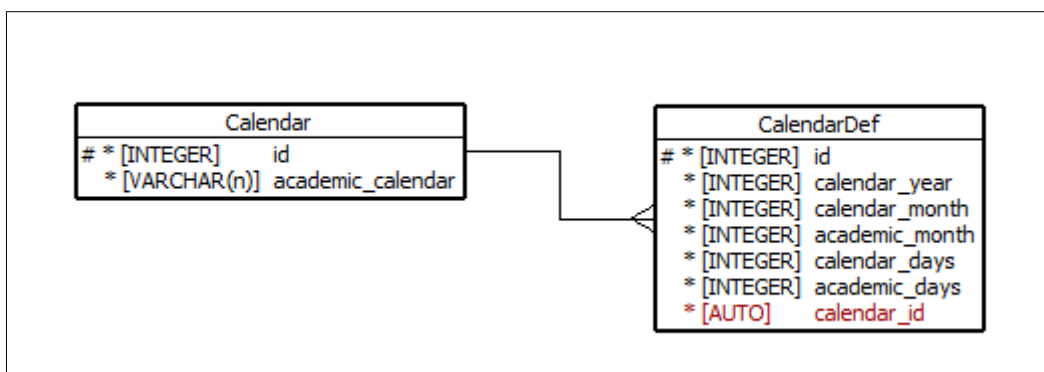


Fig. 27 – Diagrama Entidad-Relación para los calendarios.

Puesto que el proceso es complejo se ha optado por separarlo en varios diagramas que en realidad representarán un mismo marco temporal. Para ello se presenta un primer diagrama (Fig. 28) que muestra el algoritmo de forma resumida y, posteriormente, se introduce un diagrama de secuencia para cada una de las llamadas incluidas en este.

La primera parte del algoritmo es la importación de miembros a la base de datos (Fig. 29). Esta importación utiliza una aproximación de consulta agrupada, de forma que en una única llamada HTTP se pide la información de todos los miembros del grupo (en lugar de una llamada HTTP por miembro). En este proceso la información obtenida de Facebook se compara con los miembros de la base de datos, creándolos en caso de no existir, actualizando su información en caso de haber cambiado, y estableciendo las correspondientes relaciones de pertenencia al grupo (puede darse el caso de que el usuario exista en la base de datos, pero no esté relacionado como miembro de este grupo en concreto).

Un dato relevante de este primer algoritmo es la necesidad de realizar una importación recursiva de los datos ya que Facebook limita el volumen de datos a devolver por cada llamada, de forma que hay que realizar tantas llamadas HTTP como sean necesarias para recuperar todos los datos sobre los miembros. En el caso, por ejemplo, de un grupo de 25 miembros se requiere de dos llamadas para obtener los identificadores de todos ellos. Este mismo proceso se realizará en otras consultas a la API de Facebook.

La segunda parte del algoritmo es la importación de las entradas del propio grupo (Fig. 30), lo que en Facebook se obtiene empleando la arista “Feed”. Se obtiene un conjunto de elementos de tipo entrada (Post) y para cada uno de ellos se intenta recuperar su correspondiente registro de la base de datos, creándolo en caso de no existir y actualizando los campos que hayan cambiado en caso de que si exista. Posteriormente se crean las correspondientes relaciones entre la entrada y el grupo al que pertenece, y entre el usuario que lo ha creado o compartido en el grupo. También se recuperan los “me gustas” de la entrada y se genera la relación entre la entrada y el usuario que ha dado el “me gusta”. Y, por último, se recuperan los comentarios de

la entrada, creándolos o actualizando la información de cada comentario en la base de datos, y estableciendo las relaciones entre el comentario y la entrada que comenta, y el comentario y el usuario que lo ha creado.

Todas las consultas a la API de Facebook son recursivas, sin embargo en el diagrama se va a obviar dicha representación con el fin de hacerlo más compacto y fácil de leer.

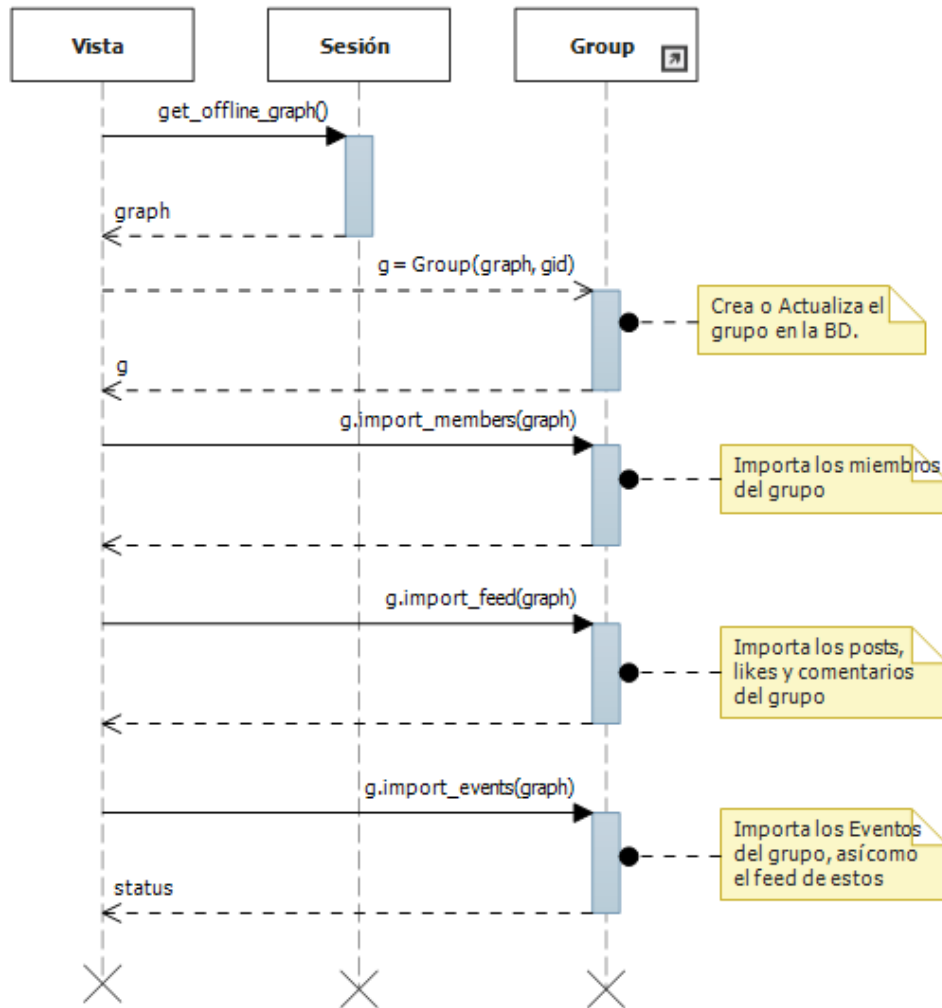


Fig. 28 – Importación del grupo.

La tercera y última parte del algoritmo es la importación de los eventos del grupo (Fig. 31). Un evento es un nodo de Facebook que funciona prácticamente igual que un grupo, con su propio “Feed” sus propias entradas, comentarios y “me gusta”. Así pues, el diagrama de secuencia de esta parte del algoritmo constará de una parte específica de importación de los datos básicos del propio evento, y un algoritmo de obtención de datos que es casi un calco del import_feed del grupo, con la excepción de que las relaciones de pertenencia se crearán con el evento, y no con el grupo. Esto es, las entradas del feed tendrán una relación Post-Pertains-Event, en lugar de una relación Post-Pertains-Group. Por tanto, y con el fin de economizar

información, no se va a volver a crear un diagrama de secuencia, sirviendo perfectamente el ya mencionado del grupo (Fig. 30).

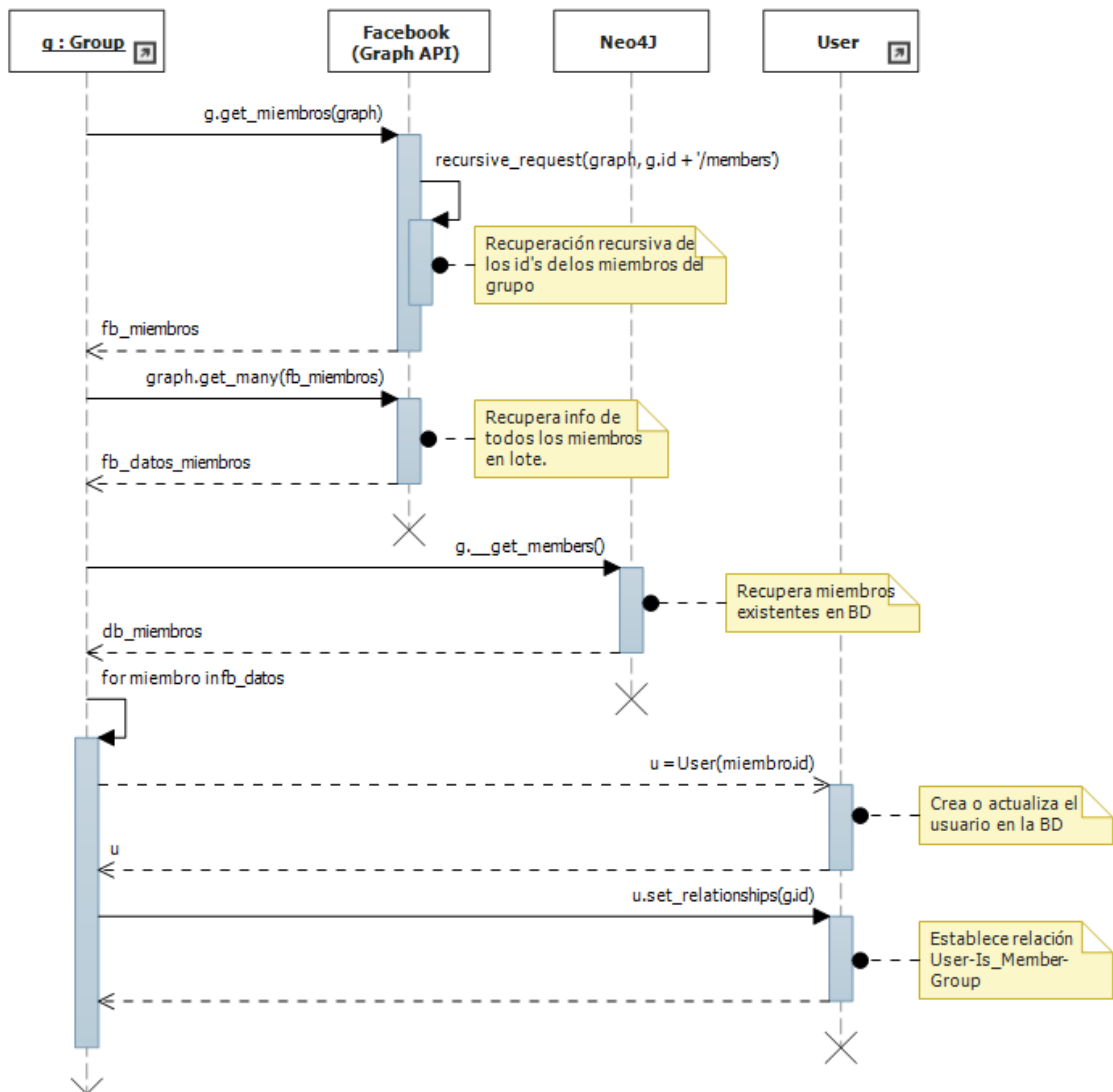


Fig. 29 – Importación de miembros (import_members).

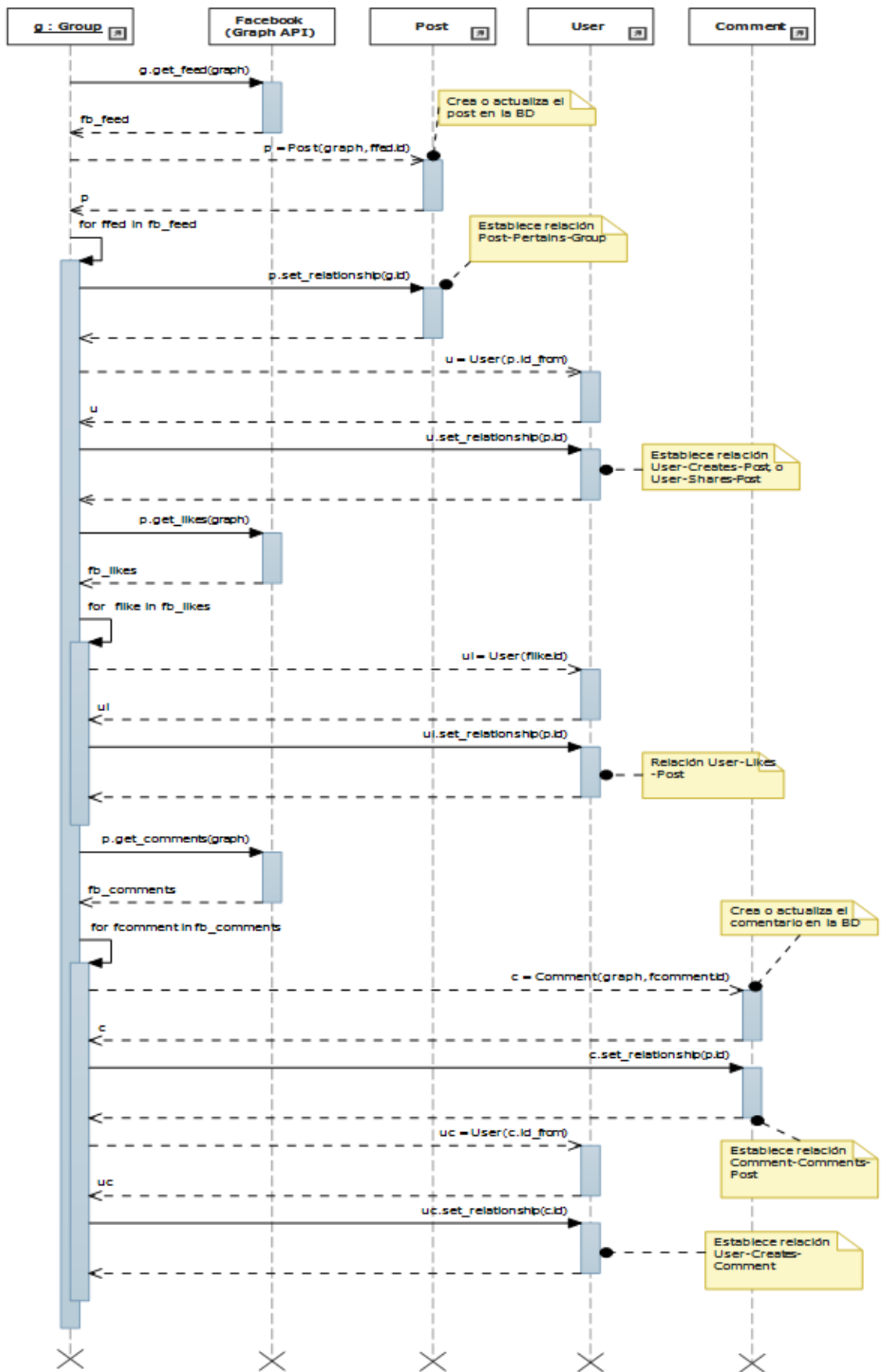


Fig. 30 – Importación del feed (import_feed).

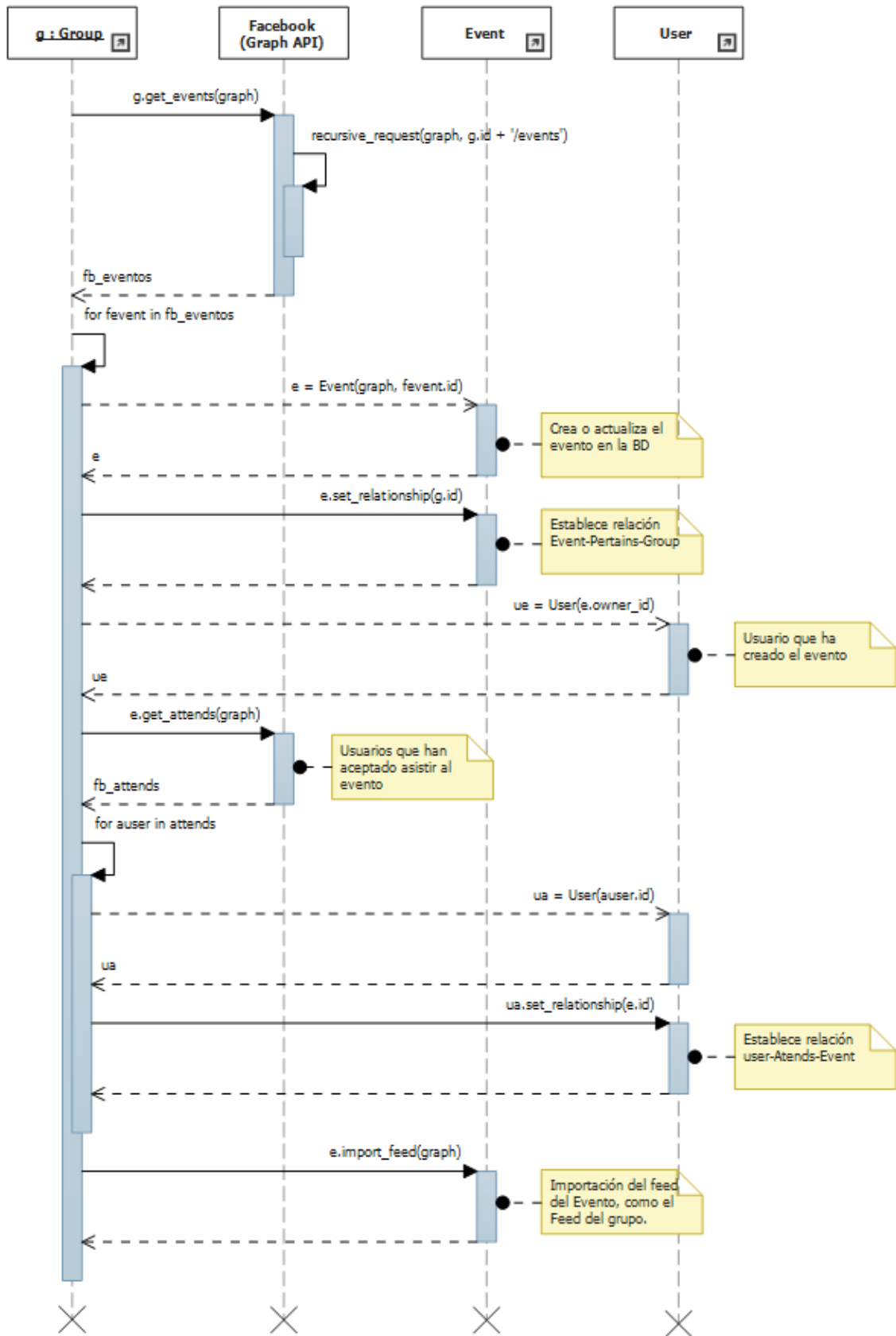


Fig. 31 – Importación de eventos del grupo (import_events).

Estos diagramas representan el núcleo complejo de la aplicación, pero además existen una serie de procesos más ligeros que, casi en su totalidad, consisten en una simple tarea de obtener información de la base de datos a través de una consulta con el lenguaje específico del sistema de gestión de base de datos (Cypher), y la posterior visualización de los mismos. Son, por tanto, procesos tan sencillos que no requieren de un modelado conceptual para su comprensión y posterior implementación.

En este conjunto entrarían procesos como por ejemplo el de recuperación de los grupos del usuario para su selección como grupo de trabajo activo, la recuperación de los datos del grupo, o de los miembros del mismo para su posterior visualización e, incluso, la generación de los índices que requiere de unos muy simples cálculos previos a la visualización de la gráfica (divisiones entre meses y poco más).

En estos casos casi resulta más simple la propia codificación que la generación de los diagramas que los definen.

4.3 Conclusiones

Se ha presentado el modelado de la aplicación utilizando diagramas de clases y diagramas de secuencia, con la suficiente granularidad como para entender los procesos realmente complejos de la misma, pero sin llegar a generar una documentación demasiado extensa y compleja de leer o interpretar.

Siguiendo esta idea se puede detectar un proceso más complejo, el de la importación de los datos de Facebook a la Base de Datos, y una serie de procesos satélites que, si bien enriquecen el uso de la aplicación, no encierran tanta complejidad como para definir su lógica interna, ni exigiendo un análisis exhaustivo para usarlo de guía en el paso de implementación de la misma.

5 Implementación e Implantación

5.1 Introducción

En los siguientes apartados de esta sección procederemos a mencionar y detallar las tecnologías, metodologías y herramientas elegidas para la implementación de la aplicación, prestando especial atención a aquellas que consideremos menos conocidas o cuya difusión no sea tan relevante.

Además se presentará la aplicación desde el punto de vista del usuario, mostrando sus funcionalidades y la experiencia de uso de la misma.

5.2 Sistema Desarrollado

La aplicación desarrollada se ha llevado a cabo bajo la idea de permitir un acceso completo a cualquier usuario haciendo uso de un simple explorador web, sin verse atado a un entorno de trabajo específico o incluso a un sistema operativo concreto.

Esto permite una gran independencia de la plataforma para el usuario, requiriendo simplemente un dispositivo con acceso a internet y, eso sí, una cuenta de Facebook propia para poder validarse en la aplicación.

No obstante, a nivel de desarrollo, se ha optado por una política híbrida de herramientas y tecnologías tanto de código libre como propietarias, convirtiendo la experiencia de trabajo en un objetivo secundario del proyecto. Así, aun eligiendo Python y Django como lenguajes de referencia, se ha utilizado el entorno de desarrollo por excelencia de Microsoft como entorno de trabajo principal.

A continuación se mencionan las principales tecnologías, conceptos y herramientas empleados en la implementación de la aplicación.

Python, Django y el paradigma MVC

El lenguaje de programación elegido para implementar la aplicación es Python, un lenguaje interpretado multiplataforma con licencia de código abierto, el cual permite la utilización de diferentes paradigmas de programación (orientación a objetos, programación imperativa, etc...).

Puesto que es un lenguaje sobradamente conocido, y de amplia difusión en entornos de código abierto y educacionales, no nos explayaremos demasiado en nombrar sus características. Lo relevante es tener en cuenta que, dada su condición de lenguaje interpretado, necesitaremos tener instalado y perfectamente configurado (con todos los paquetes necesarios), el intérprete en la máquina que vaya a ejecutar la aplicación.

Aunque, para ser fiel a la ideología de Python, qué menos que mencionar sus principios básicos:

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.¹⁵
- Ahora es mejor que nunca.
- Aunque *nunca* es a menudo mejor que *ya mismo*.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (*namespaces*) son una gran idea ¡Hagamos más de esas cosas!

Dentro de la elección de Python como lenguaje, y aprovechando su condición de lenguaje multiparadigma, se ha optado por utilizar Django como framework de desarrollo.

Django está escrito en Python, requiere de un paquete específico para su utilización, y permite la creación de sitios web complejos respetando el patrón de diseño conocido como Modelo-Vista-Controlador (MVC).

“Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself)

[[WIKIPEDIA](#), Django]

Aunque, si bien es cierto que respeta el modelo MVC, la nomenclatura empleada difiere ligeramente respecto al estándar, de manera que lo que comúnmente se conoce como Controlador, en Django se llama Vista (View), y lo que en el estándar se trata como Vista, en Django se llama Plantilla (Template). Esto no deja de ser más que una declaración de intenciones, puesto que sus desarrolladores nunca pretendieron amoldarse estrictamente a ningún paradigma establecido, siendo quizás más acertado decir que usan un patrón MTV (Model-Template-View)

No obstante, para entender las ramificaciones de dicha distinción entre modelos es necesario conocer previamente cómo está diseñado el modelo MVC.

“Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.” ¡ERROR! NO ENCUENTRA EL ORIGEN DE LA REFERENCIA.

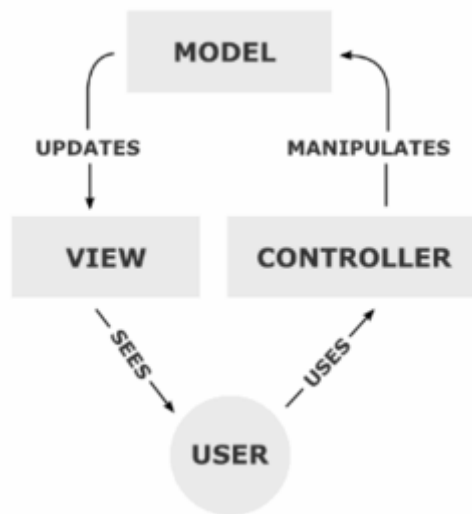


Fig. 32 – Colaboración entre los componentes de un MVC

Así pues tenemos tres elementos perfectamente diferenciados y cada uno con su propia funcionalidad dentro del patrón:

- **Modelo:** Es la representación de la información de la cual hace uso la aplicación, una especie de abstracción de la capa de datos que se encarga de recibir peticiones del controlador, procesarlas, y devolver la información resultante para su posterior representación hacia el usuario.
- **Vista:** es la encargada de representar visualmente los datos para garantizar que el usuario interactúe de forma correcta con la aplicación. Engloba la capa de presentación de los datos obtenidos a través del Modelo.
- **Controlador:** Es el intermediario ente la Vista y el Modelo. Se encarga de recibir las peticiones de información o acciones de la aplicación, traducirlas en consultas al Modelo para, posteriormente, redirigir la información obtenida a la Vista.

Este modelo, por tanto, permite la reutilización de código y la separación de conceptos, lo cual lo hace una herramienta realmente útil para procesos complejos como los incluidos en este proyecto.

Paquetes Python

Una de las grandes virtudes de Python es la posibilidad de incluir paquetes para aprovecharse de funcionalidad programada y sobradamente contrastada por otros desarrolladores y usuarios. Sin embargo, esta característica que potencia y facilita enormemente el proceso de desarrollo, puede desembocar en auténticos quebraderos de cabeza a la hora de intentar dar con la configuración correcta entre las diferentes versiones de cada uno de los paquetes y del propio núcleo de Python.

Actualmente hay dos versiones de Python descargables: la 2.7, que se mantiene por compatibilidad con aplicaciones y paquetes que todavía usan esta versión, y la 3.4 que, si bien el salto cualitativo frente a la anterior es significativo, supone un reto a la hora de organizar un todo estable si optamos por emplear módulos de terceros (muchos de ellos compatibles sólo con la versión 2.7)

En nuestro caso hemos elegido la versión 3.4 por entender que el desarrollo de una nueva aplicación siempre debe estar sustentado tanto en la estabilidad, como en la innovación y en el aprovechamiento de todas las nuevas características.

Partiendo de esta versión de Python se intentó emplear la versión 1.8 de Django (la última versión descargable), sin embargo, la inclusión del paquete `django-facebook`, y su falta de compatibilidad con esta versión, nos obligó a utilizar Django 1.7

Tomando como referencia tanto la versión de Python como la de Django, que son los dos pilares del proyecto, se han utilizado los siguientes paquetes:

- Django-facebook (6.0.2) [[SCHELLENBACH](#), Thierry. *Django*]
Es el paquete que gestiona todo el proceso de validación y gestión de Tokens de Facebook. Se encarga de dar acceso al botón de conexión de Facebook, gestionar las sesiones usando una estructura de tablas propias creadas en una base de datos en SQLite3 integrada en la propia solución de Django, y de extender la validez de los Tokens de acceso que emplea Facebook para poder acceder a su API y, por ende, a los datos del grupo.
Por defecto Facebook genera Tokens con una duración muy limitada (de un par de horas), siendo este módulo el que se encarga de extender su vida útil a unos 60 días, manteniendo para ello un objeto “Open Facebook” por usuario, objeto necesario para interactuar con la API de Facebook.



- Neomodel [[EDWARDS, Robin.](#)]
Es el paquete empleado para acceder al Sistema de Gestión de Bases de Datos Neo4j desde Python.
Gracias a él se puede generar un fichero de acceso al modelo de datos, mapeando todos los campos y recuperando y actualizando los datos del mismo como si de objetos se tratase.
Además posibilita (opción que se emplea asiduamente para la recuperación de grupos de datos complejos) la creación de consultas en Cypher [[¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.](#)], el lenguaje de consultas propietario de Neo4j.
- Python-dateutil (2.4.4) y Pytz (2014.7)
Paquetes necesarios para las conversiones de fechas entre Neo4j y Django.
- Xlwt (1.0.0) [[¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.](#)]
Paquete utilizado para la exportación de los diferentes conjuntos de datos de la aplicación (Datos del grupo, datos de los índices, etc...) a hojas de Excel. Permite la creación de ficheros .xls, y la definición de libros y hojas dentro de los mismos.

HTML, CSS y JavaScript

Dado que la aplicación se ha desarrollado para un entorno WEB, es obvia la necesidad de utilizar los lenguajes y herramientas más comunes para este fin, como son HTML, CSS y JavaScript.

No veo necesario ahondar en la definición de los tres debido a su sobrada popularidad, sin embargo sí es recomendable dar ciertas pinceladas sobre dos elementos que, basándose en los anteriores lenguajes, facilitan de una u otra forma el diseño y la elaboración de las gráficas asociadas a los índices y que sí se han empleado en la aplicación.

Por un lado se ha utilizado Bootstrap [[BOOTSTRAP.](#) [Módulo software]], framework que aúna HTML, CSS y JavaScript para ofrecer una serie de instrucciones y métodos que permiten realizar un diseño limpio de las plantillas de la aplicación. Se ha empleado, sobre todo, para la generación de los menús, y el diseño con paneles. De esta forma, mediante la aplicación de estilos a los diferentes elementos de las plantillas, el diseño conseguido es mucho más limpio, eficiente y rápido.

Por otro lado se ha utilizado D3.js [[D3](#)] para la generación de los gráficos de los índices. Es una librería JavaScript empleada para manipular documentos basados en datos, permitiendo la generación de gráficos muy potentes y detallados, así como la manipulación interactiva de los mismos por parte del usuario a través de cualquier navegador WEB.

Graph API

Graph API [¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.] es, sin lugar a dudas, la tecnología que más tiempo de estudio y mayor curva de aprendizaje requiere de entre todas las empleadas en este proyecto.

Nos encontramos frente a la API de Facebook: una extensa guía de aprendizaje, guía de referencia y documentación sobre las peculiaridades necesarias para poder interactuar y extraer información de cualquier elemento de Facebook, ya sea un perfil personal, una página o un grupo.

Como dato relevante, antes de introducir los aspectos más importantes de esta API, cabe decir que está en continua evolución y que, por tanto, no está para nada exenta de errores e imprecisiones que se han sufrido durante el diseño (prueba de ella ha sido la incapacidad de acceder a los “me gusta” de los comentarios, dato posible según la documentación, pero que una vez puesto en práctica no funcionaba)

Esta es una API basada en HTTP, usada para consultar, postear, subir fotos y otras muchas funciones. Esta API permite trabajar con una serie de elementos que representan toda la información de Facebook, y que tienen un ID único que los representa y puede ser utilizado para su consulta directa vía peticiones HTTP:

- **Nodes:** usuarios, fotos, páginas, comentarios...
- **Edges:** conexiones entre dos vértices o nodos, como las fotos de una página, o los comentarios de una foto.
- **Fields:** información de los nodos, como el cumpleaños de un usuario o el nombre de una página.

La mayoría de las peticiones a la API requieren de una serie de “tokens” que proporcionan acceso seguro (de forma temporal) a la API. Este token es una cadena opaca (que incluye información sobre cuando expira y que app lo ha generado) que identifica al usuario, app o página, y que es usada para realizar llamadas a la API,

Hay varios tipos de tokens: de usuario, de página, de aplicación y de cliente. En nuestro caso sólo nos interesan los tokens de cliente por lo que remitimos a las Referencias [¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.] del presente documento en caso de querer recabar más información sobre los otros tipos.

Los tokens de cliente son identificadores que se pueden embeber dentro de las aplicaciones, sólo es necesario pasar el identificador de la aplicación (app id) y el secreto (app secret) dentro del parámetro “acces_token” cuando se realiza una llamada. En nuestro caso estos datos se configuran dentro del fichero settings.py de la aplicación, con las etiquetas FACEBOOK_APP_ID y

FACEBOOK_APP_SECRET. En el apartado 5.3 Instalación, se explicarán los pasos necesarios para obtener estos datos.

Además de la utilización de tokens es importante conocer la metodología básica de llamadas que permite la API de Facebook. Como se ha mencionado anteriormente existen una serie de nodos que pueden ser consultados construyendo una llamada HTTP a partir de su identificador. De esta forma, por ejemplo, para acceder a la información de un grupo concreto se utilizaría una construcción similar a esta:

```
GET graph.facebook.com  
  
/{group_id}
```

Las llamadas HTTP son totalmente configurables, permitiendo selección de campos, filtros, paginación para las consultas que devuelven muchos registros o agrupaciones de objetos para consultas agrupadas.

En el caso del presente proyecto todas estas llamadas se realizan a través del paquete Django-facebook, ya analizado anteriormente.

En cualquier caso, existe una herramienta muy útil para trabajar con Tokens y probar las diferentes acciones que se pueden realizar mediante la API: el Graph API Explorer (Fig. 33) [[Graph API Explorer](#)].

Visual Studio .NET

Como contraposición directa a la ideología de código abierto, y con el valor añadido de querer probar las extensiones de Python para el IDE de Microsoft, se ha optado por desarrollar el proyecto haciendo uso de Visual Studio .NET 2012.

Lo relevante de esta elección no es tanto la plataforma de trabajo, ampliamente conocida, sino la utilización de Python Tools For Visual Studio [[MICROSOFT. Python](#)], conjunto de herramientas que permiten emplear el entorno de Microsoft como si de un IDE de Python se tratase.

En general las herramientas se comportan adecuadamente, aunque su integración con Visual Studio (y más por la versión concreta que se ha utilizado) puede llegar a generar determinadas incompatibilidades difícilmente subsanables. En mi caso, por ejemplo, he utilizado dos máquinas diferentes para desarrollar el proyecto, en una de ellas la gestión de errores de Visual Studio funcionaba perfectamente, mientras que en la otra era imposible emplearla debido a la ingente cantidad de excepciones que daban en módulos internos de Python nada más lanzar la

ejecución de la misma. Teniendo en cuenta que la configuración de ambas máquinas era casi idéntica (exceptuando que una corre sobre Windows 7 de 32 bit y la otra de 64), no llegué a localizar el problema y poder subsanarlo.

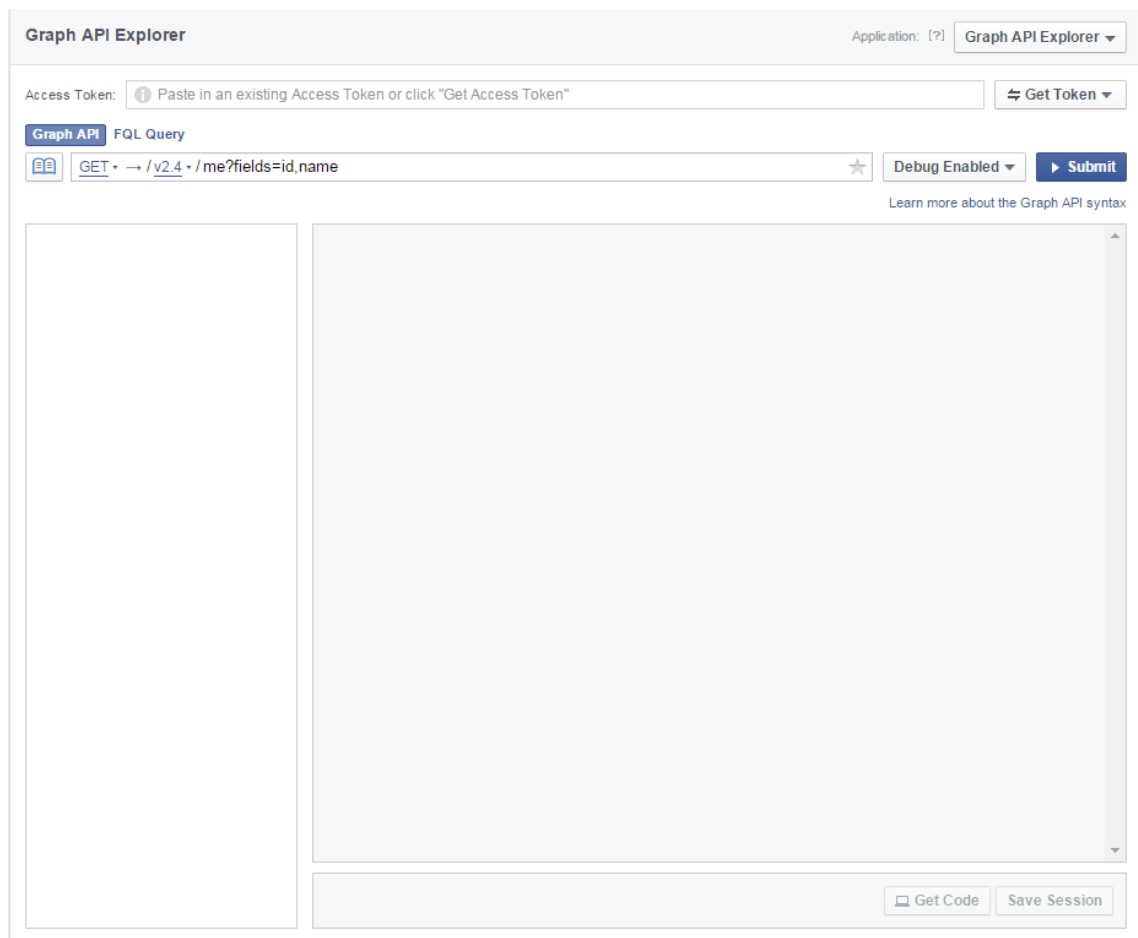


Fig. 33 – Graph API Explorer

Quedaría pendiente emplear las nuevas versiones de las herramientas sobre la versión más nueva de Visual Studio como posible mejora secundaria del proyecto (que no de la aplicación)

Sistemas de Gestión de Base de Datos (SGBD)

En la aplicación se han empleado dos sistemas de gestión de base de datos diferentes. Esta elección responde a la estructura de la propia información, a la perdurabilidad y dinamismo de la misma, y a la idea de emplear técnicas de análisis de redes y grafos como posible mejora del proyecto.

Por un lado se ha utilizado Neo4j para almacenar toda la información obtenida de Facebook, tanto de los grupos como de los propios usuarios de Facebook que son miembros o han participado de alguna forma dentro del grupo sin llegar a ser miembros.

Neo4j [¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.] es un Sistema de Gestión de base de Datos orientado a grafos, es decir, que almacena los datos estructurados en grafos, en lugar de emplear tablas como sí hacen los sistemas convencionales. Esto, además de permitir trabajar intuitivamente con conceptos de redes y teoría de grafos, tiene como ventajas una mayor escalabilidad, un mejor manejo de grandes cantidades de datos, una mayor rapidez, y un entorno web totalmente gratuito y libre para la gestión de las bases de datos.

Como pega, por poner alguna, requiere de un tiempo de aprendizaje tanto del sistema en sí, como de su lenguaje de consultas Cypher [¡ERROR! NO SE ENCUENTRA EL ORIGEN DE LA REFERENCIA.], el cual si bien se asemeja en determinados aspectos al normalizado y extendido SQL, tiene su propia idiosincrasia, sintaxis y metodología de uso que es necesario dominar para realizar las consultas complejas empleadas en la obtención de los índices (Fig. 34).

```
MATCH (u:User)-[:CREATES]-(c:Comment)-[:COMMENTS]->(p:Post)-[:PERTAINS]->(g:Group)-[:IS_MEMBER]-(u)
WHERE g.id = 799773383434385
RETURN "Comentario" AS Tipo, c.id AS ID, u.id AS Usuario, c.message AS Texto,
       im.is_administrator AS Profesor, c.created_time AS FechaCreacion, c.like_count AS Likes
UNION
MATCH (u:User)-[r1]->(p:Post)-[:PERTAINS]->(g:Group)-[:IS_MEMBER]-(u)
WHERE g.id = 799773383434385
RETURN "Post" AS Tipo, p.id AS ID, u.id AS Usuario, p.message AS Texto, im.is_administrator AS Profesor, CASE
WHEN (u)-[:CREATES]->(p)
THEN p.created_time
WHEN (u)-[:SHARES]->(p)
THEN p.created_time
ELSE r1.snap_created_time
END AS FechaCreacion, 0 AS Likes
ORDER BY FechaCreacion
```

Fig. 34 – Ejemplo de consulta en Cypher

El segundo SGBD empleado ha sido SQLite 3. Este es un sistema SQL tradicional, con la peculiaridad de ser libre, autocontenido y muy ligero, de forma que puede embeberse en la propia aplicación sin requerir de procesos de instalación, configuración ni mantenimiento posteriores.

La compatibilidad de este motor con Django es total, hasta el punto de permitir la creación de la estructura de tablas haciendo uso de la capa de Modelo del propio sistema de Django. Simplemente se requiere definir las tablas y sus relaciones en el fichero models.py, y es el sistema el encargado de crearlas, y de mapearlas para permitirte usarlas como si de objetos se tratase, haciendo innecesario el uso de SQL para la consulta, inserción o actualización de los datos contenidos.

Este SGBD se ha empleado tanto por el paquete Django-Facebook, como por la propia aplicación. Django-Facebook lo utiliza para almacenar la información de las sesiones, tokens de acceso a Facebook, y demás datos de configuración que considera relevantes. Este uso, no obstante, es totalmente transparente a la propia aplicación, no requiriendo ningún tipo de tarea de consulta o mantenimiento por parte del presente proyecto.

Aprovechando que ya estaba en uso, se consideró oportuno emplear también dicho sistema para la creación de tablas que almacenaran contenido de carácter más estático y funcional y que, por tanto, no tiene sentido almacenar en una base de datos orientada a grafos. Inicialmente se ha incluido la estructura necesaria para almacenar los días lectivos por mes de cada curso lectivo, dato requerido para calcular uno de los índices de la aplicación. Aunque perfectamente podría ser utilizada para futuras ampliaciones como almacenar información de usuarios maestros (no usuarios de Facebook) que puedan acceder a todos los grupos y no solo a los propios, o estadísticas de utilización, acceso, etc...

La forma de trabajar e interactuar fuera de la aplicación con ambos sistemas de gestión de base de datos es muy diferente. Para Neo4j se facilita un acceso web al puerto 7474 del servidor dónde se instaló, siendo el propio explorador web la herramienta de trabajo empleada para realizar consultas o para revisar y modificar la propia configuración del mismo (Fig. 35).

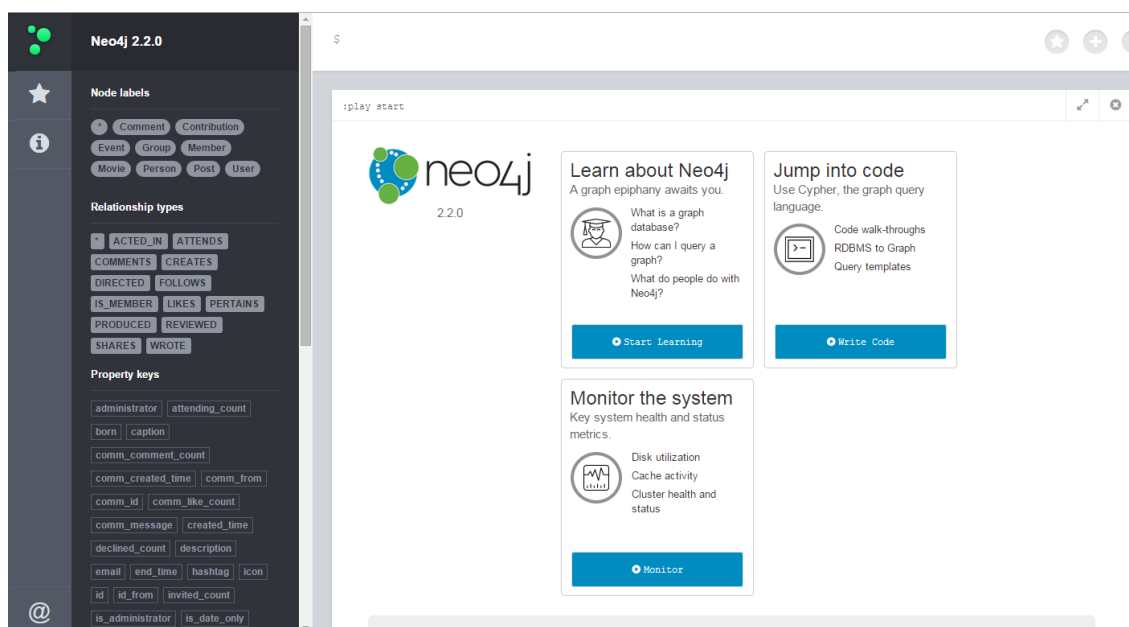


Fig. 35 – Neo4j Browser

Mientras que para gestionar la base de datos SQLite 3 existen herramientas de terceros, como SQLite Manager (Fig. 36), una extensión para el navegador Mozilla Firefox, que te permite trabajar con las bases de datos de este SGBD.

5.3 Instalación

El proceso de instalación de la aplicación requiere de un proceso previo de alta y configuración en la plataforma de desarrolladores de Facebook (Facebook Developers).

Facebook exige que cualquier aplicación que se vaya a desarrollar para su plataforma sea definida y configurada dentro de un perfil de desarrollador. Para ello cualquier cuenta de

Facebook puede registrarse como desarrollador desde su portal

<https://developers.facebook.com/>

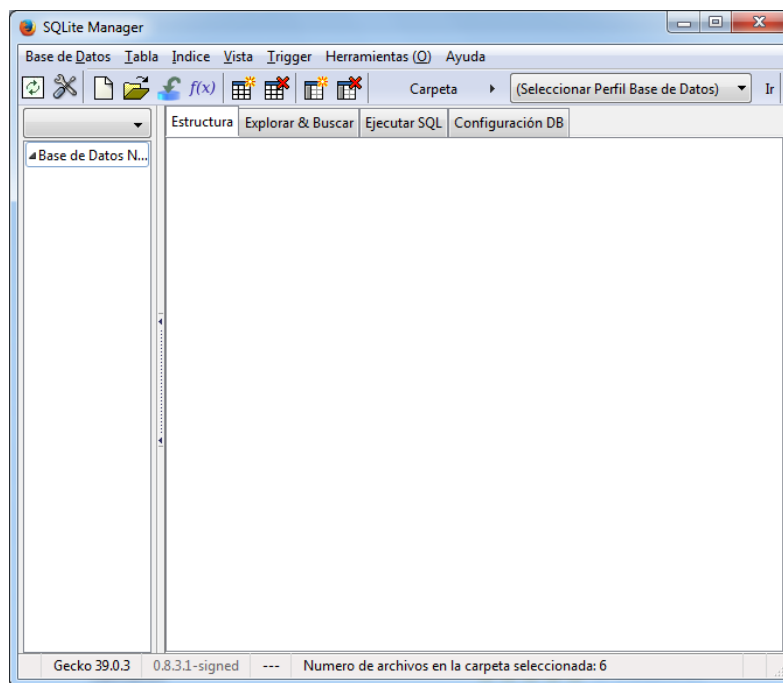


Fig. 36 – SQLite Manager

Dentro de este sitio web específico para desarrolladores existe un apartado para crear y gestionar todas las aplicaciones creadas. La primera vez que se accede, en caso de no estar registrado como desarrollador, dará la opción de llevar a cabo este registro (Fig. 37), opción que pasa por la aceptación de su política de privacidad, y por la obligatoriedad de enlazar un número de teléfono válido al que se te enviará un mensaje de texto con un código de activación (Fig. 38)

Una vez se ha dado de alta el perfil de desarrollador ya se puede crear una aplicación del tipo deseado (IOs, Android, Facebook Canvas, Sitio Web), asignándole un nombre, un espacio de nombres y una categoría como datos básicos antes de poder acceder al tablero de trabajo o Dashboard de la aplicación, desde donde se podrá configurar muchos otros datos relevantes de la misma. (Fig. 39)

Dentro del tablero de trabajo, además de una aceptable cantidad de herramientas de testeo, estadísticas de uso y acceso de la aplicación, se pueden configurar elementos imprescindibles para que las consultas a la API de Facebook no den error por falta de permisos.

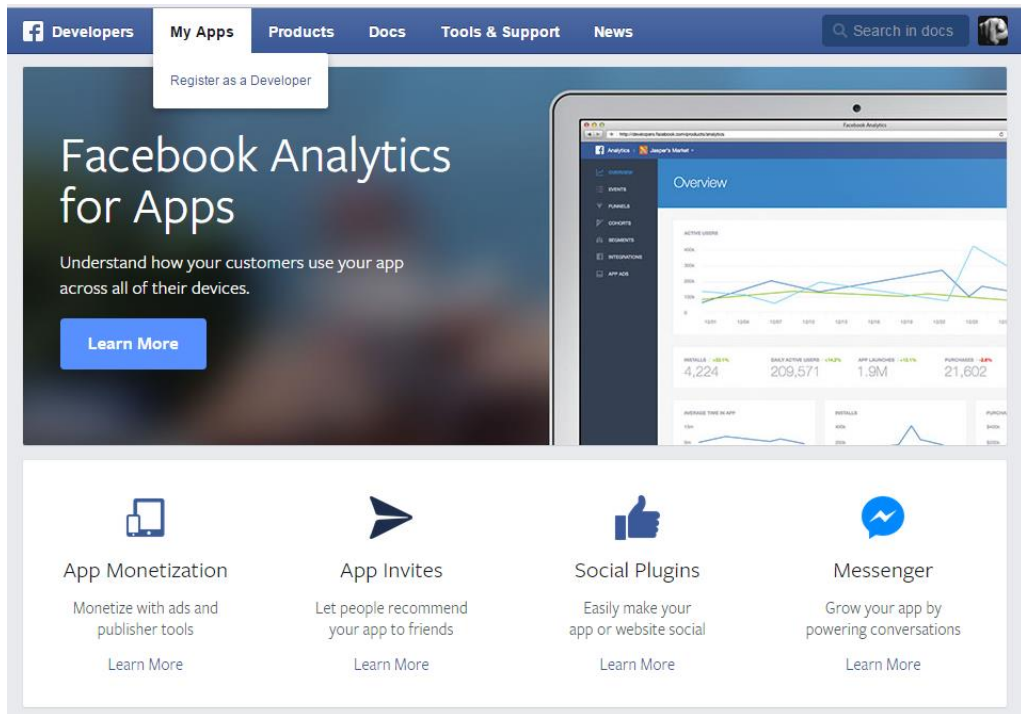


Fig. 37 – Facebook Developers

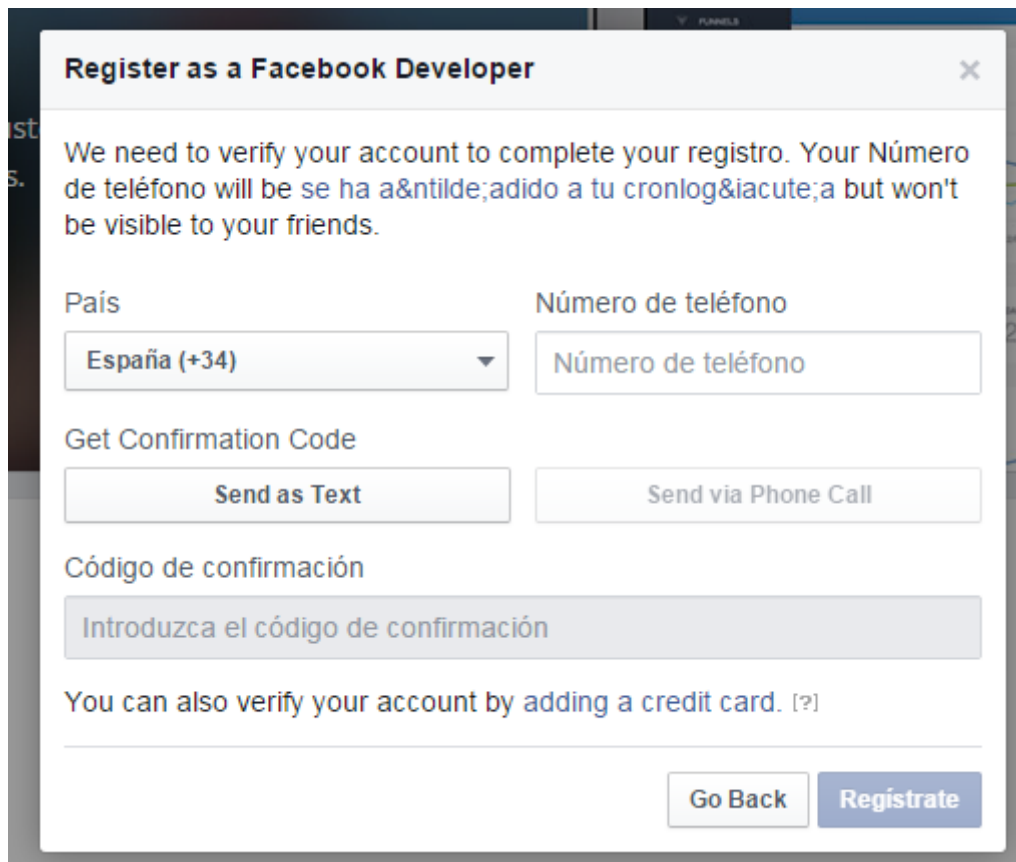


Fig. 38 – Alta como desarrollador de Facebook

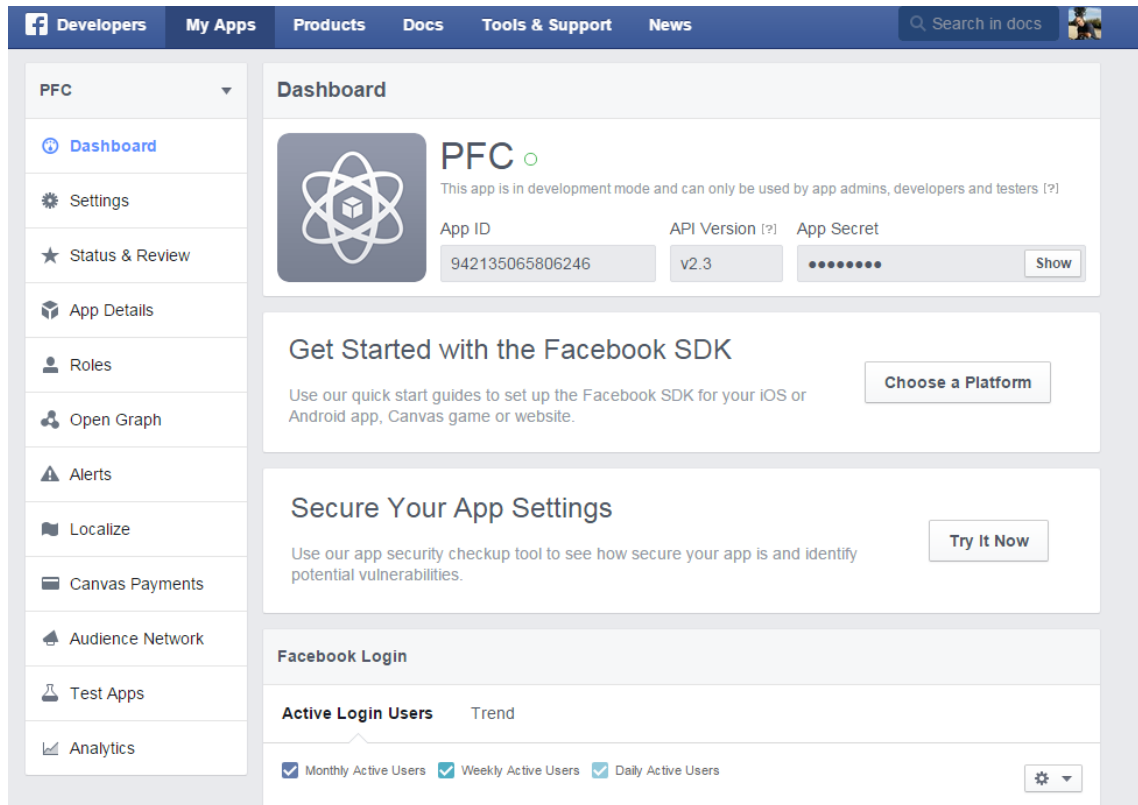


Fig. 39 – Dashboard de una aplicación de Facebook

De especial relevancia es la pestaña “Settings” (Fig. 40) en la cual se definen los siguientes datos:

- El nombre de la aplicación.
- El dominio de la aplicación, que deberá coincidir con el dominio asignado al sitio web final pero que, durante la fase de desarrollo es suficiente con apuntarlo a localhost.
- El WebSite, donde se dice cuál será la URL de acceso a la aplicación.

The screenshot shows the 'Basic' tab of the Facebook Developer Console. It contains the following fields and controls:

- App ID:** 942135065806246
- App Secret:** Masked with dots, with a 'Show' button.
- Display Name:** PFC
- Namespace:** pfcfacebook
- App Domains:** localhost
- Contact Email:** Used for important communication about your app
- Website:** Section with a 'Quick Start' button and a 'Site URL' field containing http://localhost:8080/.
- + Add Platform:** Button to add more platforms.
- Buttons:** 'Delete App' (red), 'Discard', and 'Save Changes' (blue).

Fig. 40 – Basic Settings de una aplicación de Facebook

Dentro de esta misma pestaña de configuración de la aplicación se debe configurar la política de autorización de clientes “OAuth” para permitir la utilización del token de cliente desde la aplicación (Fig. 41)

The screenshot shows the 'Client OAuth Settings' section of the Facebook Developer Console. It includes the following settings:

- Client OAuth Login:** Enabled (YES).
- Web OAuth Login:** Enabled (YES).
- Embedded Browser OAuth Login:** Enabled (YES).
- Force Web OAuth Reauthentication:** Disabled (NO).
- Valid OAuth redirect URIs:** http://localhost:8080/
- Login from Devices:** Disabled (NO).

Fig. 41 – Configuración de autenticación de clientes para una aplicación de Facebook

Una vez realizado estas modificaciones en el perfil de la aplicación, y añadiendo el App ID y el App Secret que aparecen en la información básica de la aplicación al fichero de configuración settings.py del proyecto de Django con las etiquetas FACEBOOK_APP_ID y FACEBOOK_APP_SECRET, la aplicación estará preparada para validarse contra Facebook, generar un token de acceso, y realizar consultas sobre el grupo y sus aristas (Feeds, Posts, Events, Comments, etc...). Obviamente es importante mantener ambos datos en secreto con el fin de evitar suplantaciones de identidad a la hora de utilizar la aplicación y los tokens de acceso que pudiera originar.

Un dato relevante es que mientras la aplicación esté catalogada como aplicación en desarrollo, sólo los usuarios añadidos con Rol de Administrador, Desarrollador o Tester, podrán ejecutarla y probarla. Estos roles se definen en la pestaña “Roles” del Dashboard (Fig. 42).

Existe un paso final previo a la puesta en productivo de la aplicación (paso que no se ha seguido en el presente proyecto), que consiste en que, dependiendo de las acciones que lleve a cabo la aplicación y los permisos que requiera para llevarlas a cabo, Facebook debe recibir y validar la aplicación para verificar que es válida y confiable (Fig. 43).

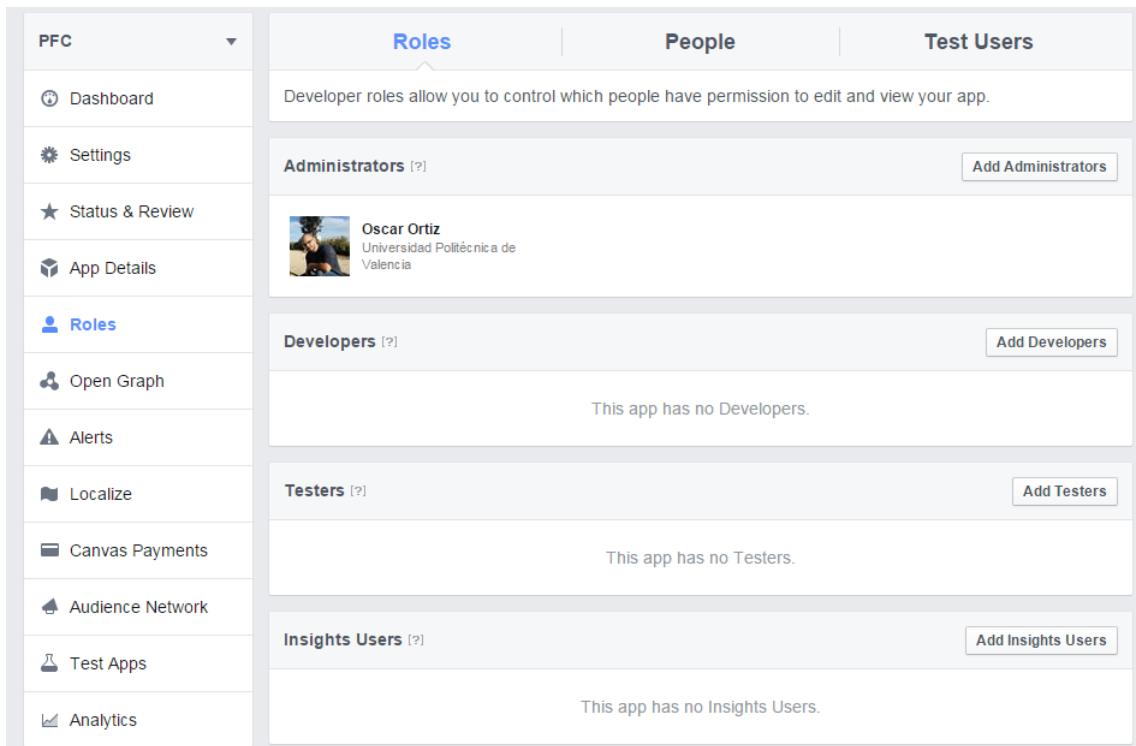


Fig. 42 – Roles de usuario para una aplicación de Facebook

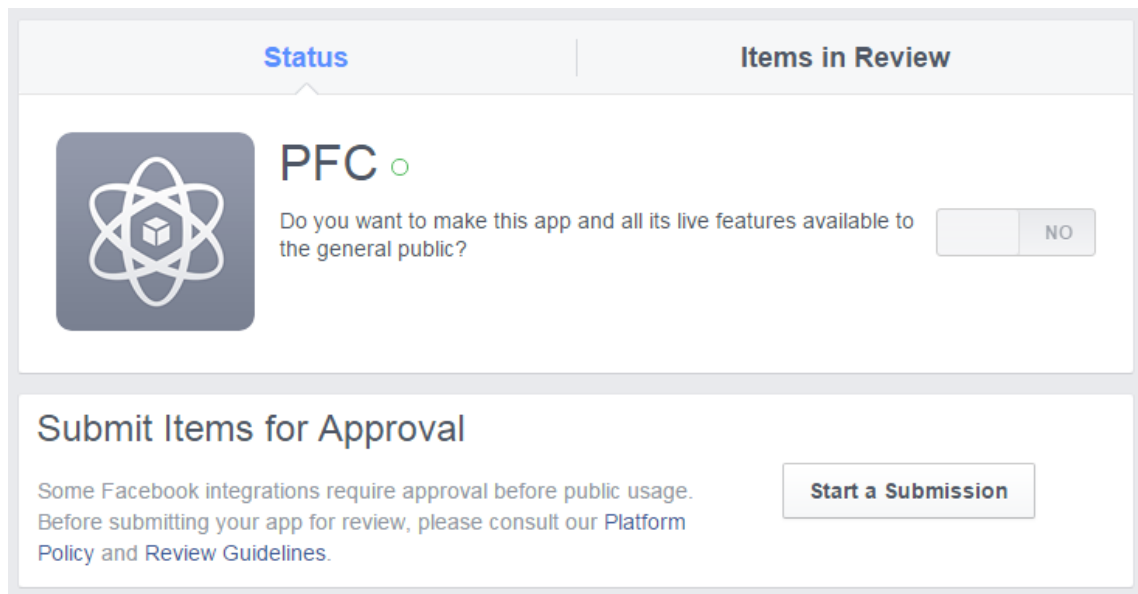


Fig. 43 – Aprobación de la aplicación por parte de Facebook

Otro punto a tener en cuenta referente a las peculiaridades de Facebook es la gestión de permisos, tan sumamente compleja que, si bien ayuda a mantener seguros los datos de Facebook, no facilita en nada la tarea de desarrollar una aplicación. Para cada nodo (Usuario, Grupo) requiere un tipo de permiso específico, y no sólo eso, sino que para determinadas acciones requiere de permisos adicionales (Fig. 57, Fig. 58).

Superada la fase de creación y configuración de la aplicación de Facebook el resto de procesos son relativamente sencillos y guiados. Desde la propia instalación de Python en Windows, a través de un paquete autoinstalable (.msi) que te guía en la instalación, pasando por la instalación del entorno de trabajo Visual Studio .NET, no tienen mayor complicación y no merecen un capítulo propio dentro de esta sección.

En cuanto a la instalación de los paquetes específicos de Python empleados en el desarrollo de la aplicación, todos ellos se instalaron dentro del propio Visual Studio, desde la opción Install Python Package (Fig. 46).

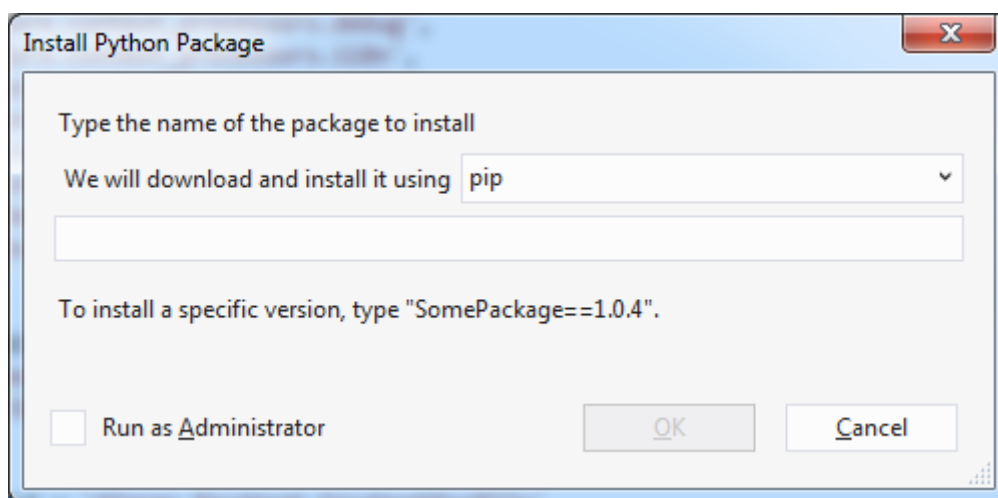


Fig. 46 – Instalador de paquetes de Python de Visual Studio .NET

Incluso el SGBD Neo4j tiene un instalador que hace del proceso algo muy sencillo, permitiendo la posterior modificación de los parámetros de configuración del servidor a través de ficheros de configuración de texto plano que, inicialmente, no es necesario tocar (Fig. 47).

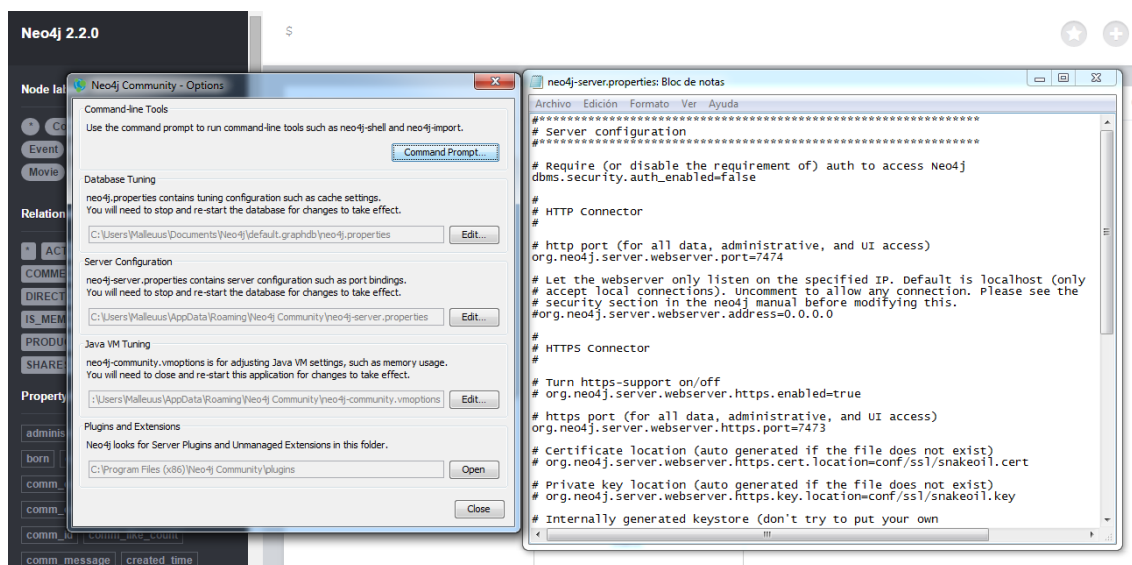


Fig. 47 – Configuración de Neo4j

5.4 Pruebas

Presentadas las herramientas y tecnologías empleadas para la implementación de la aplicación vamos a pasar a realizar una prueba guiada, detallada y bastante visual de lo que es la aplicación y las opciones que ofrece.

Puesto que la parte relevante del proyecto es la conexión con Facebook, la obtención de datos y el volcado y manipulación de los mismos en el SGBD elegido para tal fin, no se ha incidido demasiado en el aspecto visual de la aplicación, siendo un tanto aséptica y minimalista tanto en la elección de los colores como en el diseño funcional de la misma. Se plantea como posible ampliación futura la posibilidad de rediseñar la interfaz y hacerla más amigable y vistosa.

Para las pruebas se ha optado por utilizar como explorador web Chrome, por entender que es el más fiable al respetar los estándares WEB y que, por tanto, dará menos problemas de visualización y compatibilidad con los diferentes elementos utilizados (Bootstrap y D3.js)

Lo primero que nos encontramos al acceder a la URL de la aplicación variará en función de si tenemos una sesión abierta o no. Suponiendo una entrada limpia (primera vez que se entra), nos encontramos ante un página (Fig. 48) cuya única opción pasa por el típico botón de inicio de sesión de Facebook. Este botón forma parte de la lógica del módulo Django-Facebook, siendo toda su operativa totalmente transparente para el desarrollo.

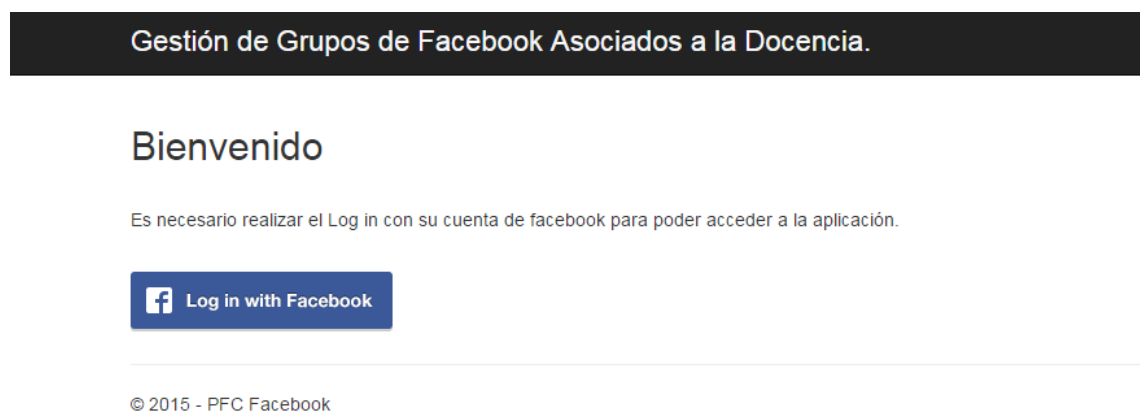


Fig. 48 – Inicio de sesión de la aplicación

Básicamente se trata de un proceso normal de iniciar sesión en Facebook, con la correspondiente aceptación por parte del usuario de los permisos que requiere la aplicación para acceder a su perfil, y a la información de los grupos de los que es miembro (bien como usuario, bien como administrador).

El proceso de inicio de sesión es dirigido, de forma que si ya se tiene abierta una sesión normal de Facebook en el explorador utilizará este usuario para el inicio de sesión en Facebook, mostrando la petición de usuario y contraseña en caso contrario. No obstante, este proceso es gestionado directamente por Facebook, de forma que los errores generados durante el proceso son errores lanzados por la propia plataforma, y no por la aplicación.

Una vez se ha iniciado la sesión, se ha generado el token de acceso a Facebook y se ha almacenado en la sesión hasta que esta se cierre de forma explícita (cerrar el navegador no cierra la sesión con la aplicación y, por tanto, guarda el usuario y el token de acceso independientemente de si el usuario de Facebook ha cerrado la sesión en Facebook o no), se entra en la página de selección de grupo de trabajo (Fig. 49).

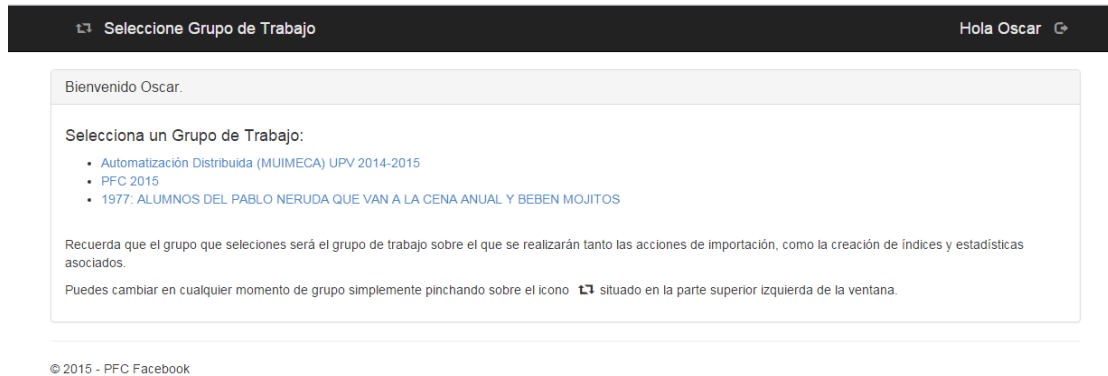


Fig. 49 – Selección de grupo de trabajo

El grupo de Trabajo es el grupo de referencia del cual se va a mostrar toda la información pertinente y sobre el que se van a realizar todas las acciones que el usuario considere oportunas.

El grupo también se guarda en la sesión, de forma que siempre que exista se accederá de manera inmediata a la ficha de grupo en lugar de a la página de selección de grupo, permitiéndose en todo momento un cambio de grupo utilizando el icono situado en la parte superior izquierda de la cabecera de la página, icono que redirige a la propia página de selección de grupo de trabajo.

Los grupos mostrados en esta página se obtienen realizando la primera consulta directa a la API de Facebook y son, por tanto, grupos de los que el usuario de la aplicación (también usuario Facebook) es miembro. Eso no quiere decir que el grupo exista como tal en la base de datos, sin embargo, en caso de seleccionarse como grupo de trabajo se insertará como nuevo grupo en la base de datos, aunque sólo con la información relativa al propio grupo, en ningún caso con entradas, eventos, comentarios y demás información que sólo se importa o actualiza mediante la opción de menú “Importar a Base de Datos”

Como norma, independientemente de en qué sección de la aplicación nos encontremos, los datos de la cabecera serán siempre los mismos (de izquierda a derecha):

- Botón para cambio de grupo de trabajo que lleva a la página de selección de grupo.

- Nombre del grupo seleccionado que sirve como acceso directo a la propia ficha del grupo, o un texto “Seleccione Grupo de Trabajo” en caso no existir un grupo en la sesión actual.
- Nombre del usuario de Facebook para identificar quién ha iniciado sesión en la aplicación.
- Botón de cierre de sesión, que borra los datos de la misma y el token de acceso a Facebook.

Una vez seleccionado un grupo determinado, se modifica el texto de la barra superior con el nombre del grupo de trabajo activo, y se abre la página de la ficha del grupo (Fig. 50).

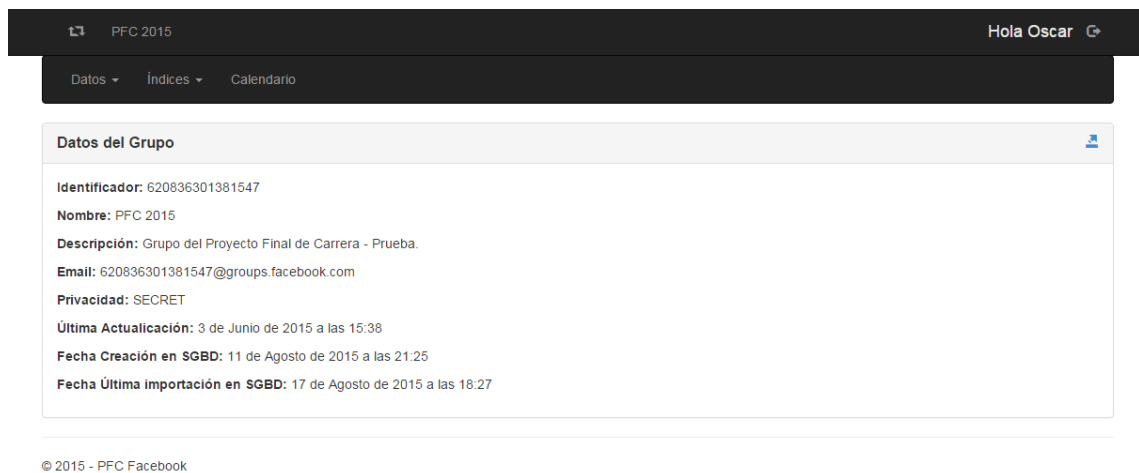


Fig. 50 – Ficha del grupo de trabajo

Este es el núcleo propiamente dicho de la aplicación, con un submenú específico de acciones a realizar sobre el grupo de trabajo seleccionado, y una plantilla dónde se mostrarán todos los resultados derivados de dichas acciones.

Por defecto se entra a la ficha de datos del grupo, dónde se muestran algunos de los datos básicos del grupo obtenidos de Facebook y almacenados en la base de datos de Neo4j, tales como el identificador de Facebook, el nombre, la descripción, el email, la privacidad del grupo (abierto, privado o secreto), y varias fechas que permiten ver el estado de importación de los datos del grupo. La primera de las fechas (Última Actualización) es la fecha de la última vez que se actualizaron los datos del grupo en Facebook. Las siguientes fechas (Fecha Creación en SGBD, y Fecha Última Importación en SGBD) muestran la fecha de cuando se creó por primera vez en la base de datos el grupo, y de cuando se importó por última vez los datos de Facebook a la base de datos de Neo4j. De esta forma se tiene una referencia visual e instantánea de cuándo fue la última vez que se importaron los datos y cuál es, por tanto, el grado de desactualización y la confiabilidad de los mismos a la hora de valorar los resultados mostrados por los índices.

Como dato relevante cabe fijarse en el icono de la parte superior derecha del encabezado de los datos devueltos por cualquier acción. 

Con este icono se posibilita la exportación a Excel de todos los datos referentes a la petición por parte del usuario, no sólo a los mostrados por pantalla, sino también a los que se crean conveniente en cada caso para poder gestionar la información de forma correcta (En el caso de los índices se exporta tanto los índices, como los datos a partir de los que se calculan estos)

La ficha del grupo activo dispone de tres menús básicos: Datos, Índices y Calendario

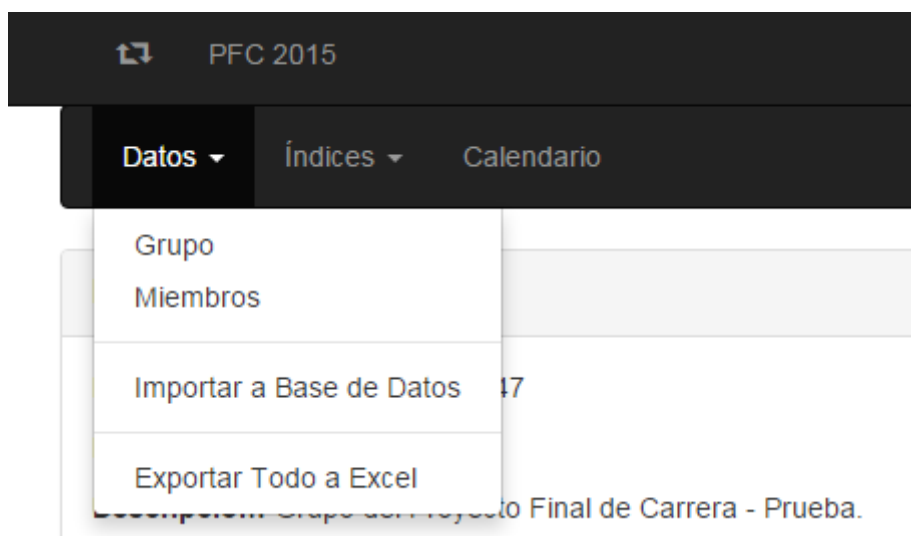


Fig. 51 – Menú de datos del grupo de trabajo



Fig. 52 – Menú de índices del grupo de trabajo

El menú de Datos (Fig. 51) Permite mostrar la información del grupo, mostrar los miembros que pertenecen al grupo (En la base de datos, no en Facebook), lanzar el proceso de importación de Facebook a la base de datos de Neo4j, o lanzar el proceso de exportación de todos los datos del grupo de trabajo activo que se encuentran en la base de datos de Neo4j.

La información de los datos del grupo ya la hemos analizado puesto que es la opción por defecto a la cual se entra una vez seleccionado el grupo de trabajo, o si ya existe un grupo de trabajo activo y guardado en la sesión.

La ficha con los miembros del grupo (Fig. 53) muestra un listado con la cantidad total de usuarios que pertenecen o han intervenido en el grupo. Es importante esta distinción puesto que los grupos públicos, o con eventos públicos, pueden llegar a invitar a interactuar a usuarios de Facebook que no pertenecen al propio grupo, siendo necesaria su inclusión en la base de datos para relacionar las acciones generadas con los diferentes elementos (crear comentario, crear entrada, etc...)

En esta ficha se muestra el nombre del miembro, su identificador de Facebook, y se resalta la condición de administrador del mismo, que normalmente coincidirá con la condición de profesor que impartirá la asignatura para la que se ha constituido el grupo de Facebook.

Si bien la información de este apartado es poco útil, la idea es ampliar la funcionalidad del proyecto añadiendo una ficha de usuario de grupo, a la que se podría acceder desde esta misma página, y en la que se mostrarían estadística personalizadas para el propio usuario como número de entradas creadas, número de comentarios, índices específicos de participación individual frente al cómputo global, etc...

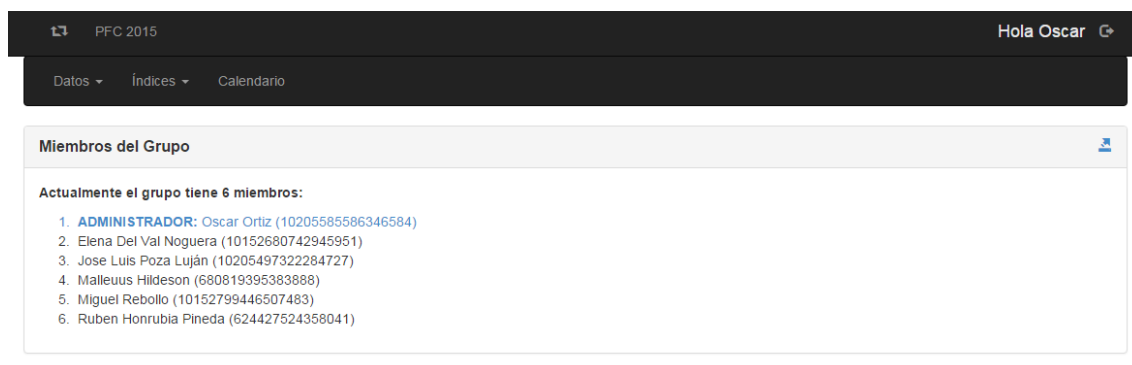


Fig. 53 – Ficha de los miembros del grupo de trabajo

Como se ha comentado, el proceso de exportación de los datos de miembros (Fig. 13) incluye todos los datos de la base de datos, no sólo los mostrados por pantalla, incluyendo por ejemplo, las fecha de primera y última importación a la base de datos de Neo4j.

La opción de importar a Base de Datos es la que encierra la mayor complejidad de la aplicación y, por tanto, es la que más se demora en el tiempo para su ejecución, requiriendo de varios minutos de espera en función de lo complejo y de la cantidad de información que incluya el grupo de Facebook.

Es por ello que como paso previo se muestra una página de información (Fig. 54) explicando la tarea que se va a realizar y pidiendo un poco de paciencia.



Fig. 54 – Importación del grupo de trabajo

Al aceptar se procede a realizar todas las acciones asociadas a la importación de los datos, tanto de inserción como de actualización de los mismos en caso de haber cambiado. Para ello se ejecutan muchas consultas a la API de Facebook (llamadas HTTP), lo cual demora el proceso y lo convierte en la acción más pesada a nivel temporal de la aplicación.

Una vez finalizado el proceso de importación se pasa automáticamente a la ficha de datos del grupo, donde se podrá comprobar que la fecha de última actualización habrá cambiado, y se podrá leer un mensaje de confirmación de proceso finalizado (Fig. 55).

Este recuadro rojo es la forma que se utiliza para mostrar tanto los errores, como cualquier otra información pertinente, y siempre se situará entre el menú y el contenedor de datos que devuelva la acción de menú seleccionada por el usuario.



Fig. 55 – Mensajes de error e información

La última opción del menú de Datos, Exportar Todo a Excel, genera un fichero Excel (Fig. 16) con tantas hojas como entidades se han creado en la base de datos de Neo4j:

- **Grupos:** la misma información que genera la exportación a Excel de los datos del grupo. ID, Nombre, Descripción, Email, ID Propietario, Privacidad, Fecha Última Actualización, Fecha Primera Importación y Fecha Ultima Importación.
- **Miembros:** la misma información que genera la exportación a Excel de los datos de los miembros. ID, Nombre, Profesor, Fecha Primera Importación, Fecha Última Importación.
- **Posts:** información de todas las entradas del grupo. ID, Tipo, Titulo, Subtitulo, Descripción, Mensaje, Historia, ID Origen, Para, Fecha Creación, Fecha Última Actualización, Link, ID Objeto, Veces Compartido, Origen, Status, Fecha Primera Importación, Fecha Ultima Importación.
- **Eventos:** información de todos los eventos del grupo. ID, Titulo, Descripción, ID Propietario, Fecha Última Actualización, Fecha Primera Importación, Fecha Última Importación, Fecha Inicio, Fecha Fin, Quizás Asistan, Declinados, Sin Respuesta, Asistirán, Invitados.
- **Comentarios:** información de todos los comentarios a entradas y eventos del grupo. ID, ID Origen, Mensaje, “Me gustas”, Fecha Creación, Fecha Primera Importación, Fecha Ultima Importación.
- **Compartidos:** información de todas las entradas compartidas en el grupo, es decir, de entradas externas que se han compartido dentro del grupo. ID, Usuario, Nombre, Usuario, Fecha Primera Importación, ID Post, Titulo Post.
- **Likes:** información de los “me gusta” sobre entradas del grupo (La opción de “me gusta” sobre comentarios aunque está implementada no recupera datos debido a un bug en la API de Facebook). Tipo, ID Usuario, Nombre Usuario, Fecha Primera Importación, ID, Título

Pasando al menú de Índices (Fig. 52), encontramos como opciones la generación de las gráficas asociadas a los índices establecidos para el proyecto. El proceso es sencillo, se recuperan los datos de Neo4j con una consulta realizada en Cypher, se aplica la ecuación asociada al índice concreto, y se devuelve una tupla con dos vectores de datos, uno para el eje X (meses), y otro para el eje Y (valores por mes), de una gráfica creada utilizando D3.js.

Si bien la gráfica originada es de carácter estático, D3.js es tan potente que se podría modificar para realizar gráficas dinámicas con las que puede interactuar el usuario, no obstante este punto se plantea como mejora futura de la aplicación.

La primera de las opciones del menú, Mostrar Todos, genera las gráficas de todos los índices y las visualiza por pantalla (Fig. 56). Es necesario tener en cuenta que los datos de los índices no se almacenan como parte de un histórico, es decir, cada ejecución de la acción del menú recupera los datos que en ese momento existe en la base de datos y los calcula. Por tanto, estos índices pueden variar de importación en importación, puesto que como se ha explicado el proceso de importación añade registros nuevos pero también modifica los existentes (como por ejemplo el número de “me gustas” de una entrada).

Las tres opciones siguientes son opciones idénticas en cuanto a su ejecución, cambiando únicamente el índice que se va a calcular y mostrar por pantalla (Fig. 57). No obstante, para cada uno de ellos se muestra un texto descriptivo de lo que pretende calcular el propio índice, de forma que el usuario tenga una buena base para interpretar los datos obtenidos a partir del mismo. Estos textos se han extraído directamente del artículo “Propuesta de parámetros y caracterización de los grupos de las redes sociales orientados a la docencia universitaria: experiencia y resultados” incluido en la Revista RED de noviembre de 2014 [POZA Luján, Jose Luis y otros. *Propuesta*], y que ha servido de base para el cálculo y creación de los índices del proyecto.

Todas las opciones de este menú de índices permiten la exportación de datos a Excel, sin embargo en este caso, para cada índice, se crea dos hojas dentro del mismo fichero de Excel: una para los datos calculados del índice, es decir, los valores obtenidos de aplicar la fórmula para cada uno de los meses del curso lectivo, y otro con todos los datos utilizados para realizar dichos cálculos, como por ejemplo todas las entradas, comentarios o “me gusta”, separados por alumno o profesor, o por la lógica específica que cada índice requiera. (Fig. 18, Fig. 21)

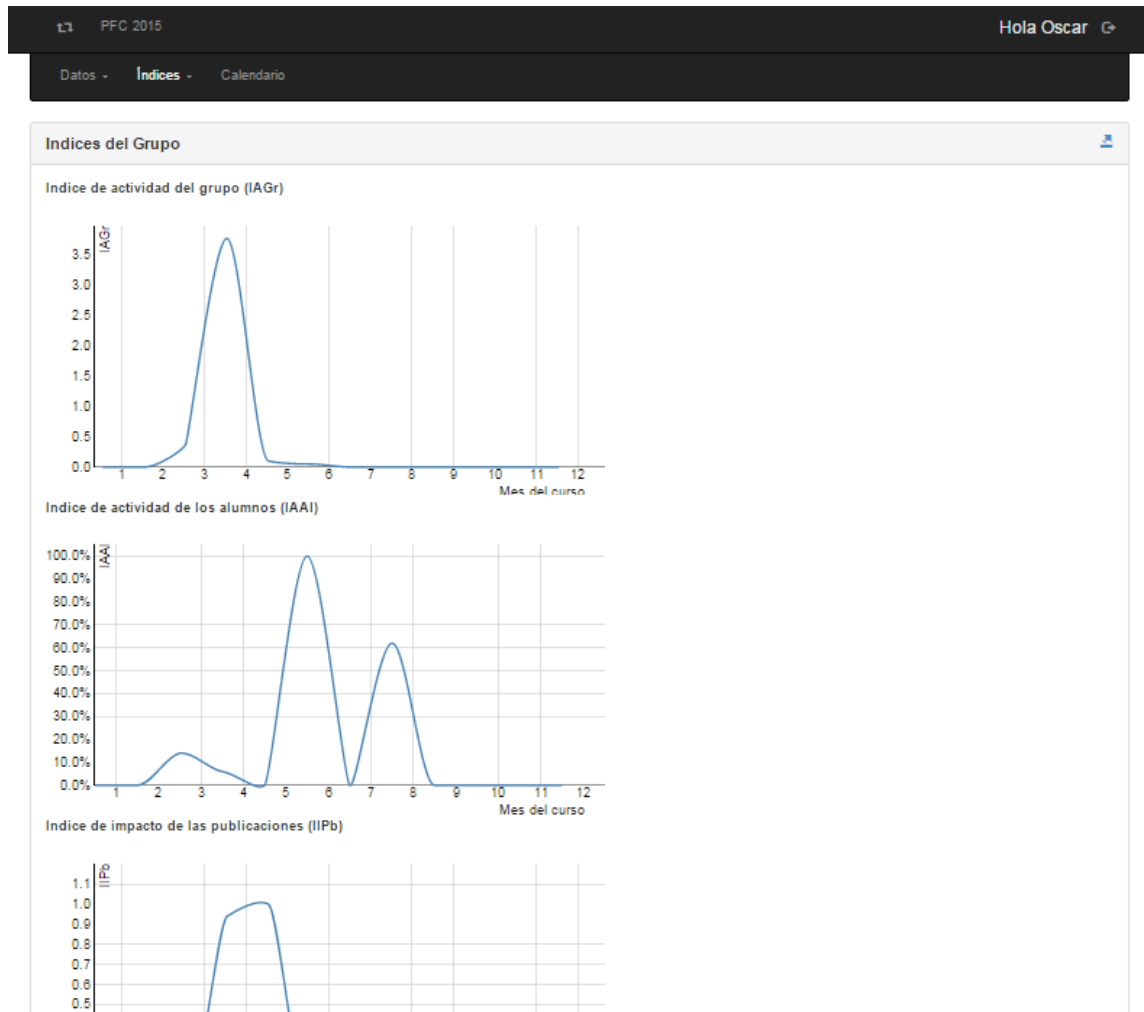


Fig. 56 – Mostrar todos los índices del grupo seleccionado



Fig. 57 – Ejemplo de índice para el grupo seleccionado.

La última acción disponible es la de Calendario. Puesto que uno de los índices propuestos requiere una división del número de entradas y comentarios entre el número de días lectivos del mes, se hacía necesaria la obtención de estos días del calendario lectivo de la universidad.

En una primera aproximación se estuvo investigando la posibilidad de obtenerlos de forma automática bien vía WebServices, o por la importación de calendarios en formato iCalc, sin embargo, dada la complejidad temporal de estas aproximaciones, y por la incapacidad de encontrar un servicio web que ofreciera estos datos, se optó por la creación de dos pequeñas tablas que contienen los días que el propio usuario puede introducir en cualquier momento.

Con el fin de facilitar la introducción se optó por permitir la modificación del calendario lectivo asociado al grupo (los años lectivos están calculados a partir de las fechas de las entradas y comentarios del mismo), con lo cual cualquier usuario que tenga un grupo en el periodo lectivo dado podrá modificar los días, siendo extensible para todos los grupos y todos los usuarios del mismo periodo lectivo. Esto es, si un profesor modifica los días lectivos del curso 2014-2015, automáticamente se utilizarán para el cálculo de los índices de cualquier otro grupo dentro del mismo periodo lectivo.

Si bien esto puede inducir a cierta confusión puesto que un determinado usuario puede establecer, por ejemplo, que enero de 2015 tiene 21 días lectivos, y al minuto siguiente entrar y ver un valor de 20 porque otro usuario lo ha cambiado para otro grupo, se considera que el permitir una modificación abierta beneficia más que perjudica. Si se piensa bien, los días lectivos son algo fijo e inmutable, y por tanto el primer usuario que lo configure correctamente estará evitando trabajo redundante a otros usuarios para otros grupos. El único requisito lógico es la confiabilidad entre los usuarios (todos somos adultos y responsables). Aunque hasta este punto es irrelevante si tenemos en cuenta que se puede modificar en cualquier momento: si un profesor encuentra una incorrección, la modifica, calcula el índice y asunto solucionado.

Además, dejar libre la introducción de los días lectivos permite hacer pruebas con los índices, de forma que si en algún momento se quiere el índice calculado no por los días lectivos, sino por los naturales, sólo tendría que entrar y modificarlos desde la ficha establecida para tal fin (eso sí, acordarse de volver a dejarlo como estaba para no molestar a otros usuarios)

El único conflicto real viene determinado por la concurrencia al segundo, es decir, que dos usuarios cambien con valores diferentes el mismo calendario, en el mismo momento, justo antes de calcular el índice correspondiente. En este caso, y sintiéndolo mucho, sólo podemos decir una cosa: qué mala suerte, inténtelo de nuevo pasado unos minutos.

El método de introducción de los datos es tan fácil como añadir el valor numérico de los días lectivos en cada una de las cajas de texto habilitada para cada mes en la página del calendario (Fig. 58).

The screenshot shows a web interface for entering school days. At the top, there's a navigation bar with 'PFC 2015' and 'Hola Oscar'. Below it, a menu has 'Datos', 'Índices', and 'Calendario'. The main content area is titled 'Calendario 2014-2015' and contains the following text:

Introduzca los días lectivos del calendario escolar.
Estos días se utilizan para realizar cálculos de índices y estadísticas siendo, por tanto, un dato necesario y muy relevante que puede llegar a desvirtuar los resultados finales en caso de no ser correctos.
Los meses se muestran en orden natural (de Enero a Diciembre), aunque debe tener en cuenta que los primeros meses del año (de Enero a Agosto) corresponden al tramo final del año lectivo, y el resto al tramo inicial del mismo (de Septiembre a Diciembre)

| | | | | | | | |
|------------|----|---------|----|-----------|----|-----------|----|
| Enero | 18 | Febrero | 20 | Marzo | 19 | Abril | 13 |
| Mayo | 20 | Junio | 22 | Julio | 23 | Agosto | 0 |
| Septiembre | 22 | Octubre | 21 | Noviembre | 20 | Diciembre | 15 |

At the bottom left, there is a 'Guardar' button with a checkmark icon.

Fig. 58 – Calendario de días lectivos para el grupo seleccionado.

Puede darse el caso de que el calendario para ese curso lectivo no exista en la base de datos antes de calcular el índice. En este caso se creará de forma automática, introduciendo un valor de 20 días como valor estimado para cada uno de los meses.

De la misma forma, la primera vez que se entra a la página del calendario, en caso de no existir, se creará automáticamente con el mencionado valor de 20 días lectivos para cada uno de los 12 meses, siendo el usuario el que posteriormente tendrá que modificar dichos valores por los correctos.

5.5 Conclusiones

Si bien se va a ahondar en el tema de las conclusiones en el siguiente punto, podríamos mencionar en este apartado algunos corolarios extraídos del uso de las tecnologías empleadas.

Por un lado destacar el uso de Visual Studio como entorno de desarrollo para Python, cumpliendo casi a la perfección con las necesidades básicas de trabajo con Python y Django. Salvo los incidentes ya mencionados con las divergencias a la hora de tratar las excepciones en dos máquinas con una configuración muy similar, no he tenido mayor problema en ninguna de las fases del desarrollo para implementar la aplicación.

Por otro lado, tanto los módulos de Python utilizados (para conectar con Facebook, Neo4j y crear hojas de Excel) tienen una curva de aprendizaje muy rápida y fácil de asimilar, y están ampliamente testados como para considerarlos estables y fiables.

En general todas las herramientas han sido benévolas con el desarrollo y el desarrollador, exceptuando casos puntuales de la API de Facebook que, quizás debido a su complejidad, no llegan a comportarse como deben tal y como se recoge en la documentación de la API (prueba de ello son las estadísticas de bugs abiertos que ronda la cifra entre 600 y 1000.

<https://developers.facebook.com/bugs/>)

6 Conclusiones

6.1 Trabajo Desarrollado

Llega el momento de sacar conclusiones finales sobre todo el proceso de análisis e implementación del proyecto.

Con el fin de dar una visión global del escenario sobre el que se ha desarrollado el proyecto, y atendiendo únicamente a las implicaciones tecnológicas, cabe decir que me he enfrentado voluntariamente a determinados retos que han supuesto un enriquecimiento personal. En el momento de iniciar el proyecto tuve la opción de elegir un lenguaje de programación en el que tengo sobrada experiencia y que me hubiera facilitado enormemente el trabajo, sin embargo, tomé la decisión de formarme en otro lenguaje con otra filosofía muy diferente como es el caso de Python y de Django.

Esta decisión supuso aumentar el tiempo de estudio, el tiempo de investigación y ciertos escollos a la hora de utilizar técnicas de programación específicas, o a la hora de configurar los propios entornos de trabajo: pasé por la elección de Python 3.4, Pycharm como plataforma de desarrollo que posteriormente tuve que desechar porque su edición libre (no de pago) no tenía soporte nativo para Django, varios módulos de trabajo con Facebook que me hicieron cambiar de versión de Python a la 2.7 por incompatibilidad con la 3.4, Django 1.8 que finalmente tuve que cambiar por la versión 1.7, hasta encontrar un todo estable y acabar utilizando Visual Studio .Net como IDE.

Como usuario neófito de Python sólo puedo decir que, lo que gana debido a ser un lenguaje abierto y ampliamente difundido entre una muy extensa comunidad de desarrolladores, lo complica precisamente por el mismo motivo: es muy fácil encontrar incompatibilidades entre módulos, e incluso versiones de los propios módulos.

El segundo gran reto, este sí propiamente parte del proyecto, fue el estudio de una API de terceros, la API de Facebook. En este aspecto decir que el proceso por el cual Facebook te obliga a darte de alta como desarrollador (con el consecuente gasto económico de un SMS, dependiendo de la compañía telefónica y el plan que tengas contratado), crear una aplicación en su sistema y morir siempre en su forma de trabajar, si bien es algo lógico desde el punto de vista empresarial, resulta un tanto pesado y excesivo, sobre todo si tenemos en cuenta que es para un proyecto de ámbito educacional, no comercial.

Es cierto que la documentación que ofrecen para llevar a cabo todo este proceso, además de las guías de referencia para el uso de la API, son bastante detalladas y útiles. Sin embargo, una

vez metidos en materia, una vez empiezas a trabajar con la API de Facebook, a realizar consultas y extraer datos, sacas dos conclusiones claras:

1. El sistema de trabajo es tan grande, con tantos frentes abiertos (páginas, grupos, entradas, comentarios, usuarios...), que irremediamente hay inconsistencias entre lo que dicen que hacen, y lo que realmente hacen. Durante el desarrollo nos encontramos con varios ejemplos de este tema como por ejemplo la incapacidad de obtener las entradas compartidas utilizando la arista (Edge) “/{post-id}/sharedposts”
2. La gestión de permisos es muy compleja y ramificada a varios niveles, existiendo tanto permisos genéricos de acceso a perfiles personales, de grupo o página, como específicos para determinadas acciones sobre los nodos principales.

A nivel del desarrollo propiamente dicho me quedo con la sensación de no haber tenido tiempo material para depurar el código. Al ser un entorno de trabajo totalmente nuevo, con un lenguaje nuevo, he ido aplicando los conocimientos que iba adquiriendo en libros y páginas de internet paulatinamente, pero sin poder llegar a reescribir código anterior por no poder invertir el tiempo suficiente. Es probable que, viendo el código fuente, se encuentren partes en las que la codificación sea muy tosca y mejorable, mientras que otras se ajustan más a lo que en la comunidad Python llaman ser “Pythonic”. Independientemente de ello el trabajo final es bueno desde un punto de vista funcional, aunque desde un punto de vista personal, y con mi experiencia como desarrollador en otros lenguajes de programación (Visual Basic .NET), me quedo con la sensación de que es mejorable.

Tengo que destacar también el uso de Visual Studio .NET como herramienta de trabajo. En un principio me resistía a la idea de usar este IDE como plataforma de desarrollo, no en vano Microsoft tiene una bien merecida fama en lo referente a la compatibilidad con algo que no es suyo, sin embargo me lo propuse como objetivo secundario del proyecto: analizar cómo se podía trabajar en un entorno Windows, con herramientas que no respetan la ideología de código libre, con lenguajes y tecnologías que son totalmente lo contrario.

A este respecto debo decir que la experiencia ha sido positiva. Es cierto que los inicios siempre son complicados, necesitando de la instalación de varias versiones de las Python Tools for Visual Studio (las versiones más estables funcionan sobre versiones de Visual Studio más actuales), varias creaciones de soluciones que hubo que descartar por no entender muy bien cómo funcionaba el entorno virtual de Python, y errores de compatibilidad con Django 1.8 que requirió utilizar la versión anterior del mismo, la 1.7. Pero cuando finalmente se encontró la configuración adecuada, el desarrollo de la aplicación resultó mucho más sencillo, sobre todo teniendo en cuenta que es el IDE que uso a diario y que conozco muy bien.

En general, y como conclusión subjetiva, estoy bastante satisfecho con el trabajo realizado, con el aprendizaje de nuevos lenguajes, nuevas tecnologías y una API muy útil hoy en día (las redes sociales están de moda) pero, y a pesar de la gran cantidad de tiempo invertido en todas las etapas del proyecto, me quedo con ganas de más: de finalizar un producto realmente redondo, bien acabado y realmente útil. Prueba de ello es el apartado 6.3 Ampliaciones Futuras, donde pongo de manifiesto todo lo que me hubiera gustado plasmar en el producto final de haber tenido más tiempo.

En resumen, creo que ha sido un trabajo muy enriquecedor a nivel personal y no sólo un simple proyecto desarrollado con el fin de cubrir expediente. Y este, creo yo, es el mejor objetivo que podría tener un proyecto final de carrera.

6.2 Aportaciones

Desde una perspectiva funcional creo que este proyecto podría servir de punto de partida para la creación de un sistema completo de gestión de grupos de Facebook a nivel docente.

Es un buen primer paso que abre el camino para tratar de forma directa con la API de Facebook y sus peculiaridades, así como para analizar el uso de grupos como herramienta colaborativa y de enseñanza.

Actualmente permite la gestión de toda la información relevante de los grupos, la creación y actualización de un modelo de datos específico de un SGBD NoSQL orientado a grafos, y la recuperación de tres índices que permiten evaluar el uso de los grupos dentro de un determinado marco lectivo.

Sin embargo creo que para poder llegar a algo realmente útil requiere de más trabajo, ampliando las estadísticas e índices que podrían extrapolarse de los datos obtenidos, estando atentos a los cambios de la API de Facebook para poder obtener todavía más datos que ahora mismo están en un limbo de inconsistencia entre lo que debería de ser, y lo que es, y añadiendo análisis de grafos y teoría de redes para, finalmente, dar una visión mucho más amplia de cómo y por qué es bueno emplear los grupos de Facebook como parte del proceso educativo universitario.

Por otro lado, el propio trabajo desarrollado como por ejemplo el análisis estratégico realizado con el fin de estudiar la existencia de herramientas similares, analizando sus fortalezas y debilidades; el documento de especificación de requisitos; el análisis de la API de acceso a Facebook y su utilización; e incluso la bibliografía incluida en el último capítulo, pueden y deben ser consideradas como aportaciones del proyecto. Hay que tener en cuenta que cualquier persona, sea o no sea desarrollador experimentado, se enfrentará a los mismos problemas y

exigencias en caso de verse ante la necesidad de realizar una aplicación de estas características, siendo este documento y todo el material adicional generado (como por ejemplo el código fuente), una guía que puede servir a modo de introducción y clarificación de muchos conceptos y técnicas que de otra forma podrían suponer un incremento en la curva de aprendizaje previa a la especificación e implementación de la aplicación.

6.3 Ampliaciones Futuras

Como he mencionado anteriormente se han quedado muchas ideas en el tintero, ideas que habrían dotado de una mayor profundidad y utilidad a la aplicación, pero que debido a las exigencias temporales no se han podido llevar a cabo. Huelga decir que esto no supone, bajo ningún concepto, que el trabajo realizado haya sido escaso o sencillo, nada más lejos de la realidad. Espero que se entienda como lo que es: una forma de preocuparse por mejorar un sistema ya de por sí complejo, y una demostración de que cualquier aplicación puede aumentar exponencialmente su utilidad si se le dedica un poco de tiempo a pensar en cómo hacerlo.

Gestión de Usuarios

Actualmente sólo hay un tipo de usuario (obviando al usuario anónimo), que es el usuario validado en Facebook.

Creo que sería conveniente generar un sistema de usuarios específico para la aplicación, con roles determinados, alta en el sistema, y validación de acceso.

Si analizamos el desarrollo nos encontramos con una limitación en cuanto al uso y análisis de grupos, y es que cada usuario que ha iniciado sesión en el sistema sólo puede ver sus propios grupos de Facebook. Pero, ¿qué pasaría si se quisiera analizar o comparar grupos diferentes desde la propia aplicación? ¿Y si un determinado profesor quisiera ver las estadísticas de un grupo al que no pertenece? Simplemente no podría. Es por eso que debería haber un sistema de usuarios de la aplicación de forma que se pudiera otorgar permisos de acceso a determinados grupos, quizás por departamento, quizás por orden directa del administrador del grupo, o por cualquier otro método que se definiera consensuado con los usuarios finales de la aplicación.

Es más, ¿qué pasará cuando el sistema esté lleno de grupos de años anteriores? También debería poder desactivarse los grupos para que, sin borrarlos de la base de datos, no aparecieran en la página de selección de grupo.

Todas estas limitaciones actuales se verían solucionadas generando un modelo de datos paralelo a Neo4j, con tablas relacionales que permitieran asignar grupos a usuarios, acciones posibles a determinados usuarios, y otras muchas funcionalidades que habría que analizar.

Ampliación de índices, estadísticas y entidades

Otro punto importante sería ampliar la parte de extracción y procesado de la información para mostrar nuevos índices, nuevas estadísticas de utilización, y nuevas entidades que doten a la aplicación de una mayor utilidad informativa.

Actualmente sólo se muestran tres índices con gráficas estáticas, pero usando la misma lógica de trabajo se podrían añadir otros muchos, simplemente se requeriría elegir qué mostrar e implementarlo.

Por otro lado, se ha prestado especial atención al grupo como entidad de trabajo, pero creo que añadir otras entidades como el usuario del grupo sería una buena ampliación. Puesto que ya tenemos toda la información en la base de datos, ¿por qué no realizar una ficha por usuario (alumno) dónde mostrar información como el número de entradas que ha creado, el número de comentarios, el número de “me gusta” e índices individualizados de actividad? Sería una buena herramienta de evaluación para el profesor.

Otra posibilidad sería crear una entidad para los eventos, de forma que se obtenga categorizada toda la información de determinado evento (como por ejemplo la realización de una práctica), con índices específicos del impacto de este evento dentro del grupo, o a nivel del alumnado.

Los datos ya están ahí, sólo sería necesario tiempo y consensuar qué índices, estadísticas y valores mostrar para cada entidad mencionada.

Generación de grafos y estudio de redes

Uno de las motivaciones básicas para la selección de Neo4j como sistema de gestión de base de datos es la facilidad para la generación de grafos a partir de los datos incluidos en su base de datos.

A partir de esto, y teniendo en cuenta que la teoría de redes es un submundo complejo que requiere de tiempo de estudio, su utilización para analizar el grupo desde un punto de vista colaborativo entre alumnos, o alumnos y profesor, otorgaría al proyecto de herramientas visuales muy potentes y de información muy útil.

Para ello propondríamos usar un módulo Python como iGraph para generar los grafos y obtener los datos relevantes de la red generada, y el propio D3.js ya empleado para visualizar los índices como motor para plasmar dichos grafos en cualquier explorador web.

Es más, si unificáramos la utilización de Facebook con la de Twitter (sobre el cual hay otros proyectos abiertos y en desarrollo), se podría generar información conjunta de ambas redes sociales, con el enorme interés que ello supone.

Mejoras en el diseño

Como se ha mencionado durante el proyecto el verdadero núcleo de interés del mismo era la obtención y extracción de datos de Facebook vía su API. Es por ello que se ha dejado en un segundo plano el aspecto visual de la aplicación, tomando como objetivo simplemente la visualización de los datos para demostrar que se ha llevado a cabo correctamente el proceso de importación.

Es por esto que, con muy poco tiempo y esfuerzo, se podría mejorar enormemente el estilo visual de la misma: desde un diseño más vistoso y funcional; un sistema de menús más adecuado; un sistema de visualización de errores más potente; o la inclusión de elementos javascript para, por ejemplo, mostrar barras de progreso en el proceso de importación a base de datos.

También se puede ahondar mucho en la utilización de D3.js como generador de gráficas. Con un poco de tiempo se pueden llegar a hacer gráficas dinámicas e interactivas que mejorarían notablemente la experiencia del usuario.

Mejoras en la implementación

En el terreno de la implementación, o de la propia codificación de la aplicación, se podrían llevar a cabo determinadas tareas que mejorarían el rendimiento.

La primera y más importante sería la de utilizar consultas por lotes en las llamadas a la API de Facebook. Graph API es una API basada en consultas HTTP y, como tal, requiere de muchas llamadas para obtener todos los datos de un grupo: datos del propio grupo, de sus usuarios, de sus entradas, sus comentarios, eventos. Estas llamadas HTTP tienen bastantes opciones de configuración o formas de utilizarlas, y una de ellas es agrupando los diferentes identificadores de los objetos que se desean consultar en una misma llamada.

Si optáramos por este método en lugar de hacer una llamada por cada entrada del grupo, se podría hacer una única llamada para todas las entradas del grupo, ahorrando notablemente en tiempo de ejecución, aunque ello suponga complicar ligeramente el algoritmo para tratar la información obtenida.

En un primer momento opté por la solución simple (un objeto = una consulta HTTP) por facilitar el aprendizaje tanto de la API, como del lenguaje de programación, sin embargo, una vez adquirida soltura en ambos campos sí que rescribí la importación de usuarios utilizando la aproximación de llamadas por lotes, experimentando una mejoría en el tiempo del algoritmo.

Es decir, las llamadas agrupadas funcionan y deberían aplicarse sí o sí como mejora de la aplicación, cosa que yo mismo habría hecho de disponer de más tiempo para la consecución del proyecto.

Por otro lado se debería crear un proceso automático de importación de los grupos ya que alguno de los elementos importados (por ejemplo los “me gusta”) no son nodos como tal, lo cual supone que no tengan una fecha de creación importable. Esto supone que la fecha establecida para cada uno de estos registros es la propia fecha de la importación, de forma que los índices que requieran fiabilidad en cuanto a las fechas en las que se han realizado determinadas acciones podrían verse afectados.

Poniendo un ejemplo: si el último día del mes en un grupo se han dado 50 “me gusta” a una entrada, y no se realiza la importación hasta el día siguiente, estos “me gusta” contarán no como el día en el que se han dado porque no hay una fecha recuperable por la API de Facebook, sino en el día que se ha importado el grupo a la base de datos, de forma que las estadísticas se verían afectadas contabilizando 50 “me gusta” en un mes que no le corresponde.

Esto sería fácilmente solucionable si se creara un proceso automático de importación con una periodicidad diaria, o si se concienciara a los profesores para importar los grupos a diario, aunque depender del factor humano para un correcto funcionamiento de la aplicación no siempre es recomendable.

7 Referencias

1. Wikipedia. *Django (framework)*. [Publicación seriada en línea]. Fecha última actualización: 2015-05-13, 20:55:00. [Consulta: 2015-05-15].
Disponible en: [https://es.wikipedia.org/wiki/Django_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework))
2. Wikipedia. *Modelo-vista-controlador*. [Publicación seriada en línea]. Fecha última actualización: 2015-06-17, 15:38:00. [Consulta: 2015-07-21].
Disponible en:
<https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>
3. Schellenbach, Thierry. *Django Facebook*. [Publicación seriada en línea]. Revisión 60003f62d79dfbe6f910d39d0b3cbe70de3c2462. [Consulta: Abril-Septiembre 2015].
Disponible en: <http://django-facebook.readthedocs.org/en/latest/index.html>
4. Edwards, Robin. *Neomodel documentation*. [Publicación seriada en línea]. 2014.
[Consulta: Abril-Septiembre 2015].
Disponible en: <http://neomodel.readthedocs.org/en/latest/>
Disponible en: <https://github.com/robinedwards/neomodel>
5. The Neo4j Manual. *Cypher Query Language*. [Guía de referencia Web]. Versión 2.2.1.
[Consulta: Mayo-Septiembre 2015].
Disponible en: <http://neo4j.com/docs/2.2.1/cypher-getting-started.html>
Disponible en: <http://neo4j.com/docs/2.2.1/cypher-query-lang.html>
6. Chrisw, sjmachin. *Xlwt*. [Módulo software] [Publicación seriada en línea]. Versión 1.0.0. [Consulta: Julio 2015].
Disponible en: <https://pypi.python.org/pypi/xlwt>
Disponible en: <http://xlwt.readthedocs.org/en/latest/>
7. Bootstrap. [Módulo software] *Bootstrap Components*. [Página Web]. Versión 3.3.5.
[Consulta Julio-Septiembre 2015].
Disponible en: <http://getbootstrap.com/components/>
8. D3.js. [Módulo software]. *D3 Data-Driven Documents*. [Página Web]. Versión 3.5.6.
[Consulta Julio-Septiembre 2015].
Disponible en: <http://d3js.org/>
9. Facebook Developers. *The Graph API*. [Guía de referencia Web]. Versión 2.4.
[Consulta Abril-Septiembre 2015].
Disponible en: <https://developers.facebook.com/docs/graph-api>
10. Microsoft. *Python Tools for Visual Studio*. [Módulo Software]. [Página Web]. Versión 2.1 para Visual Studio 2012.

- Disponible en: <http://microsoft.github.io/PTVS/>
11. Neo Technology, Inc. *Neo4j*. [Programa] [Página Web] Versión 2.3.0 para Windows.
Disponible en: <http://neo4j.com/>
 12. Poza Luján, Jose Luis 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. ISSN 1578-7680.
Disponible en: <http://www.um.es/ead/red/44/Poza-Lujan.pdf>
 13. Network Graphs. The Social Media Research Foundation. *NodeXL Excel Template 2014*. [Programa] [Página Web] Última actualización 2014-06-23, 09:00:00.
Disponible en: <http://nodexl.codeplex.com/>
 14. Wolfram Research Company. *Wolfram|Alpha Personal Analytics for Facebook*. [Programa En Línea] Última Actualización Abril 2015. [Consulta: Abril 2015]
Disponible en: <http://www.wolframalpha.com/facebook/>
 15. Rieder, B. *Studying Facebook via data extraction: the Netvizz application*. WebSci 2013 Proceedings of the 5th Annual ACM Web Science Conference (pp. 346-355). New York: ACM.
Disponible en: http://rieder.polsys.net/files/rieder_websci.pdf
Disponible en: <https://apps.facebook.com/netvizz/>
 16. Lubanovic, Bill. *Introducing Python. Modern Computing in Simple Packages*. Sebastopol, California (United States of America). O'Reilly Media, Inc. Noviembre 2014. 478 p. ISBN 978-1-4493-5936-2.
 17. Cecil Martin, Robert. *UML for Java Programmers*. Englewood Cliffs, New Jersey (United States of America). Prentice-Hall, Inc. ca. 2002. 236 p. ISBN 0-13-203837-4
 18. Bendoraitis, Aidas. *Django Tricks. How to Export Your Data as CSV, XLS, or XLSX*. [Blog]. Diciembre 2013. [Consulta: Julio 2015]
Disponible en: <http://djangotricks.blogspot.com.es/2013/12/how-to-export-data-as-excel.html>
 19. Autores Varios. *Análisis de redes sociales y Facebook como herramienta de trabajo universitario*. [Blogs]. [Consulta: Abril-Septiembre 2015].
Disponible en: <http://networking.marketing-xxi.com/ars-analisis-redes-sociales.html>
Disponible en: <http://www.socialancer.com/>
Disponible en: https://en.wikipedia.org/wiki/Social_network_analysis
Disponible en: <https://techknowtools.wordpress.com/2012/06/14/facebook-for-learning-communities-groups-vs-pages/>

Disponible en: <http://www.onlinecollege.org/2012/05/21/100-ways-you-should-be-using-facebook-in-your-classroom-updated/>

20. Python Software Foundation. *Python documentation*. [Publicación seriada en línea] Versión 3.4.3. Última actualización 2015-09-23. [Consulta: Abril-Septiembre 2015]. Disponible en: <https://docs.python.org/3/>