



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

TEMA:

SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE CONTENIDO
ACADÉMICO PUBLICADO EN REDES SOCIALES PARA LA FACULTAD
DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL.

Proyecto de Trabajo de Graduación. Modalidad: Proyecto de investigación, presentado previo
a la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLÍNEA DE INVESTIGACIÓN:

Ingeniería de Software

Autor: Diego Fernando León Solís
Tutor: Dr. Félix Oscar Fernández Peña

Ambato - Ecuador

ENERO, 2017

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema: “Sistema de información para la gestión de contenido académico publicado en redes sociales para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”, del señor Diego Fernando León Solís, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, enero de 2017

EL TUTOR

Dr. Félix Oscar Fernández Peña

AUTORÍA

El presente proyecto de investigación titulado: “Sistema de información para la gestión de contenido académico publicado en redes sociales para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial” es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, enero de 2017

Diego Fernando León Solís

CC: 180506007-4

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, enero de 2017

Diego Fernando León Solís

CC: 180506007-4

APROBACIÓN COMISIÓN CALIFICADORA

La comisión calificadora del presente proyecto conformada por los señores docentes Ing. Carlos Núñez e Ing. Patricio González, revisó y aprobó el Informe Final del proyecto de graduación titulado “Sistema de información para la gestión de contenido académico publicado en redes sociales para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”, presentado por el señor Diego Fernando León Solís de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, enero de 2017

Ing. Mg. Vicente Morales
PRESIDENTE DEL TRIBUNAL

Ing. Carlos Núñez
DOCENTE CALIFICADOR

Ing. Patricio González
DOCENTE CALIFICADOR

DEDICATORIA

La concepción de este proyecto está dedicada a mi madre, un pilar fundamental en mi vida.

Sin ella, jamás hubiese podido conseguir lo que hasta ahora he logrado.

Su tenacidad y lucha insaciable han hecho para mi un gran ejemplo a seguir y destacar.

Diego F. León Solís

AGRADECIMIENTO

A Dios por darme la fuerza y la salud para proseguir en el transcurso de carrera universitaria.

A mi madre, por formarme siempre buenos valores de esfuerzo y paciencia. A mi familia brindarme siempre su respaldo de forma incondicional.

A la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato por darme la oportunidad de formarme como persona y profesional.

A la Ing. Pilar Urrutia, por su ayuda y colaboración en la concepción y desarrollo del proyecto.

Especial agradecimiento a mi Director de Tesis el Dr. Félix Fernández por sus consejos, y apoyo en todo este proceso.

Diego F. León Solís

RESUMEN

Este proyecto de investigación se centró en el estudio de la relación entre el uso de Facebook, como plataforma de redes sociales, el acceso y difusión de contenido académico.

Las redes sociales hoy en día permiten un amplio grado de difusión de información pero no fueron creadas para la gestión académica. Se hace necesario incrementar la usabilidad de los mecanismos que facilitan la diseminación del conocimiento presente en el contenido académico a través de redes sociales.

En el presente trabajo se describe el diseño e implementación de un sistema de información para la gestión de contenido académico publicado en Facebook. En su construcción se utilizó JavaScript como lenguaje de programación en el cliente y servidor, también se dio utilización como lenguaje de datos, con lo que se garantizó una implementación software libre de código abierto.

Al tratar con el acceso a un sistema de información que se ejecuta dentro del ambiente de Facebook, es posible cuantificar la calidad y pertinencia de los contenidos académicos en función de las acciones sociales que se realizan sobre ellos. Las implicaciones fundamentales de la propuesta son analizadas en este documento.

Palabras Claves: Red social, Facebook, academia, javascript

ABSTRACT

This research project focuses in studying the relationship between making use of Facebook, as social media, and the broadcasting of academic content.

Nowadays, social networks make possible spreading a huge amount of information but they were not conceived for the academia. So, it is necessary to increase the usability of mechanisms for the dissemination of the knowledge inherent to academic resources through social networks.

In this work, the design and implementation of an information system for managing academic content on Facebook is described. The proposed solution was carried out by using JavaScript for client and server programming and as data language. This way, a free open-source implementation was guaranteed.

Been a Facebook application, it is possible to quantify the quality of academic content been shared on Facebook. Main outcomes of the proposal are analysed in this document.

Keywords: Social network, Facebook, academic, javascript

GLOSARIO DE TÉRMINOS

BD Base de Datos

ORM Mapeo objeto-relacional (Object-Relational mapping)

API Interfaz de programación de aplicaciones (Application Programming Interface)

RESTful Transferencia de Estado Representacional (Representational State Transfer).

HTTPS Protocolo seguro de transferencia de hipertexto (Hypertext Transfer Protocol Secure)

DOM Modelo de Objeto de Documento (Document Object Model). Es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

FISEI Facultad de Ingeniería en Sistemas, Electrónica e Industrial

SOMAN “Sistema de Gestión Académica en las Redes Sociales”

Open Source Código abierto

Back-end Se determina como la parte que procesa la entrada de datos desde el front-end. Son los procesos en el sistema que se encargan de resolver las peticiones de los usuarios.

Esta área es también la encargada de optimizar recursos, de la seguridad de un sitio y demás. Cosas que el usuario no ve de primeras pero que existe código detrás que supe a estas necesidades.

Front-end Se determina como la parte del software que interactúa con el o los usuarios. Se incluye todas aquellas tecnologías que se ejecutan en el lado del cliente, es decir, las tecnologías que tienen su ejecución en navegadores web.

SPA Se conocen como el paradigma de arquitectura de software de aplicaciones de una sola página (Single Page Application).

AJAX Es el acrónimo de Asynchronous Javascript and XML, se define como Javascript y XML Asíncrono.

Facebook Plataforma web de redes sociales

Facebook Canvas Área de interacción de las aplicaciones dentro de la plataforma de Facebook

Node.js Entorno en tiempo de ejecución multiplataforma para la capa del servidor «back-end»

NPM Módulo de manejo de paquetes de NodeJS (Node Package Module).

Usuario Persona que usará el sistema para gestionar procesos

ÍNDICE

Aprobación del tutor	II
Autoría	III
Derechos de autor	IV
Aprobación comisión calificadora	V
Dedicatoria	VI
Agradecimiento	VII
Resumen	VIII
Abstract	IX
Glosario de términos	X
Introducción	xviii
CAPÍTULO 1 El Problema	1
1.1 Tema	1
1.2 Planteamiento del problema	1
1.3 Delimitación	3
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 General	4
1.5.2 Específicos	4
CAPÍTULO 2 Marco Teórico	6
2.1 Antecedentes Investigativos	6
2.2 Fundamentación teórica	7
2.2.1 Web 2.0	7
2.2.2 Red Social	8
2.2.3 Sistema de Información	8
2.2.4 Base de Datos	9
2.2.5 Aplicación Web	9
2.2.6 Servicio Web	9
2.2.7 Modelo - Vista - Controlador	10
2.2.8 Modelo - Vista - Vista Modelo	10
2.2.9 API	11
2.2.10 Facebook	12

2.2.11	Node.js	13
2.2.12	Sail.js	13
2.2.13	Bootstrap	13
2.2.14	AngularJS	14
2.2.15	Usabilidad	15
2.2.16	Principios Heurísticos de la Usabilidad Web	16
2.3	Propuesta de Solución	18
CAPÍTULO 3 Metodología		19
3.1	Modalidad de la investigación	19
3.2	Recolección de información	19
3.3	Procesamiento y análisis de datos	19
3.4	Desarrollo del proyecto	20
CAPÍTULO 4 Desarrollo de la propuesta		21
4.1	Análisis de requerimientos del sistema de información	21
4.1.1	Levantamiento de requerimientos	21
4.2	Determinación del alcance del sistema	22
4.2.1	Definición de ámbitos del sistema	22
4.3	Diseño de arquitectura del sistema de información	23
4.3.1	Diseño de modelo lógico de base de datos	23
4.3.2	Diseño de modelo físico de base de datos	24
4.3.3	Elaboración diagrama de casos de uso	25
4.3.4	Comportamiento dinámico de los procesos del sistema	26
4.4	Desarrollo del sistema de información	32
4.4.1	Especificación de casos de uso	32
4.4.2	Producción de software	40
4.4.3	Transición	76
CAPÍTULO 5 Conclusiones y Recomendaciones		79
5.1	Conclusiones	79
5.2	Recomendaciones	80
Bibliografía		81
ANEXOS		88

ÍNDICE DE FIGURAS

1	Modelo-Vista-Controlador [1]	10
2	Modelo - Vista - Vista Modelo [2]	11
3	Servicio Web Rest API [3]	12
4	Modelo de datos del API de Facebook [4]	12
5	Arquitectura de los componentes de AngularJS [53]	14
6	Arquitectura de los componentes de AngularJS [55]	15
1	Modelo lógico de la base de datos	23
2	Modelo físico de la base de datos	24
3	Diagrama de casos de usos del sistema	25
4	Diagrama de actividades. Iniciar sesión	26
5	Diagrama de actividades. Visualizar recursos de contenido	27
6	Diagrama de actividades. Buscar contenidos	27
7	Diagrama de actividades. Registrar acción «visita»	28
8	Diagrama de actividades. Registrar un nuevo contenido	28
9	Diagrama de actividades. Realizar acción «me gusta»	29
10	Diagrama de actividades. Eliminar «me gusta»	29
11	Diagrama de actividades. Realizar acción «compartir»	30
12	Diagrama de transición de estados. Sesión	31
13	Diagrama de transición de estados. Acción «me gusta»	31
14	Diagrama de transición de estados. Acción «share»	32
15	Arquitectura cliente/servidor tradicional [5]	40
16	Arquitectura cliente/servidor SPA [6]	40
17	Arquitectura del sistema Sigma	42
18	Estructura de árbol de directorios del sistema Sigma	43
19	Diagrama de secuencias. Iniciar sesión	44
20	Diagrama de secuencias. Registrar acción «visita»	45
21	Diagrama de secuencias. Buscar contenidos	45
22	Diagrama de secuencias. Registrar un nuevo contenido	45
23	Diagrama de secuencias. Visualizar recursos de contenido	46
24	Diagrama de secuencias. Realizar acción «compartir»	46
25	Diagrama de secuencias. Realizar acción «me gusta»	47
26	Diagrama de secuencias. Eliminar «me gusta»	47
27	Diagrama de clases. Programación del servidor	48

28	Diagrama de clases. Programación del cliente	49
29	Interfaz de Sigma. 1) Dirección URL. 2) Identificación de aplicación en Facebook. 3) Opciones de filtrado de contenidos. 4) Metadatos. 5) Área de visualización. 6) Acciones sociales disponibles. 7) Agregar un nuevo contenido.	50
30	Extracto de código. Cliente - Realizar acción «me gusta»	51
31	Extracto de código. Servidor - Realizar acción «me gusta»	52
32	Ventana modal para compartir en el muro social del usuario.	53
33	Extracto de código. Cliente - Realizar acción «compartir»	54
34	Extracto de código. Servidor - Realizar acción «compartir»	54
35	Extracto de código. Cliente Registrar un nuevo contenido	55
36	Áreas de acceso a visualización de origen. 1) Visualizar en una nueva pestaña. 2) Visualizar dentro del sistema en una ventana modal.	57
37	Ventana modal.	57
38	Extracto de código. Cliente. Visualización de sitio de origen en una nueva pestaña.	58
39	Extracto de código. Servidor Descargar sitio web de origen	58
40	Extracto de código. Cliente Registrar «visita»	59
41	Extracto de código. Servidor Registrar «visita»	59
42	Interfaz. Opciones en componente de filtrado de contenidos. 1) Recientes. 2) Recomendado. 3) Con más acciones de “me gusta”. 4) Con más acciones de “compartir”. 5) Por categoría.	60
43	Extracto de código. Servidor Contenidos con mayor número de acciones «me gusta»	61
44	Extracto de código. Cliente Contenidos con mayor número de acciones «me gusta»	61
45	Mensaje de error en la de búsqueda de contenidos	63
46	Interfaz de búsqueda de contenidos	63
47	Fuentes tipográficas seguras	64
48	Fuentes tipográficas del sistema Sigma	65
49	Paleta de colores de Facebook	66
50	Paleta de colores del sistema Sigma.	66
51	Colores en la interfaz del sistema Sigma.	67

52	Elementos de los contenidos del sistema Sigma. 1) Título. 2) Autor de contenido. 3) Fecha. 4) Tiempo de lectura. 5) Contenidos relacionados. 6) Descripción. 7) Visualizar sitio web de origen. 8) Número de Visitas. 9) Categorías que se relacionan. 10) Me gusta. 11) Compartir.	67
53	Retícula aplicada a los contenidos del sistema Sigma	68
54	Retícula aplicada en la ventana de visualización de contenidos. . .	69
55	Estado de la ejecución del sistema Sigma en pm2	70
56	Detalles de la ejecución del sistema Sigma en pm2	70
57	Interfaz del sistema Sigma a 1280x890	71
58	Interfaz del sistema Sigma a 640x960	72
59	Autenticación OAuth de Facebook [7]	73
60	Ventana modal de acceso a permisos a la información del usuario.	74
61	Ventana modal de aceptación de permisos para publicar en nombre del usuario.	74
62	Carga de recursos en entorno de desarrollo.	75
63	Carga de recursos en entorno de producción.	75

ÍNDICE DE TABLAS

1	Caso de uso. Registrar “visita”	32
2	Caso de uso. Inició de sesión	33
3	Caso de uso. Visualizar recursos de contenidos	34
4	Caso de uso. Visualizar recursos de contenidos	35
5	Caso de uso. Filtrar contenidos	35
6	Caso de uso. Buscar contenidos	36
7	Caso de uso. Gestionar los “me gusta”	36
8	Caso de uso. Realizar la acción “me gusta”	37
9	Caso de uso. Eliminar un “me gusta”	38
10	Caso de uso. Realizar la acción de “compartir”	39
11	Sintaxis de búsqueda Sigma.	62
12	Ejemplo de proceso de selección de resultados de búsqueda	63
13	Cuadro comparativo entre gestores de procesos en Node.js	70

INTRODUCCIÓN

El presente proyecto de investigación “Sistema de información para la gestión de contenido académico publicado en redes sociales para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”, consta de cinco capítulos los cuales se detallan a continuación:

Capítulo I. “El Problema”, se identifica el problema que se suscita en un contexto de la realidad, para plantearlo de forma concreta, delimitando el alcance, con una respectiva justificación y el planteamiento de los objetivos que guiarán todo el proyecto.

Capítulo II. “Marco Teórico”, consta del fundamento teórico que ayuda a comprender de forma clara el problema gracias a los antecedentes investigativos, para luego plantear la propuesta de solución.

Capítulo III. “Metodología”, Se describe las metodologías de investigación que se utilizaran, el enfoque, la modalidad de la investigación utilizada, el tipo de investigación realizada.

Capítulo IV. “Desarrollo de la Propuesta”, en este capítulo se describe todo el desarrollo de la solución, definiendo los requisitos necesarios, los criterios que se aplicaron dando como resultado el plan estratégico.

Capítulo V. “Conclusiones y Recomendaciones” estableciendo las conclusiones a las que llega el investigador luego del desarrollo del proyecto, así también las recomendaciones pertinentes.

Finalmente se incluye las referencias citadas en este documento, en los anexos se incluye los instrumentos utilizados para la recolección de la información correspondientes del presente proyecto.

CAPÍTULO 1

El Problema

1.1. Tema

Sistema de información para la gestión de contenido académico publicado en redes sociales para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.2. Planteamiento del problema

Es en la Web 2.0 que se encuentra vinculación de las aplicaciones y plataformas presentes en la Internet que facilitan el intercambio de información y colaboración entre distintos individuos. Su desarrollo ha traído consigo un impacto en la sociedad, especialmente en los jóvenes. Este sector de la sociedad tiene diferentes preferencias y necesidades de información; priorizan el valor del contenido que se encuentra en las plataformas y aplicaciones siendo este uno de los factores en el que incide una mayor utilización y acceso. Actualmente, las aplicaciones y herramientas web de preferencia para el segmento de los jóvenes no han sido utilizadas de forma amplia en el sector universitario, limitándose la academia en gran medida al uso de blogs y de wikis especializadas [8].

Es en la Web 2.0, que el concepto de redes sociales adquiere relevancia. Las redes sociales se podrían definir, de cierta forma, como las asociaciones de diferentes personas unidas por motivos heterogéneos; constituyen una estructura unida por nodos interconectados entre ellos por más de un tipo de relación [9]. Referente a esto, las aplicaciones web de redes sociales que existen hoy en día concretan esta definición general permitiendo que personas con diversos intereses se comuniquen a través de la red. Los servicios que se ofrecen por este tipo de sistemas incluyen, entre otros, la creación de subgrupos y la mensajería pública o privada [10][11][12].

El origen de estas aplicaciones web es diverso, con diferentes planteamientos y enfoques: algunas, como Facebook, nacieron para unir e interactuar a distintos grupos de alumnos en los campus universitarios norteamericanos, mientras que otras, como Moodle, surgen ante la necesidad de entornos de aprendizaje virtuales. Por su parte, el enfoque profesional prima en el diseño de la red social LinkedIn, orientada a establecer contactos de carácter laboral y comercial me-

diante el denominado «networking» [11][13][12][14].

Las redes sociales, en un ámbito mayor de consideración, juegan un rol fundamental en la implementación de metodologías activas y participativas de educación, especialmente con el trabajo colaborativo. Se entiende por trabajo colaborativo al intercambio y desarrollo de conocimiento por parte de grupos reducidos de iguales, con la meta de alcanzar objetivos académicos similares. También pueden funcionar como plataformas de consolidación de comunidades de aprendizaje [15].

Se involucran a las redes sociales al consolidar comunidades de aprendizaje disponiendo potencialidades que son plenamente útiles si se dispusieran de una manera conjunta, se destacan las siguientes: se aumenta los niveles de motivación; beneficia en un aumento de los niveles de rendimiento académico dado que el aprendizaje del individuo y del grupo consta de una retro-alimentación continua, fomenta la retención del contenido aprendido; ayuda a potenciar el pensamiento crítico; aumenta la diversidad de conocimientos y experiencias adquiridas [16].

Estudios previos evidencian que el uso de las redes sociales con fin académico es limitado, lo cual significa la no explotación en el ámbito académico de un recurso que es de amplia utilización y de frecuente uso entre los estudiantes universitarios [17]. A pesar de la opinión generalizada de que las potencialidades de su explotación pedagógica son amplias y positivas [17][18], el aprovechamiento académico de esos recursos se ha reflejado solo en experimentos aislados en la educación universitaria [10][19].

La cuestión a resolver entre la academia y las redes sociales es la recolección y utilización de la información generada a partir de la interacción entre los individuos. Las redes sociales son medios por los cuales docentes y estudiantes forman comunidades de aprendizaje determinadas en base a correlaciones de estudio; es el factor de involucramiento, la relación del conocimiento respecto a los individuos lo que no se logra determinar aún de una forma efectiva. Las diversas redes sociales proveen de procesos por los cuales se obtiene información relevante para el individuo, pero aún no se contempla los aspectos relacionados a las comunidades de aprendizaje como la información de los contenidos difundidos por un individuo, la categorización de respecto a una temática, la recomendación de contenidos referente al resultado de los procesos de interacción.

Se considera entonces que si se contara con herramientas con el enfoque para la

gestión de actividades académicas en las redes sociales, que respeten las actividades de índole personal, sería posible potenciar el uso de las redes sociales en el ámbito académico y al mismo tiempo cuantificar el uso de las redes sociales con fin académico para estimar el valor real de las redes sociales en medio de comunidades de aprendizaje.

1.3. Delimitación

Área Académica:

Software.

Línea de Investigación:

Ingeniería de Software

Sublínea de Investigación:

Ingeniería del Software Orientada a la Web.

Delimitación Espacial:

La presente propuesta de investigación se llevará a cabo en la carrera de Sistemas Computacionales de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

Delimitación Temporal:

La realización del proyecto está planificada para un período de 6 meses de desarrollo contemplados a partir de la fecha de aprobación del perfil.

1.4. Justificación

El valor del conocimiento y su difusión e interacción cumple un papel importante en los procesos de enseñanza, ante lo cual se han realizado múltiples investigaciones de la web social en los ámbitos académicos.[20]. Se ha logrado a través de la creación y mantenimiento de blogs, de mecanismos para comentar publicaciones en la web, wikies, publicación de presentaciones de conferencias, vídeos instructivos, entre otros medios integra las nuevas tecnologías como parte del apoyo a la educación [17] [20].

La educación en el siglo XXI enfrenta el reto de adaptarse a las condiciones del mundo tecnológico en que se vive, natural para los estudiantes, y que extranjeriza a los docentes de la academia clásica [20]. La brecha generacional existente hace imperativo que los docentes adquieran habilidades en el uso y manipulación de tales herramientas para adaptarse a estos nuevos entornos [21].

La presente investigación propone el desarrollo de un sistema de información para la gestión de la «vida social» en el ámbito académico. Permitiendo la cuantificación del uso de las redes sociales para la colaboración en el fortalecimiento de los procesos de enseñanza/aprendizaje. Se pretende que el sistema provea una retro alimentación activa, permitiendo evaluar la calidad y pertinencia de los contenidos académicos distribuidos a través de las redes sociales. Además, el uso de la herramienta propuesta ha de fortalecer la capacidad de colaboración entre los estudiantes y docentes del entorno universitario. Además de aportarle un componente lúdico al modelo educativo universitario, este proyecto de investigación permitirá analizar el proceso académico y generar influencia en el uso de los medios sociales con fin académico.

1.5. Objetivos

1.5.1. General

Implementar un sistema de información que permita la gestión del contenido académico publicado en redes sociales por parte de los docentes y estudiantes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.5.2. Específicos

- Analizar los requerimientos del sistema de información para la gestión de contenido académico publicado en redes sociales por estudiantes y docentes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.
- Determinar el alcance del sistema de información para la gestión de contenido académico en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.
- Diseñar la arquitectura del sistema de información para la gestión de contenido académico en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial basada en el modelo de datos de la red social Facebook.

- Desarrollar el sistema de información para la gestión de contenido académico en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes Investigativos

Entre los desarrollos más recientes en la creación y comunicación de información en formato electrónico, la web social es un término que se refiere al conjunto de tecnologías web que permiten la creación e intercambio de contenido generado por los usuarios [17].

Castells define en «The Rise of the Network Society» el proceso por el cual transcurre la información en la sociedad contemporánea. Manuel Castells describe el paso de una sociedad industrial a una sociedad de la información, que se inició en la década de 1970. La sociedad está estructurándose en torno a redes en lugar de actores individuales, y funciona a través de un flujo constante de información a través de la tecnología. Castells destaca la interrelación de las características sociales, económicas y políticas de la sociedad, y argumenta que la «red» es la característica definitoria que marca nuestra época actual.[22]

En el estudio experimental de Tsovaltzi, realizando de conjunto con investigadores de la Carnegie Mellon University, en los EE.UU, se analizó la influencia de Facebook en el entorno de aprendizaje colaborativo [23]. Posteriormente, estudiaron las capacidades de discusión de Facebook para el aprendizaje [24]. Por su parte, en Holanda se ha conducido un estudio profundo del uso de Facebook como facilitador de procesos de enseñanza/aprendizaje y su implicación como herramienta de discusión [25].

Kirschner defiende el criterio de que los docentes pueden utilizar de una manera diferente y a su vez efectiva las redes sociales para la discusión de temas de tecnologías en procesos de enseñanza/aprendizaje para jóvenes y adolescentes. En su estudio, Kirschner concluye que el trabajo de investigación en áreas referentes a las redes sociales para disminuir la brecha entre el aprendizaje formal e informal es extenso. [25]

Por su parte, se han realizado investigaciones por la University of Kang Ning Tainan, Taiwan, en las que se han centrado esfuerzos en el estudio de las capacidades de interacción de los estudiantes en comunidades creadas con el API de

Facebook [18]. Chen resalta la importancia de la capacitación de los docentes por cuanto, según el estudio que realizaron, los estudiantes prestan mayor atención y son más reactivos ante las publicaciones de los docentes que a las de compañeros de estudio.

En el área latinoamericana, Cabero y Marín definieron un instrumento que les permitió corroborar la predisposición de los estudiantes del área de ciencias sociales al trabajo en grupo haciendo uso de las redes sociales [19]. Desde un poco antes, se identificó la necesidad en el área latinoamericana de institucionalizar lógicas participativas y las dinámicas basadas en las comunidades que se desarrollan en los entornos de las redes sociales. [10].

2.2. Fundamentación teórica

2.2.1. Web 2.0

El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual [26].

La Web 2.0 es un concepto que se acuñó en 2003 y que se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet. El término establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para que se generara la interacción) y la revolución que supuso el auge de los blogs, las redes sociales y otras herramientas relacionadas [27].

En conclusión, la gran mayoría de aplicaciones y plataformas actuales se encuentran en la Web 2.0 permitiendo realizar el intercambio de información y la colaboración entre los usuarios [26].

La Web 3.0 constituye el futuro de la Internet, que busca dar soporte al avance de la sociedad basado en la gestión inteligente de la información disponible. Constituye un nuevo arquetipo de Web en la que se adiciona contenido semántico a los documentos que la conforman y ello conlleva que el procesamiento de la misma sea realizada por agentes de software. La implementación de la Web 3.0

por parte de las plataformas y sistemas informáticos de la Internet actual es aún una problemática que se resuelve de forma progresiva considerándose el desarrollo y uso de ciertos aspectos específicos de la Web 3.0 como son la segmentación de datos y la generación de perfiles de usuarios. [28][29].

2.2.2. Red Social

Una red social es una estructura social compuesta por un conjunto de actores (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio o relación social como profesión, amistad, parentesco, entre otras. Normalmente se representan simbolizando los actores como nodos y las relaciones como líneas que los unen [27].

Las investigaciones han mostrado que las redes sociales constituyen representaciones útiles en muchos niveles, desde las relaciones de parentesco hasta las relaciones de organizaciones a nivel estatal (se habla en este caso de redes políticas), desempeñando un papel crítico en la determinación de la agenda política y el grado en el cual los individuos o las organizaciones alcanzan sus objetivos o reciben influencias. La red social también puede ser utilizada para medir el capital social (es decir, el valor que un individuo obtiene de los recursos accesibles a través de su red social) [30].

2.2.3. Sistema de Información

Un sistema de información se puede definir técnicamente como la agrupación de componentes relacionados que recolectan, procesan, almacenan y distribuyen información para apoyar en la toma de decisiones y dar guía en el control de una organización en determinados aspectos [31].

Según Claudio Ciborra, el estudio de los sistemas de información se define como el estudio de la inserción y el uso de la tecnología de la información en las organizaciones, instituciones, y en general en la sociedad [32].

Los sistemas de información se enfocan en el desarrollo, uso y administración de la infraestructura de las tecnologías de la información en una organización [33].

2.2.4. Base de Datos

Según Henry F. Korth autor del libro «Fundamentos de Bases de Datos», se define una base de datos como una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar esos datos.

Uno de los propósitos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de como se almacenan y mantienen los datos [34].

2.2.5. Aplicación Web

Según Rafael Barzanallana, de la Universidad de Murcia: «desde la perspectiva de un usuario, puede ser difícil percibir la diferencia entre un sitio web y una aplicación web» [35]. Según el Diccionario Oxford en línea, una aplicación es «un programa o conjunto de programas para ayudar al usuario de un ordenador para procesar una tarea específica» [36].

Una aplicación web es básicamente una manera de facilitar el logro de una tarea específica en la web, a diferencia de un sitio web estático que es más bien una herramienta, no siendo menos importante, para la comunicación con los diversos usuarios.

El término más decisivo de esta definición es acorde a la ejecución de una «tarea específica». Definiendo por lo tanto que una aplicación web es el medio por el cual se permite la interacción directamente entre usuarios y datos. [37].

2.2.6. Servicio Web

Un servicio web es un servicio ofrecido por un dispositivo electrónico a otro dispositivo electrónico, que se comunican entre sí a través de la web (World Wide Web). En un servicio web, la tecnología web, como el protocolo HTTP, diseñado originalmente para la comunicación de persona a máquina, se utiliza para la comunicación de máquina a máquina, más específicamente para máquinas traslación formatos legibles como XML y JSON [38].

El servicio web normalmente proporciona una interfaz basada en web orientado a objetos que se comunica con un servidor de base de datos, utilizado en su caso otro servicio web, o mediante una aplicación, que proporciona una interfaz de usuario para el usuario final [39].

El W3C define un servicio web en general como «un sistema de software diseñado para apoyar la interacción de máquina a máquina en una red» [38][39].

2.2.7. Modelo - Vista - Controlador

Es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con los usuarios. Este patrón de diseño organiza la aplicación en tres modelos separados. El primero es un modelo que representa los datos de la aplicación y sus reglas de negocio; el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información; el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema [40].

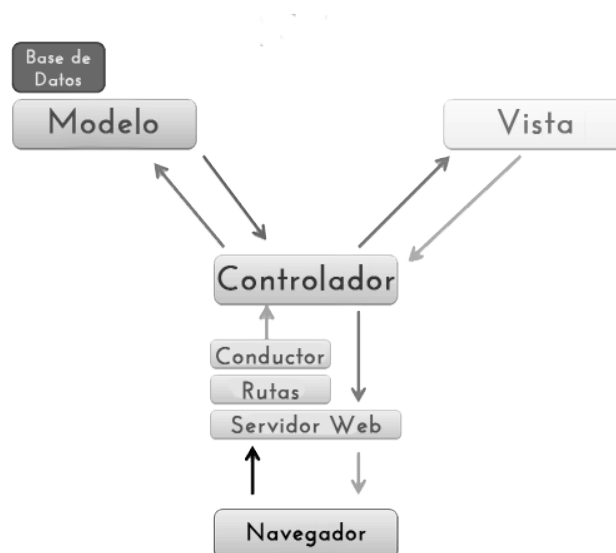


Figura 1: Modelo-Vista-Controlador [1]

2.2.8. Modelo - Vista - Vista Modelo

Es un patrón arquitectónico basado en el patrón Modelo Vista Controlador (MVC). MVVM ayuda al desarrollo de una lógica de negocio independiente (modelo) de la interfaz de usuario (vista) utilizando el ViewModel(vista-modelo) [41].

MVVM tiene como objetivo que los datos que utiliza la vista se puedan presentar y administrar de la manera más óptima. Es la vista-modelo quien expone los datos desde el modelo ,en este sentido, la vista-modelo es más un modelo que una vista. Adicionalmente gestiona la lógica de visualización de la vista por lo que esta, a partir del punto de la vista, es más una vista que un modelo. Aún al día

de hoy la mezcla de las responsabilidades y relaciones en MVVM es un tema de conversación y exploración en el sector [2].

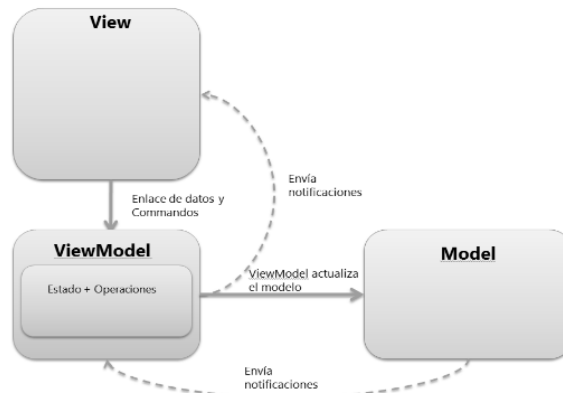


Figura 2: Modelo - Vista - Vista Modelo [2]

2.2.9. API

Los servicios web son creados específicamente para proveer información a una aplicación o sitio web frente a una necesidad. Los programas clientes usan «interfaces de programas de aplicaciones» (API) para comunicarse con los servicios web. Generalmente una API expone un conjunto de datos y funciones para facilitar las interacciones entre los programas y permitir el intercambio de información [42].

2.2.9.1. REST

La arquitectura REST es comúnmente aplicada al diseño de API modernas de servicios web. El término REST (REpresentational State Transfer) que traducido vendría a ser «transferencia de representación de estado» es un conjunto de principios para construir aplicaciones web respetando HTTP fue definido por el Phd Roy T. Fielding en el año 2000 [3].

Tener una implementación REST API permite a un servicio web ser «RESTful». Servicios web con REST API se comunican sobre HTTP (Hypertext Transfer Protocol), esta comunicación es disponible con comandos del protocolo HTTP (GET, POST, PUT, DELETE) que los clientes web indistintamente de su plataforma recuperan y envían información a servidores remotos [42][3].



Figura 3: Servicio Web Rest API [3]

2.2.10. Facebook

Es un sitio web de redes sociales lanzada en 2004 y fue fundado por Mark Zuckerberg, estudiante de la Universidad de Harvard. Inicialmente, el acceso a Facebook se limitaba solamente a los estudiantes universitarios de Estados Unidos, con el tiempo se ha ido expandiendo de un segmento específico de la población hacia ser conocido y usado por el público en general alrededor del mundo.[11][43]

Facebook es gratuito para los usuarios y genera ingresos por la publicidad expuesta, incluyendo los banners y los grupos patrocinados. Los usuarios pueden registrarse a través de su correo electrónico y pueden hacerlo como celebridades, músicos o grupos de música, negocios o empresas, o personas individuales. Pueden crear perfiles que contienen fotos, listas de intereses personales e información privada o no, y pueden realizar un intercambio de mensajes privados y públicos entre sí y en los grupos de amigos. La visualización de los datos detallados de los miembros está restringido a los miembros de la misma red, a los amigos confirmados, o puede ser libre para cualquier persona [11].

2.2.10.1. Graph API

Esta es la principal forma de obtener datos gestionados por la plataforma de Facebook. Es una API basada en HTTP de bajo nivel que se puede utilizar para consultar datos, publicar contenido, administrar anuncios, subir fotos y una variedad de otras tareas que una aplicación podría necesitar.[44]

Se consulta en base a un clave de autorización que se le provee a los usuarios que solicitan información de Facebook [44].

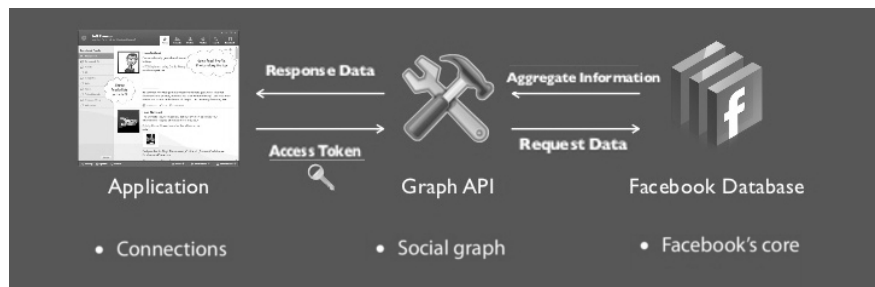


Figura 4: Modelo de datos del API de Facebook [4]

2.2.11. Node.js

Es un marco de trabajo multiplataforma de código abierto de desarrollo y entorno de tiempo de ejecución para aplicaciones web escalables y de desarrollo ágil. Este se basa en el motor Google V8 Javascript y en el lenguaje de programación ECMAScript, con una arquitectura orientada a eventos. Implementa una ejecución del código en el servidor a diferencia de la mayoría del código JavaScript. [45][46][47]

2.2.12. Sails.js

Sails.js permite el desarrollo de sistemas y aplicaciones a medida, a un nivel empresarial en el entorno de Node.js. Su desarrollo constituye un marco de trabajo diseñado para imitar los patrones de otros marcos de trabajo como Ruby on Rails. Debido a que utiliza la biblioteca Socket.io en su infraestructura, permite la comunicación en tiempo real [48].

En la estructura de Sails se utiliza un conjunto de marcos de desarrollo como Express.js que añade extensible de manejo de servidores HTTP , ORM Waterline que permite la simplificación de la capa de datos al posibilitar intercambiar adaptadores para la mayoría de las bases de datos SQL y no SQL. Una de las características resaltables es la generación automática de entornos API REST. Siendo esto de utilidad dado que permite crear APIs de simple acceso con un diseño sólido. Por último, es de destacar su compatibilidad adicional con los diversas herramientas y bibliotecas del entorno de Node.js [48][49].

2.2.13. Bootstrap

Bootstrap es un marco de desarrollo de código abierto desarrollado por el equipo de Twitter. Es una combinación de HTML, CSS, y Javascript cuyo fin es ayudar a la construcción y diseño de interfaces y componentes adaptativos. Bootstrap fue programado para soportar CSS3. También es denominado como un marco de desarrollo para la programación en el cliente [50][51].

Algunos de los componentes que se encuentran ya construidos y disponibles en Bootstrap son:

- Contenedores
- Menús desplegados
- Botones y grupos de botones

- Barra de Navegación
- Barra de progreso
- Etiquetas
- Alertas
- Barras de progreso
- Ventanas modales

2.2.14. AngularJS

AngularJS es un marco de desarrollo de JavaScript desarrollado y mantenido por Google y la comunidad [41] [52]. AngularJS incorpora los principios básicos del patrón de desarrollo MVC (Modelo-Vista-Controllador) para el desarrollo de aplicaciones web. La forma de cómo AngularJS aplica el patrón MVC difiere del sentido tradicional siendo que más bien se acerca al concepto de MVVM (Modelo-Vista-Vista Modelo).[41][53]

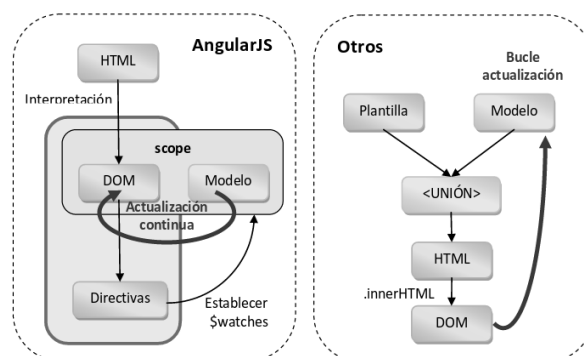


Figura 5: Arquitectura de los componentes de AngularJS [53]

Angular es utilizado para el desarrollo de la programación en el cliente (front-end); es decir que se encarga de la interacción con el usuario final. Una funcionalidad importante que posee es el desarrollo de las Single Page Application (SPA) siendo esto el desarrollo de aplicaciones en una sola página web, separa, dentro de la arquitectura, la responsabilidad de cada tecnología en su ámbito (CSS, HTML, JavaScript) y las sincroniza cuando es necesario.[54]

La filosofía de AngularJS es el trabajo de modo agnóstico; esto es que la modificación del HTML se realiza sin modificar el DOM. Como resultado se logra un código fuente más limpio y mantenible.[41]

2.2.14.1. Arquitectura de AngularJS

- **Vista:** La vista en una aplicación en realidad es una pieza que es representada en un navegador a través del usuario puede interactuar o ver cualquier dato que haya sido solicitado. En una aplicación AngularJS, la vista esta compuesto por directivas, filtros y data-bindings (enlaces de datos). Para la mantenibilidad y facilidad en la creación de pruebas no se incluye todo el código en la vista.
- **Controlador:** El controlador tiene toda la lógica de la aplicación en AngularJS; controla y prepara los datos en los formularios para que sean debidamente presentados en la vista. El controlador es el responsable de recuperar los datos desde el servidor (utilizando peticiones Ajax) y de enviar los datos al servidor desde la vista.
- **Modelo/Vista Modelo/\$Scope:** Es la parte principal de la arquitectura de AngularJS denominado modelo, vista-modelo(view model) o \$scope. \$Scope es un término que es introducido en AngularJS, responsable de detectar los cambios en el modelo; proporciona un contexto a la vista en su función.

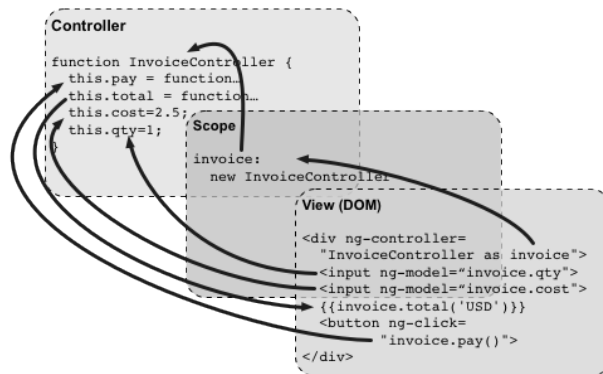


Figura 6: Arquitectura de los componentes de AngularJS [55]

2.2.15. Usabilidad

La palabra «usabilidad» se refiere a métodos para mejorar la facilidad de uso durante el proceso de diseño [56]. En la usabilidad se definen componentes de calidad como son:

- La facilidad de aprendizaje para que el usuario realice una tarea.
- La eficiencia después del tiempo de aprendizaje, relacionada con la rapidez con que el usuario puede realizar las tareas.

- La memoria, como el factor de cuando los usuarios recuerdan la concepción después de un tiempo sin utilizar una aplicación o sistema.
- La facilidad con la que pueden restablecer la conexión con las funciones del sistema.
- Los errores más frecuentes y la facilidad con la que los usuarios pueden recuperarse de estos

- La satisfacción en cuan agradable es la utilización del diseño.

2.2.16. Principios Heurísticos de la Usabilidad Web

Jakob Nielsen, enunció 10 principios para el diseño de la interacción con el usuario indistintamente del ambiente. Estos principios son denominados heurísticos a causa de que constituyen amplias reglas de directrices al momento de definir usabilidad en una aplicación web [57][56]. A continuación se analizan estos principios.

Visibilidad del estado del sistema

Siempre se debería mantener informados a los usuarios de lo que está ocurriendo, a través de la retroalimentación.

Relación entre el sistema y el mundo real

Es un valor importante hablar el lenguaje de los usuarios mediante palabras, frases y conceptos que les sean familiares ,más que con términos técnicos o relacionadas a un tema específico. Se debe hacer que la información aparezca en un orden natural y lógico, siguiendo las convecciones del mundo real.

Control y libertad del usuario

Los usuarios a menudo escogen funciones por error y necesitarán de una «salida de emergencia» para dejar el estado no deseado y regresar al estado previo. Se deben soportar las funciones de deshacer y rehacer.

Consistencia y estándares

Los usuarios no deberían cuestionarse si las diferentes palabras, situaciones o acciones significan una misma cosa. Se debe priorizar seguir convenciones ya establecidas.

Prevención de errores

En vez de un elegante diseño de mensajes de error es preferible un diseño adecuado que evite la aparición de problemas.

Reconocimiento antes que recuerdo

Se deben hacer visibles los objetos, acciones y opciones, El usuario no tendría que recordar la información que se le da en una parte del proceso, para seguir adelante. Las instrucciones para el uso del sistema deben estar a la vista o ser fácilmente recuperables cuando sea necesario.

Flexibilidad y eficiencia de uso

La presencia de aceleradores, funciones que no son vistas por los usuarios novatos. A menudo pueden acelerar la interacción para el usuario experto de tal manera que el sistema puede servir tanto a los usuarios inexpertos y experimentados. Se debe permitir a que los usuarios adapten la aplicación para sus usos frecuentes.

Estética y diseño minimalista

Los diálogos no deben contener información irrelevante o que pocas veces necesitadas, ya que esto disminuirá la visibilidad y efectividad de la información. Cada unidad extra de información en un diálogo, compite con las unidades de información relevante y disminuye su visibilidad relativa.

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Los mensajes de error deben ser expresados en lenguaje claro, indicando precisamente el problema y, de una manera constructiva, sugerir una solución. Se deben redactar en un lenguaje claro y simple.

Ayuda y documentación

Inclusive en los casos en que se podría usar un sistema, aplicación o plataforma sin documentación, es necesario ofrecer ayuda a los usuarios. La información debería ser fácil de buscar y enfocada en las tareas del usuario.

2.3. Propuesta de Solución

El presente proyecto propone un sistema de información basado en el modelo de datos de Facebook para la gestión centralizada del contenido académico publicado en la web por docentes y estudiantes, facilitando el compartir y la colaboración académica en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

CAPÍTULO 3

Metodología

3.1. Modalidad de la investigación

Con la finalidad de obtener la información mas fidedigna del sistema de información para la gestión de contenido académico, se realizó la investigación de campo donde se obtuvo la información necesaria para definir la propuesta en cuestión.

Se realiza también una investigación bibliográfica para la elaboración del marco teórico, en donde se ha establecido el conjunto de conocimientos en los que se sustenta la investigación, además de presentar investigaciones previas que sirven de soporte a la investigación.

3.2. Recolección de información

Se recolectó la información primaria mediante el método de observación para conocer la problemática que ocurre y determinar los procesos a mejorarse. También se aplicó encuestas a los grupos de usuarios potenciales del sistema para conocer la implicación de las redes sociales en la academia.

La información secundaria se recolecto de documentación técnica, investigaciones, artículos científicos que se han desarrollado en cuanto al funcionamiento e interacción de los sistemas con los usuarios y el rol de las redes sociales en los ámbitos académicos.

3.3. Procesamiento y análisis de datos

Para la recolección, procesamiento y análisis de la información se aplicará los siguientes procedimientos:

- Elaboración del instrumento de recolección de datos
- Aplicación de instrumento de recolección de información.
- Análisis de la información.
- Resultados.

3.4. Desarrollo del proyecto

- Análisis de requerimientos del sistema de información
 - Levantamiento de requerimientos
 - Realización de entrevistas
 - Realización de entrevistas
 - Realización de observación de procesos
- Determinación del alcance del sistema
 - Definición ámbitos del sistema
- Diseño de arquitectura del sistema de información
 - Diseño de modelo lógico de base de datos
 - Diseño de modelo físico de base de datos
 - Elaboración diagrama de casos de uso
 - Comportamiento dinámico de los procesos del sistema
- Desarrollo del sistema de información
 - Especificación de casos de uso
 - Producción de software
 - Arquitectura
 - Requisitos funcionales
 - ◇ Elaboración diagrama de secuencias
 - ◇ Elaboración de diagrama de clases
 - ◇ Detalle de funcionalidades
 - Requisitos no funcionales
 - ◇ Diseño
 - ◇ Seguridad
 - ◇ Mantenibilidad
 - Transición
 - Pruebas de funcionalidad
 - Puesta de producción
 - Estabilización

CAPÍTULO 4

Desarrollo de la propuesta

Para concretar el desarrollo de la propuesta, se tomó como guía los objetivos específicos planteados; por cada objetivo específico se realizaron pasos concretos que se detallan a continuación:

4.1. Análisis de requerimientos del sistema de información

4.1.1. Levantamiento de requerimientos

Por parte de la Ing. Pilar Urrutia Urrutia, MSc.y del Dr. Félix Oscar Fernández Peña, coordinadora y coordinador subrogante, respectivamente, del proyecto “Sistema de Gestión Académica en las Redes Sociales” (SOMAN) se obtiene información inicial , accediendo a documentos científicos en lo referente a la intervención de las redes sociales en los ámbitos académicos y el valor del contenido respecto a las tecnologías de la información.

A partir de los registros de peticiones al servidor DNS, proveídos por la Dirección de Tecnología de Información y Comunicación (DITIC) de la Universidad Técnica de Ambato, se realizó un estudio del tráfico web, cuyos resultados se encuentran descritos en el **Anexo A**, en el que se analiza el número de peticiones que se han realizado , sitios con mayor tráfico y los porcentajes de predominancia en comparación a otros sitios y al tráfico total. Adicionalmente, se llevó a cabo un análisis de los resultados de la encuesta aplicada referente a la relación entre Facebook y la Academia (**Anexo B**), la cual fue aplicada a un conjunto de 300 personas tanto hombres como mujeres que disponían del perfil de ser participantes de ambientes académicos y cumplir roles como estudiantes, docentes, e investigadores. En base al **Anexo A**, en conjunto con la información dada por el proyecto «SOMAN» se realizó la guía de observación(**Anexo C**).

Como resultado, se obtiene el enfoque del proyecto hacia la red social Facebook, teniendo en cuenta la predominancia existente de Facebook, entre las redes sociales que se utilizan en el ambiente académico.

Mediante la guía de observación y la encuesta aplicada se analizaron los diferentes procesos que intervienen en la gestión y difusión de contenido, teniendo en cuen-

ta el punto de vista del usuario y de la academia. Así también se definieron los aspectos primordiales necesarios, a tomar en cuenta en el desarrollo del proyecto.

4.2. Determinación del alcance del sistema

Para la determinación de los ámbitos y alcance del sistema se elaboró un documento en el cual se especifican los requerimientos y funcionalidades, se utiliza el estándar de la norma IEEE 830 y el proyecto cumple con lo establecido en el documento de requerimientos. **Anexo D**

4.2.1. Definición de ámbitos del sistema

El presente proyecto de investigación se enfoca en la mejora del acceso a los contenidos académicos dentro de la plataforma de redes sociales Facebook para la FISEI, por lo cual se presentan a continuación los procesos que automatiza el sistema de información «Sigma».

- Gestionar el registro y acceso de usuarios en la plataforma de aplicaciones de Facebook.
- Registrar y visualizar los contenidos académicos que proveen los usuarios en el sistema.
- Realizar búsquedas de contenidos académicos registrados en el sistema.
- Visualizar los sitios web que hacen referencia los contenidos académicos.
- Registrar las acciones sociales realizadas por los usuarios en los contenidos.
- Proveer opciones para filtrar y ordenar contenidos.

La implementación de este proyecto permitió gestionar las acciones sociales generadas por los usuarios en Facebook y utilizar dicha información para valorar la relevancia de los contenidos, facilitando el consumo de información de valor académico en Facebook.

4.3. Diseño de arquitectura del sistema de información

4.3.1. Diseño de modelo lógico de base de datos

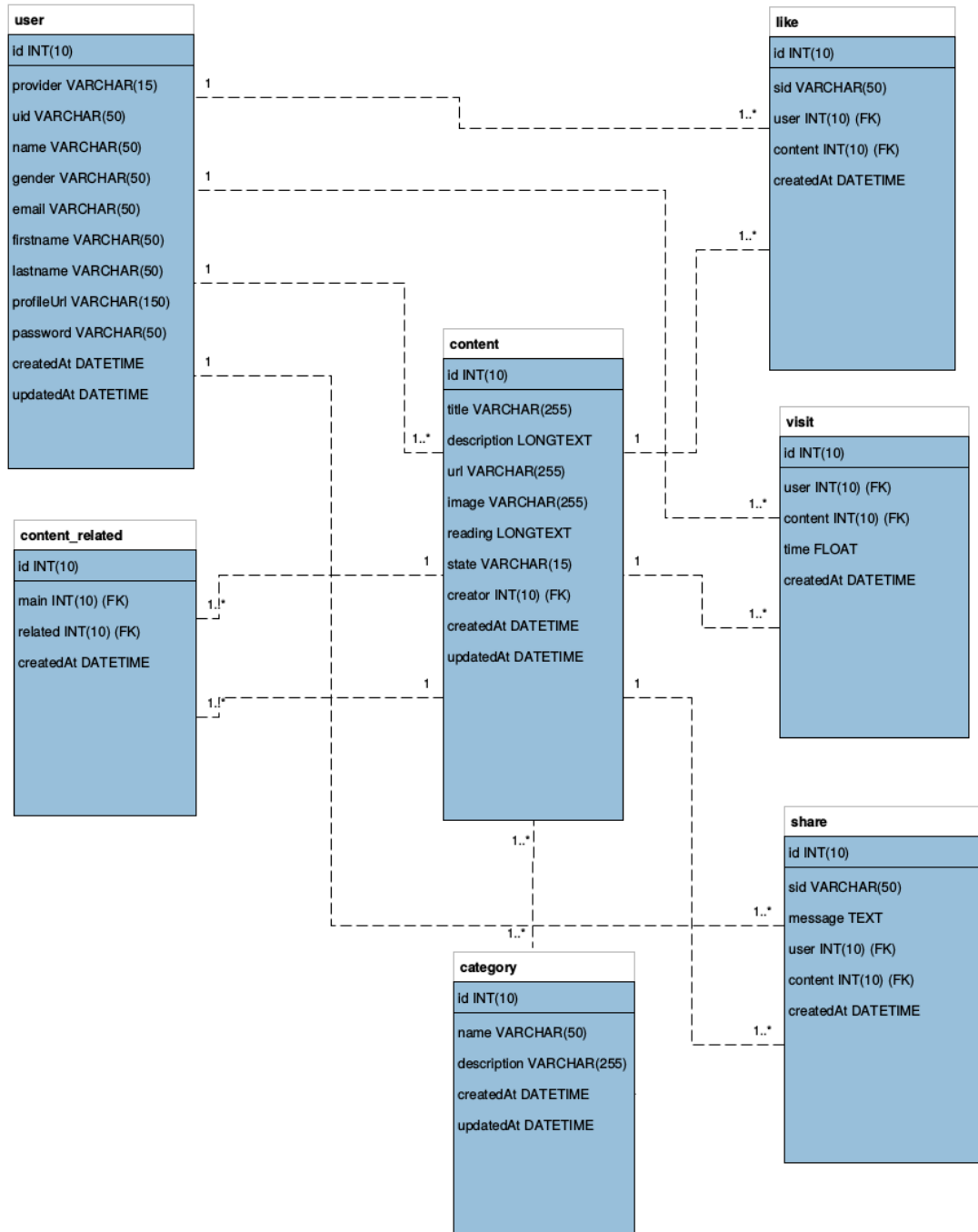


Figura 1: Modelo lógico de la base de datos

4.3.2. Diseño de modelo físico de base de datos

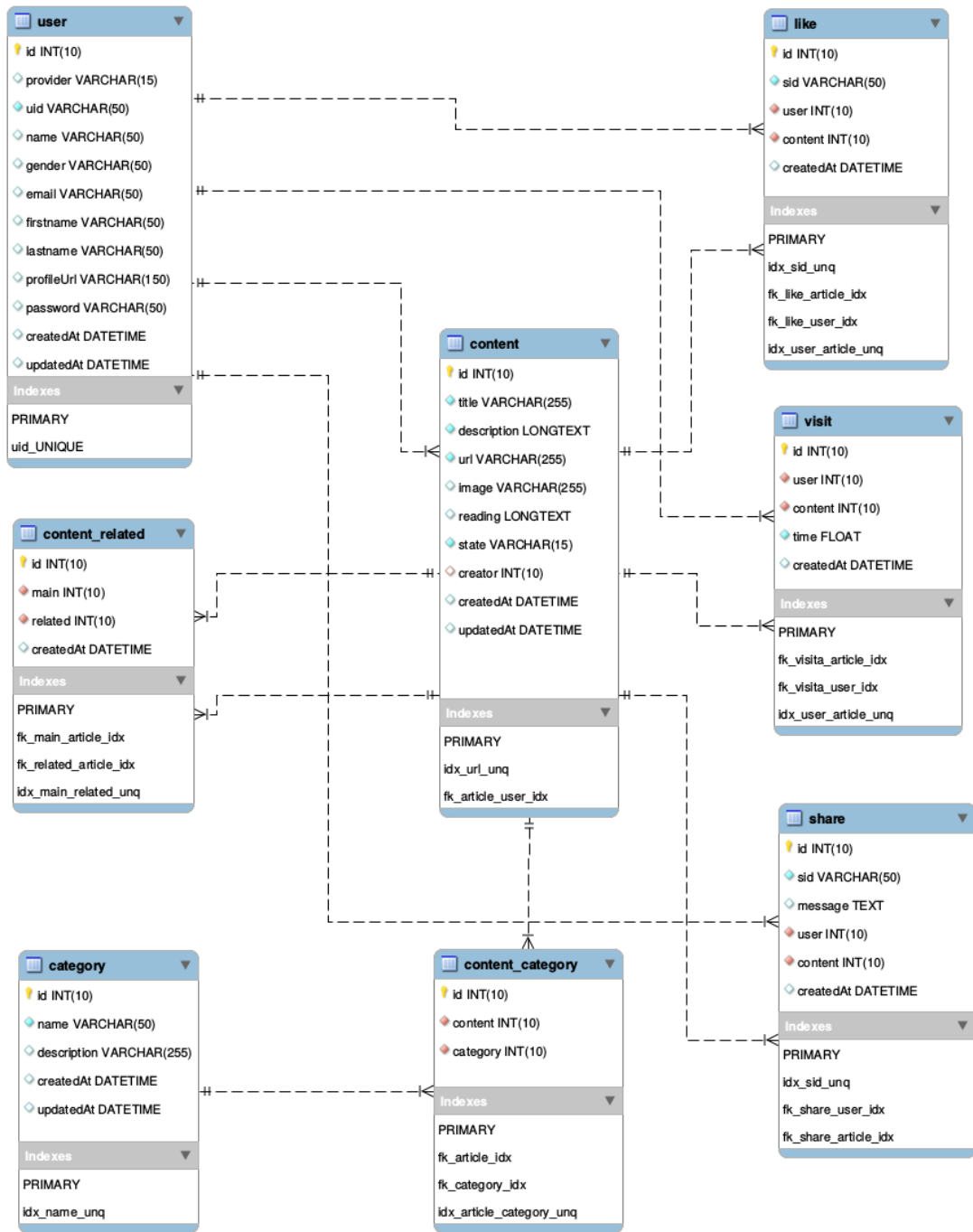


Figura 2: Modelo físico de la base de datos

Se analiza el diccionario de datos en el **Anexo E**

4.3.3. Elaboración diagrama de casos de uso

Los diagramas de casos de uso ayudan en la representación de cómo un usuario (actor) opera con el sistema, además de la forma, tipo y orden en cómo los elementos interactúan (operaciones o casos de uso) [58].

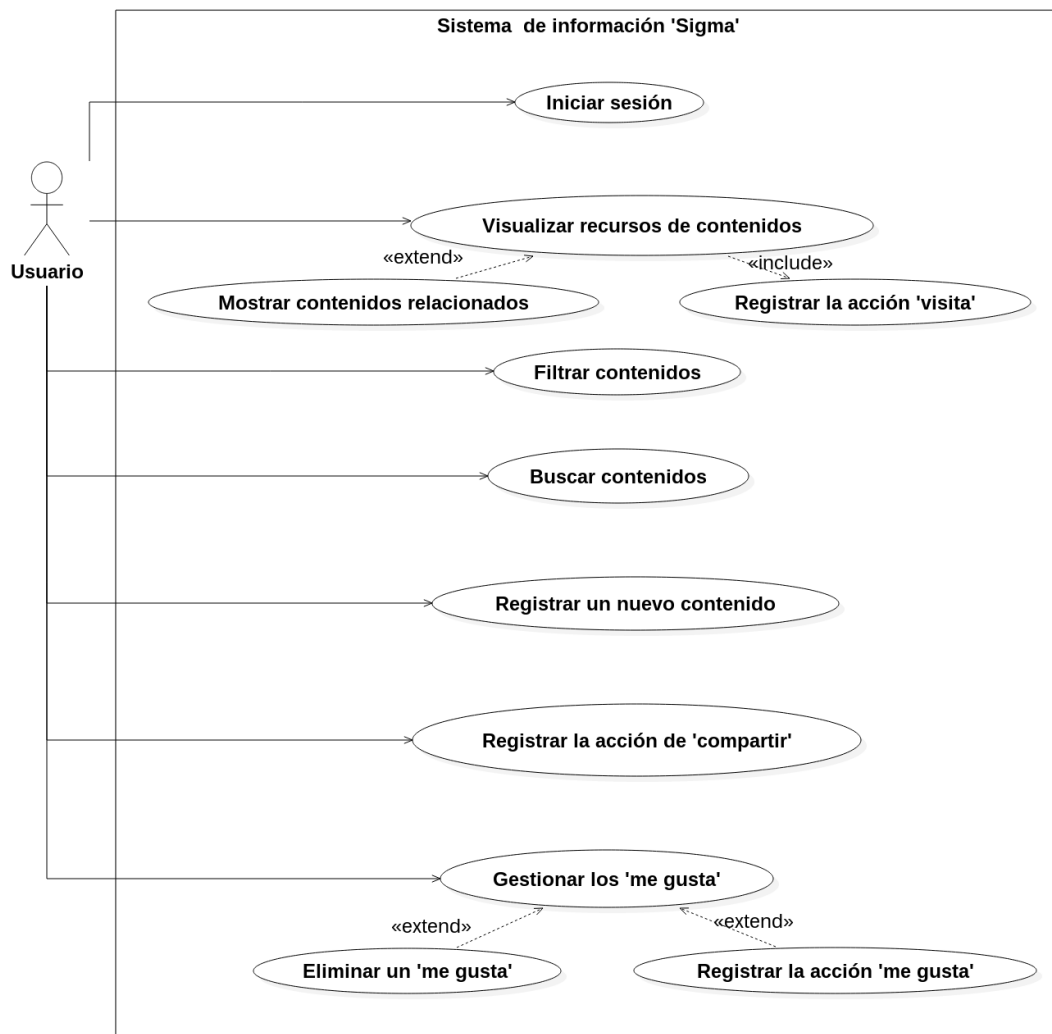


Figura 3: Diagrama de casos de usos del sistema

4.3.4. Comportamiento dinámico de los procesos del sistema

El comportamiento del sistema se lo describe a continuación utilizando los *diagramas de actividades*. Para explicar los flujos de acciones que el sistema implementa en su funcionalidad. [58][59]

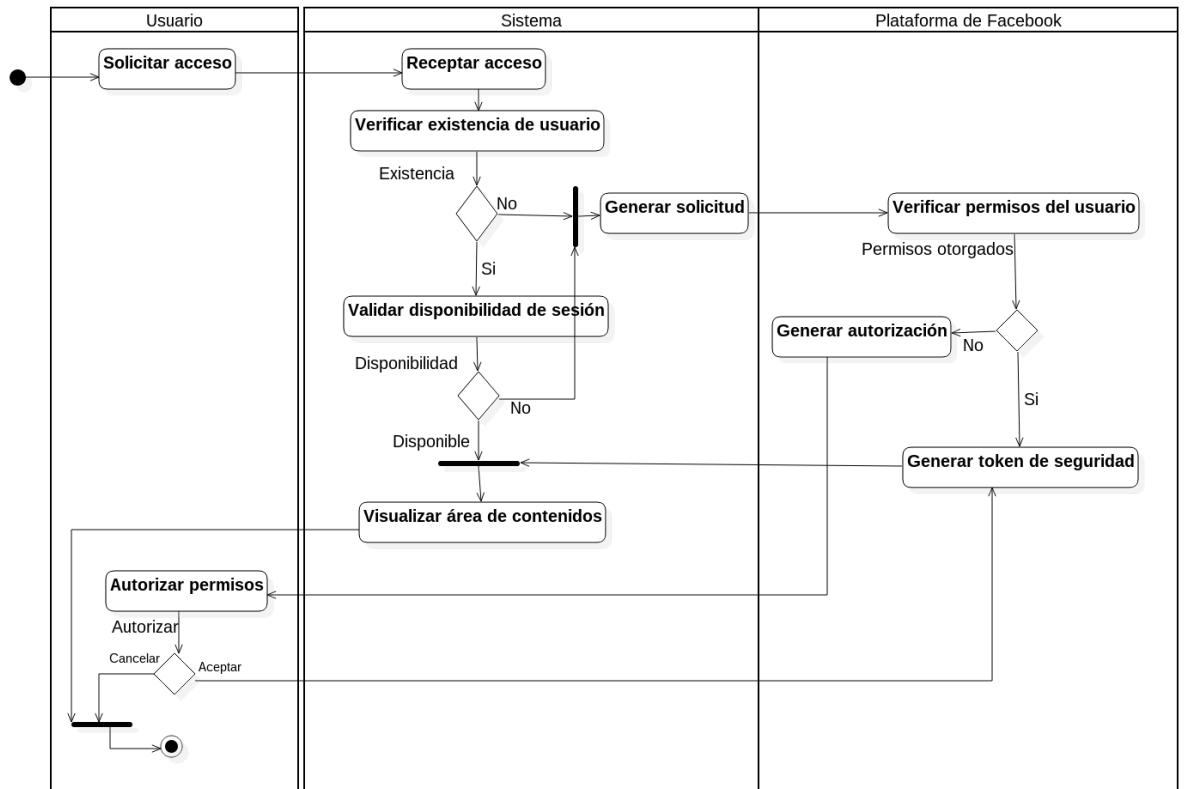


Figura 4: Diagrama de actividades. Iniciar sesión

El flujo para iniciar sesión es una interacción entre el usuario, el sistema y la plataforma de Facebook. Cuando el usuario desea iniciar sesión en el sistema se comprueba la existencia del usuario, la disponibilidad de la sesión es decir si cuenta con una sesión activa, y el fin del flujo es la visualización del área de contenidos.

Cuando no llegará a existir el usuario o la sesión del usuario ha caducado, el sistema genera una solicitud a Facebook para obtener el token de seguridad, si el usuario no ha otorgado permisos aún al sistema Sigma Facebook genera una ventana modal en la que el usuario decide aceptar o cancelar la autorización de permisos, si decide no hacerlo el flujo se termina.

En el flujo para la visualización de los recursos de un contenido, es la acción de solicitar al sistema mayor información de un contenido, es decir visualizar el sitio web del origen del contenido. El sistema procesa la petición, si un contenido tiene disponible el directorio de los recursos los visualiza y se termina el flujo, en caso de no haber aún los recursos, el sistema los descarga y los guarda en un directorio que es generado a partir de la dirección URL que hace referencia el contenido.

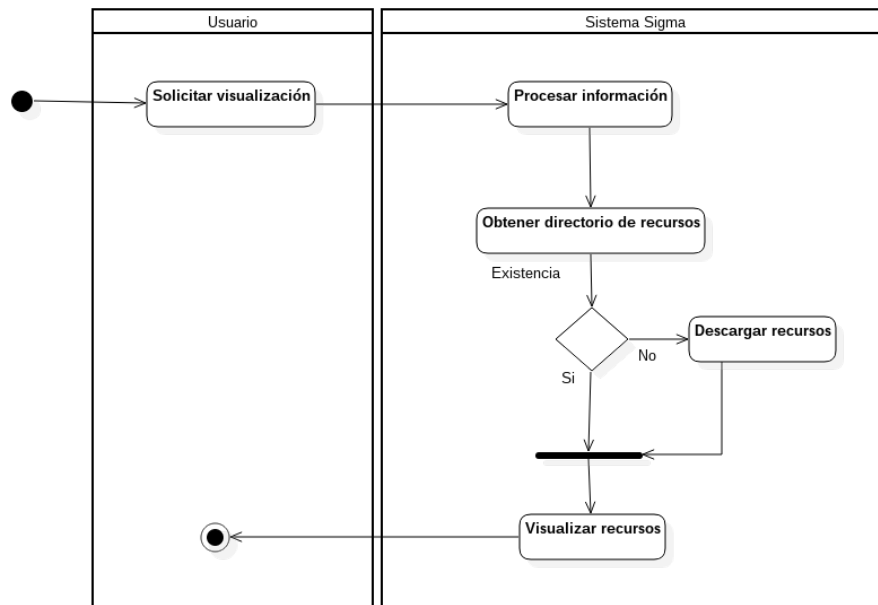


Figura 5: Diagrama de actividades. Visualizar recursos de contenido

La búsqueda de contenidos es un proceso en que el usuario envía parámetros para realizar una selección de contenidos, el sistema procesa los parámetros, realiza la búsqueda visualiza los contenidos que cumplan con los parámetros sino llegará a ver resultados muestra un mensaje de error al usuario.

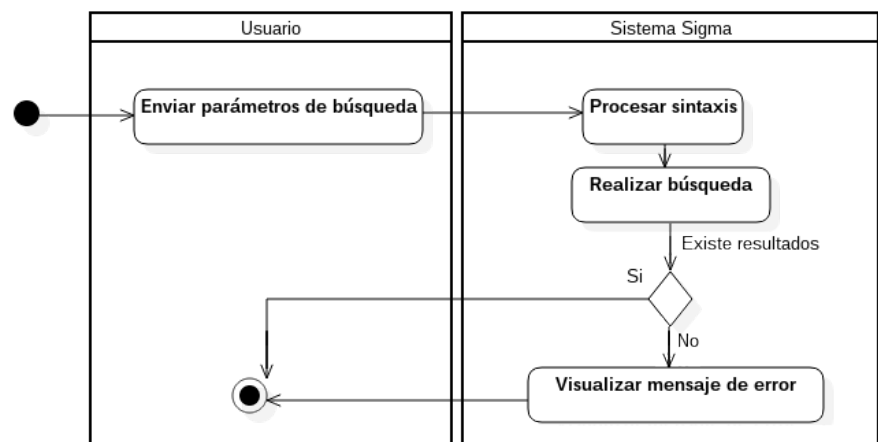


Figura 6: Diagrama de actividades. Buscar contenidos

El flujo de proceso de registrar una visita se lo realiza en cada petición de visualizar el contenido. Existe dos orígenes de la petición de visualización la que es dentro del sistema y la que es externa respecto a esta el tiempo que se le adjudica es el designado por defecto . Cuando el origen de la petición es para visualizar dentro del sistema se registra con tiempo desde la apertura del componente que se encarga de visualizar el contenido hasta el momento en que el usuario decidió salir del componente.

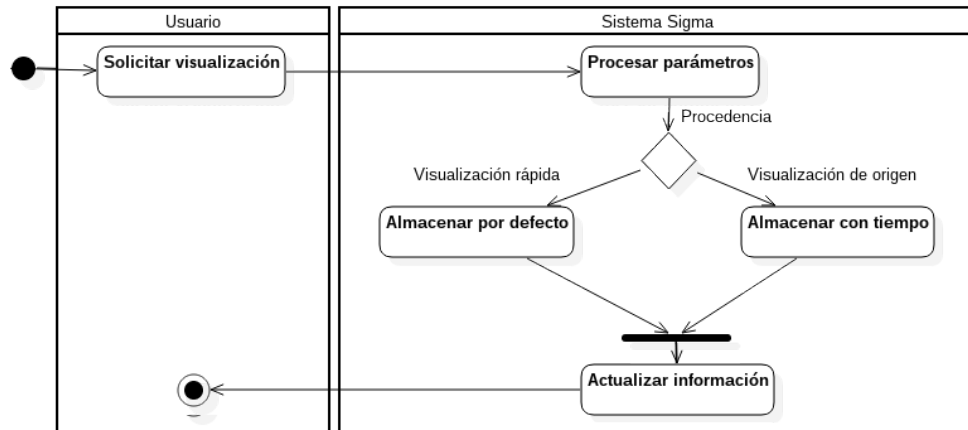


Figura 7: Diagrama de actividades. Registrar acción «visita»

Para registrar un nuevo contenido el usuario envía la información, el sistema la valida respecto a los tipos de dato, y que la dirección URL a la cual se hace referencia no se halla previamente registrado. Si fuera el caso de que la información enviada no es valida, el sistema permite al usuario editar la información. El flujo termina al confirmarse que sea ha creado un nuevo contenido

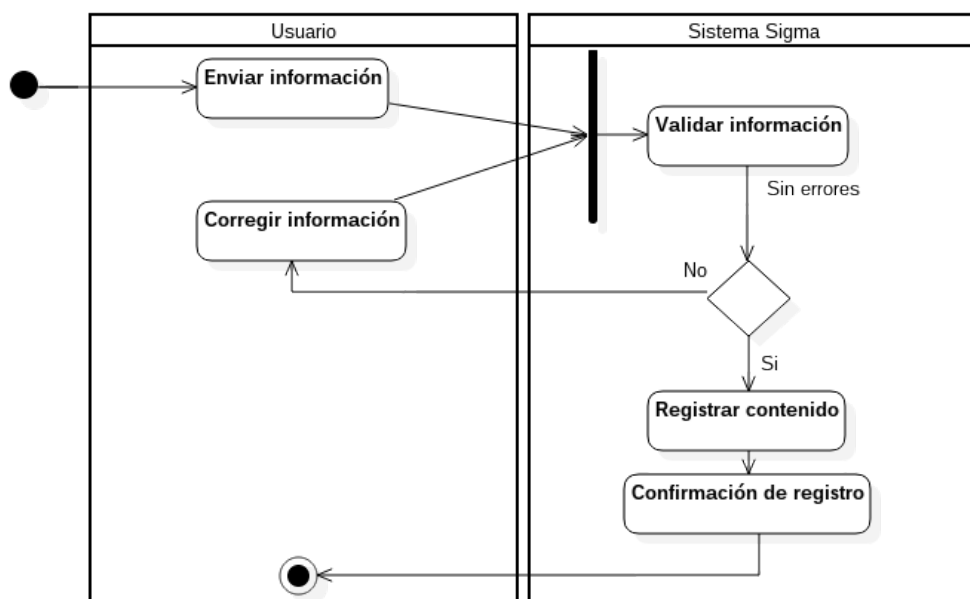


Figura 8: Diagrama de actividades. Registrar un nuevo contenido

El flujo de procesos para un registro de la acción «me gusta» en el sistema se realiza en interacción con Facebook. Cuando el usuario envía la información al sistema, este confirma que la información crea un objeto dentro del grafo social de Facebook a partir de ahí el sistema crea el registro que representará al objeto en el grafo de Facebook.

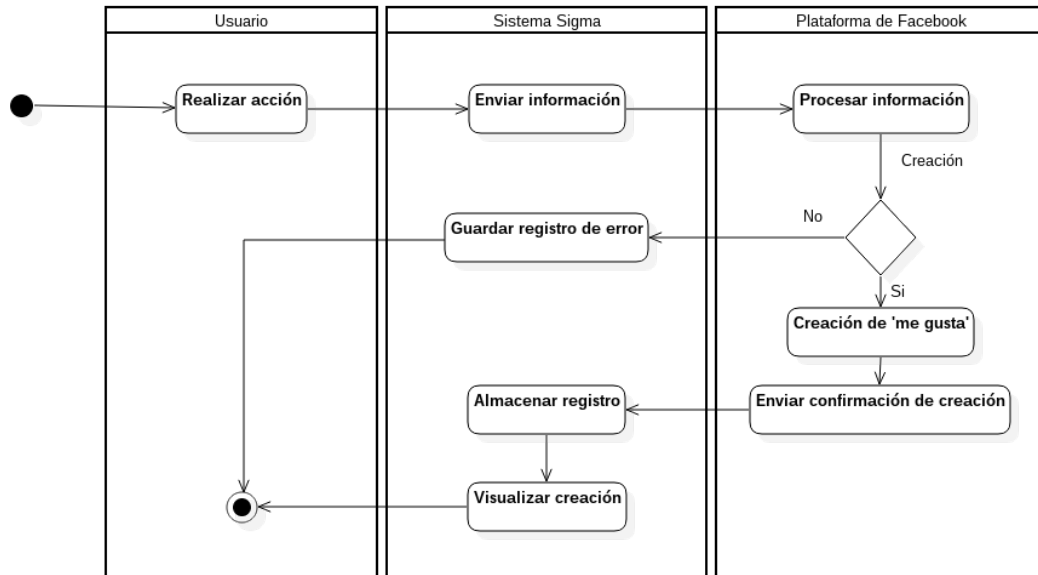


Figura 9: Diagrama de actividades. Realizar acción «me gusta»

Para eliminar un registro de la acción «me gusta» en el sistema, se lo realiza en interacción con Facebook, esto dado que se debe eliminar el objeto que representa al «me gusta» en el grafo social de Facebook.

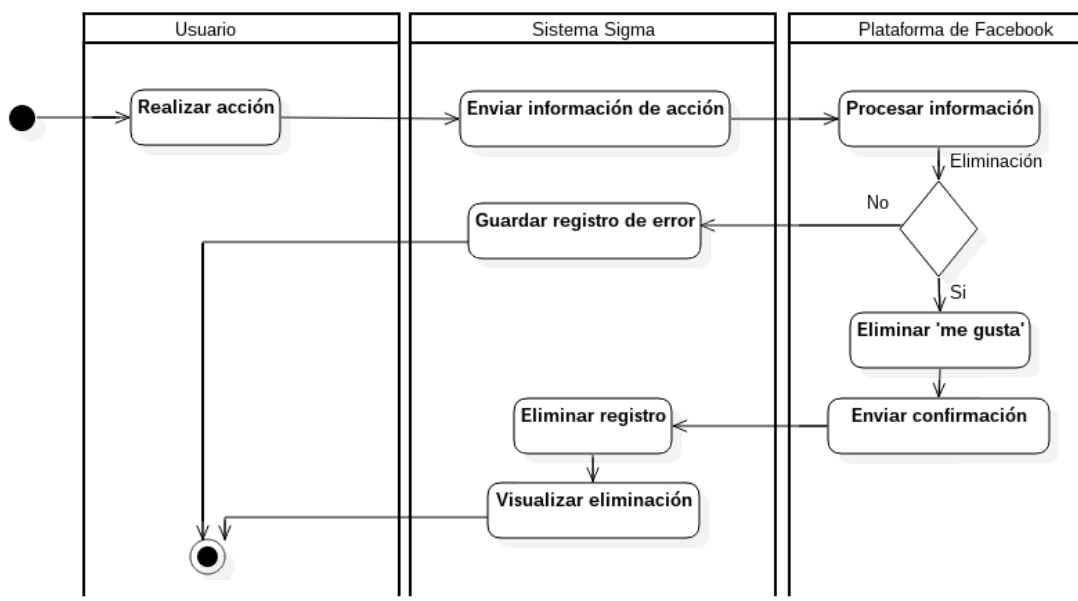


Figura 10: Diagrama de actividades. Eliminar «me gusta»

En la acción de «compartir» existe una interacción entre la plataforma de Facebook, el sistema y el usuario. El usuario realiza la acción, el sistema envía la información referente a la acción a Facebook el cual procesa la información.

Si no existiera un error, se envía al sistema la confirmación de la creación del objeto donde se guarda la información de la acción «compartir» y el sistema, en base al identificador de la confirmación, realiza una consulta respecto al objeto creado para obtener la información de su estructura y contenido. En base a la información recuperada se crea el registro.

El flujo termina con la actualización del contador de acciones «compartir» que posee el contenido en cuestión.

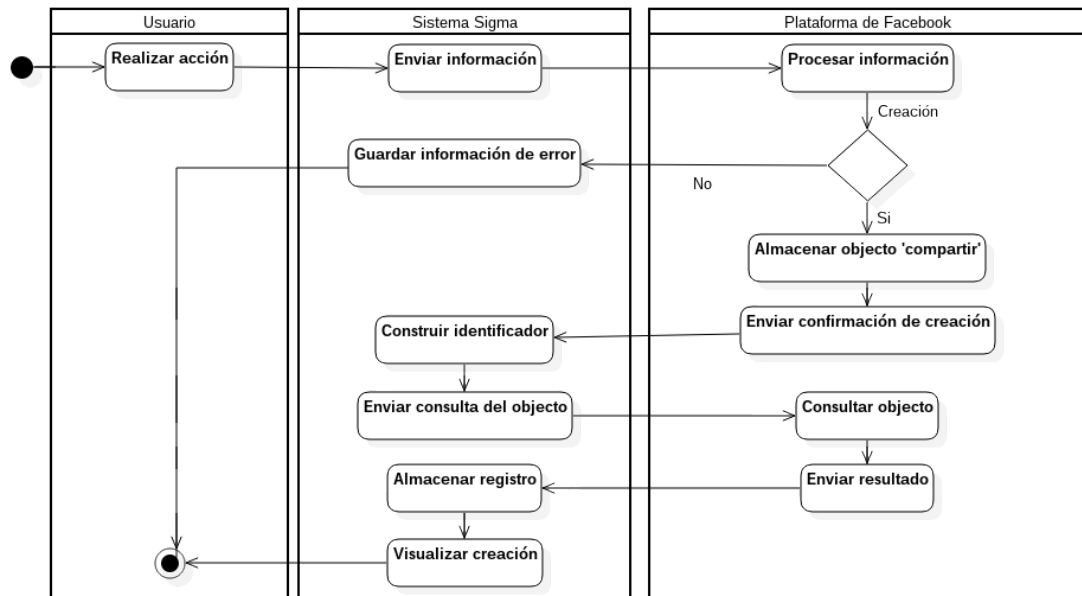


Figura 11: Diagrama de actividades. Realizar acción «compartir»

4.3.4.1. Transición de estados

A continuación se describen los estados posibles de objetos utilizando los *diagramas de transición de estados* con el fin de indicar el comportamiento de los componentes del sistema. Se describe a partir de objetos específicos con el fin de indicar la transición que existe de un estado a otro, a partir de los eventos que interactúan con el objeto [58].

El estado de la sesión de usuario posee transiciones respecto a diversos eventos en los que interviene el usuario, el sistema, y la plataforma de Facebook. La sesión posee un tiempo para estar disponible que, transcurrido el tiempo la sesión, entra a un estado de inactividad, siendo necesario re-inicializarse.

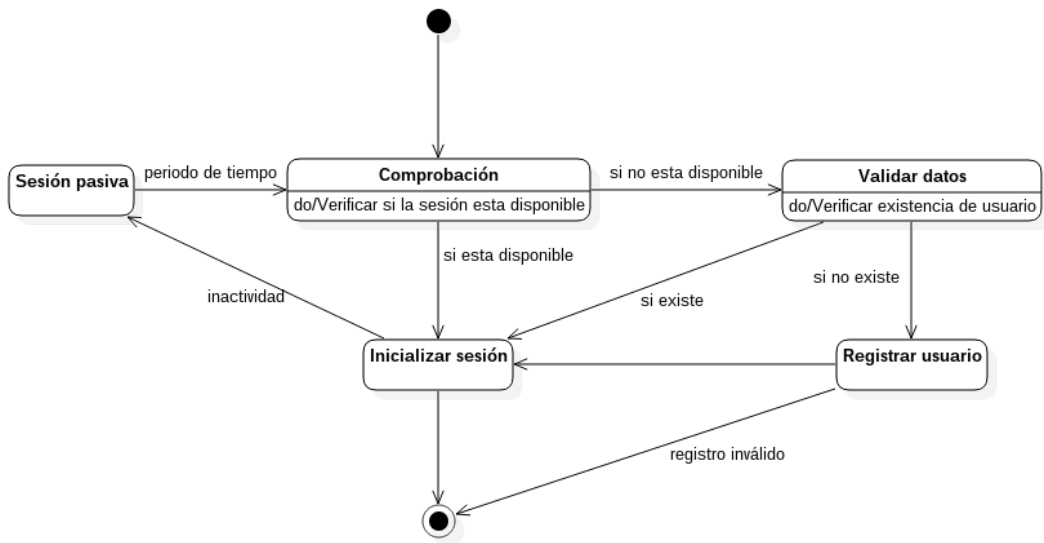


Figura 12: Diagrama de transición de estados. Sesión

La sesión solo puede ser inicializada por usuarios que se encuentren registrados en el sistema. Si no existe el usuario, se lo registra en el sistema y se inicializa la sesión.

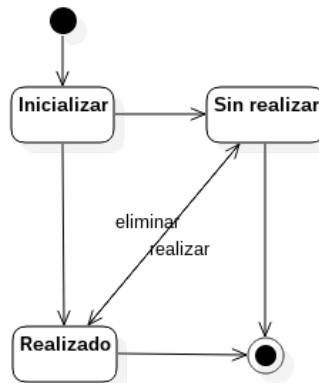


Figura 13: Diagrama de transición de estados. Acción «me gusta»

El objeto referente a la acción «me gusta» transita por dos estados fundamentales: el de haberse realizado la acción y el de no tener realizada la acción. La inicialización se origina al momento que un contenido es mostrado al usuario.

El objeto de la acción «share» posee un estado de espera, durante el cual tiene lugar la creación del objeto en Facebook. Una vez que se cumple con el registro en Facebook se da la transición al estado de estar registrado.



Figura 14: Diagrama de transición de estados. Acción «share»

4.4. Desarrollo del sistema de información

4.4.1. Especificación de casos de uso

Se procede a explicar los diferentes casos de uso identificados, con el propósito de detallar los procesos y operaciones de los mismos.

Tabla 1: Caso de uso. Registrar “visita”

Caso de Uso :	Registrar “visita”
Resumen :	Almacenar y actualizar el número de visitas, registrar el tiempo empleado desde la apertura del diálogo de visualización.
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible. Solicitar la visualización de un contenido y haber transcurrido 15 segundos desde la apertura del diálogo.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor se encuentra navegando en el sitio web de origen del contenido en el diálogo de visualización. ▪ El actor cierra el diálogo. ▪ El sistema registra la visita y el tiempo utilizado desde la apertura del diálogo. ▪ Actualiza en el contenido el número de la visitas que posee.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ El actor dispuso visualizar el sitio web origen externamente. <ul style="list-style-type: none"> • Se registra la visita. • Se define como tiempo utilizado a 1 segundo.
Poscondiciones :	Registros almacenados, actualizados, consultados.

Tabla 2: Caso de uso. Inicio de sesión

Caso de Uso :	Inicio de Sesión
Resumen :	Permite ingresar al actor en el sistema.
Actor :	Usuario
Precondiciones :	El actor debe disponer de una cuenta en Facebook
Flujo Normal :	<ul style="list-style-type: none">▪ El actor solicita el ingreso al sistema desde la plataforma de Facebook.▪ El sistema comprueba la existencia del actor dentro del registro de datos.▪ El sistema comprueba la disponibilidad de la sesión.▪ Visualizar el área de contenidos.
Flujo Alternativo :	<ul style="list-style-type: none">▪ El usuario no existe.<ul style="list-style-type: none">• El sistema genera una solicitud de acceso a la información del actor.• Facebook generará una ventana modal en que el usuario permitirá el acceso.• Facebook retorna un token de acceso.• Visualizar el área de contenidos.▪ El tiempo de la sesión ha caducado.<ul style="list-style-type: none">• El sistema genera una solicitud de acceso a la información del actor.• Facebook retorna un token de acceso.• Visualizar el área de contenidos.
Poscondiciones :	Sesión de actor se encuentra inicializada y disponible

Tabla 3: Caso de uso. Visualizar recursos de contenidos

Caso de Uso :	Visualizar recursos de contenidos
Resumen :	Permiten la visualización del sitio web de origen al cual esta referenciando un determinado contenido que existe en el sistema
Actor :	Usuario
Precondiciones :	El actor debe haber iniciado sesión.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor solicita la visualización de un contenido al pulsar en el en la área de «Leer más». ▪ Se visualiza una ventana modal. ▪ El sistema verifica la existencia del directorio donde se encuentra los recursos necesarios para la visualización del sitio web de origen ▪ Se visualiza el sitio web en la área designado de la ventana modal.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ El sistema no encuentra un directorio disponible. <ul style="list-style-type: none"> • Crear un directorio a partir de la dirección url del contenido. • Se descarga los recursos necesarios para la visualización. • Se visualiza el sitio web en la área designado del diálogo. ▪ El actor solicita la visualización de un contenido al pulsar en uno de los nombres de la lista en «Contenidos relacionados». <ul style="list-style-type: none"> • Visualizar una ventana modal. • El sistema verifica la existencia del directorio donde se encuentra los recursos necesarios para la visualización del sitio web de origen • Se visualiza el sitio web en la área designado de la ventana modal.
Poscondiciones :	Ejecutar proceso para el registro de la visita

Tabla 4: Caso de uso. Visualizar recursos de contenidos

Caso de Uso :	Registrar un nuevo contenido
Resumen :	Almacenar un nuevo registro referente a un contenido académico
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor realiza la pulsación del botón flotante «+». ▪ Se visualiza el formulario de registro. ▪ El actor ingresa la información necesaria. ▪ Acepta el guardar el contenido. ▪ El sistema valida la información. ▪ Almacenar registro. ▪ Visualizar mensaje de éxito de registro.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ El actor ingreso datos incorrectamente <ul style="list-style-type: none"> • Se muestra mensaje de error.
Poscondiciones :	Registro almacenados y consultados.

Tabla 5: Caso de uso. Filtrar contenidos

Caso de Uso :	Filtrar contenidos.
Resumen :	Visualizar contenidos que suplan con parámetros predefinidos de selección y ordenamiento
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor realiza la pulsación de una opción de filtrado. ▪ El sistema procesa la petición. ▪ Mostrar mensaje de encontrarse procesando información. ▪ Visualizar los contenidos que cumplan con el criterio de selección.
Flujo Alternativo :	
Poscondiciones :	Registros consultados.

Tabla 6: Caso de uso. Buscar contenidos

Caso de Uso :	Buscar contenidos
Resumen :	Consultar contenidos que cumpla con los parámetros de búsqueda.
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor ingresa parámetros. ▪ El actor solicita la búsqueda. ▪ El sistema valida la sintaxis. ▪ Visualiza los contenidos que cumplan con los parámetros de búsqueda.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ El actor ingreso parámetros de búsqueda que no tiene contenidos que los cumplan. <ul style="list-style-type: none"> • Mensaje de error de búsqueda sin resultados . ▪ El actor ingreso parámetros de búsqueda con errores <ul style="list-style-type: none"> • Mensaje de error. ▪ El actor solicito búsqueda sin ingresar parámetros. <ul style="list-style-type: none"> • Visualizar contenidos recientes.
Poscondiciones :	Ninguna

Tabla 7: Caso de uso. Gestionar los “me gusta”

Caso de Uso :	Gestionar los “me gusta”
Resumen :	Consultar, agregar, actualizar la acción social “me gusta” del sistema.
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor dispone de la acción al pulsar el botón en determinado contenido. ▪ Se envía información a la plataforma de Facebook. ▪ Retorno de mensaje con éxito del registro de la información en Facebook. ▪ Disponibilidad para el registro de la acción
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ No es posible realizar la acción social <ul style="list-style-type: none"> • El sistema registra en el archivo de monitoreo las circunstancias y contexto del error. • No se cambia el estado de la acción “me gusta” del contenido.
Poscondiciones :	Registro en Facebook consultado y almacenado.

Tabla 8: Caso de uso. Realizar la acción “me gusta”

Caso de Uso :	Registrar la acción de “me gusta”
Resumen :	Consultar, agregar, actualizar los “me gusta” de un contenido
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor dispone del botón para accionar un “me gusta” . ▪ Botón de la acción «me gusta» se inhabilita. ▪ Cambio del estado del botón a espera. ▪ El sistema confirma el éxito del registro de la acción en Facebook. ▪ Almacenar registro de la acción en el sistema. ▪ Estado en el botón de un “me gusta” registrado. ▪ Habilitación del botón. ▪ Actualización de la información del contenido.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ Existió un error al registrar la información referente al “me gusta” en Facebook. <ul style="list-style-type: none"> • Visualización de un “me gusta” no registrado en el estado del botón. • Habilitación del botón de la acción «me gusta».
Poscondiciones :	Registros almacenados,actualizados,consultados.

Tabla 9: Caso de uso. Eliminar un “me gusta”

Caso de Uso :	Eliminar un “me gusta”
Resumen :	Eliminar y actualizar “me gusta” de un contenido
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible. El contenido sobre el cual se realiza la acción debe haberse registrado previamente un “me gusta”
Flujo Normal :	<ul style="list-style-type: none">▪ El actor dispone del botón de la acción «me gusta» de un contenido .▪ Botón de la acción «me gusta» se inhabilita.▪ Cambio del estado del botón a espera.▪ El sistema confirma el éxito de eliminación de la acción en Facebook.▪ Eliminar registro de la acción en el sistema.▪ Estado en el botón de un “me gusta” no registrado.▪ Habilitación del botón.▪ Actualización de la información del contenido.
Flujo Alternativo :	<ul style="list-style-type: none">▪ Existió un error al registrar la eliminación de un “me gusta” en Facebook.<ul style="list-style-type: none">● Visualización de un “me gusta” registrado en el estado del botón.● Habilitación del botón de la acción «me gusta».
Poscondiciones :	Registros almacenados,actualizados,consultados.

Tabla 10: Caso de uso. Realizar la acción de “compartir”

Caso de Uso :	Realizar la acción de “compartir”
Resumen :	Consultar, agregar, actualizar los share de un contenido
Actor :	Usuario
Precondiciones :	El actor debe tener una sesión inicializada y disponible.
Flujo Normal :	<ul style="list-style-type: none"> ▪ El actor dispone del botón para accionar un share . ▪ Se cambia el estado del botón a espera de la respuesta del sistema . ▪ Visualizar diálogo para llenar información del contenido a compartirse. ▪ El sistema envía la información del contenido a hacer share en Facebook. ▪ Facebook retorna un identificador de la realización de la acción. ▪ Consultar información adicional en base del identificador de la acción. ▪ Se registra la acción. ▪ Se cambia el estado del botón. ▪ Actualización de la información del contenido.
Flujo Alternativo :	<ul style="list-style-type: none"> ▪ El actor realizo por primera vez la acción y existió un error al registrar la información de la acción en Facebook. <ul style="list-style-type: none"> • El sistema registra en el archivo de monitoreo las circunstancias y contexto del error. • Visualización de no haber realizado aún ninguna acción de “compartir” en el estado del botón. • Habilitación del botón de la acción «compartir». ▪ El actor ejecuto nuevamente la acción en un contenido que previamente compartió y existió un error al registrar la información de la acción en Facebook. <ul style="list-style-type: none"> • El sistema registra en el archivo de monitoreo las circunstancias y contexto del error. • Habilitación del botón de la acción «compartir».
Poscondiciones :	Registros almacenados,actualizados,consultados.

4.4.2. Producción de software

4.4.2.1. Arquitectura

La arquitectura clásica cliente/servidor es de amplio uso para el desarrollo de aplicaciones que se ejecutan en entornos web. Las dificultades que enfrenta esta arquitectura están ligadas a la necesidad de la carga y procesamiento, desde cero, de estilos, plantillas y bibliotecas en el cliente. Esto implica, que al cambiar entre vistas se sufrirá el problema de la latencia en la navegación web [60].

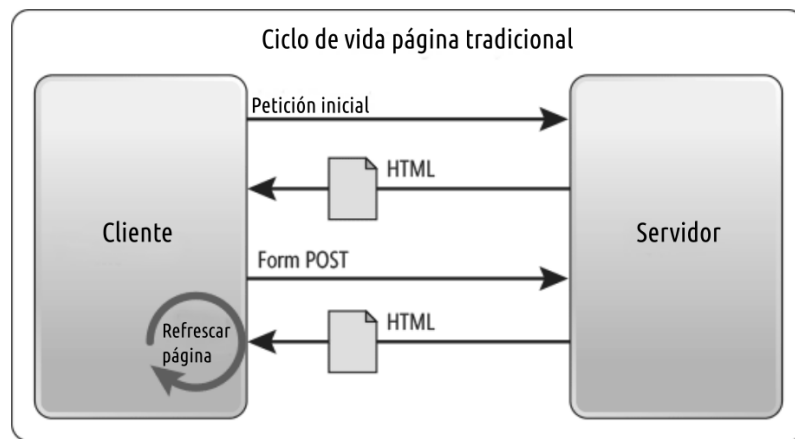


Figura 15: Arquitectura cliente/servidor tradicional [5]

La arquitectura que se utiliza para el sistema de información «Sigma» es la arquitectura SPA (single page application), la que constituye una modificación a la arquitectura tradicional cliente/servidor en la que las interfaces del sistema se procesan y despliegan como parte del contenido de la página web inicialmente cargada en el navegador; dicha página no se recarga sino que se actualiza su contenido durante el tiempo en que se usa la aplicación [6][5].

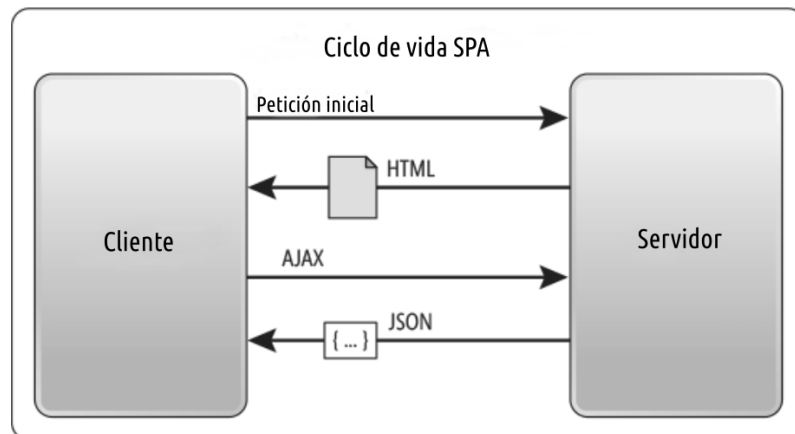


Figura 16: Arquitectura cliente/servidor SPA [6]

La mayor parte de la lógica de negocio de la aplicación se lleva al cliente. El código en la capa del servidor se usa básicamente para proveer de una API RESTful, a la cual accede el código cliente haciendo uso de AJAX [5].

En la programación del cliente se ocupó JavaScript para la programación dinámica. La implementación se apoyó en el uso del marco de desarrollo AngularJS, HTML5 y hojas de estilo CSS3.0, además de la utilización del SDK JavaScript de Facebook.

Para la programación del lado servidor, de igual manera se utilizó código Javascript, ejecutado en un servidor Node.js, sobre el que se instancia Sails.js como marco de trabajo de desarrollo principal en el servidor, permitiendo la utilización de Express.js para el procesamiento de las solicitudes HTTP y la gestión de rutas web en el sistema, Waterline.js que suple el trabajo de ORM para facilitar el acceso a la base de datos, la que se desplegó en un servidor MySQL [49]. Se integra Passport.js a Sails.js para el control de sesión de usuarios y la gestión de permisos respecto a la plataforma de Facebook.

En la figura 17 se observa la arquitectura del sistema, que hace uso de herramientas y marcos de desarrollo de software libre constituyéndose en un desarrollo realizado 100 % en código Javascript.

Sails.js permite la utilización del paradigma MVC en la programación de entornos con Node.js disponiendo una metodología para la estructura y formación de servicios y procesos en la construcción de sistemas. Sin la utilización de un marco de desarrollo como Sails.js se depende en gran manera de la experiencia del desarrollador con varios marcos de desarrollo de Node.js para la implementación de los servicios y funcionalidades que se encuentra en el sistema Sigma. En la figura 18 se puede corroborar la estructuración del código, de acuerdo a la metodología de Sails.js.

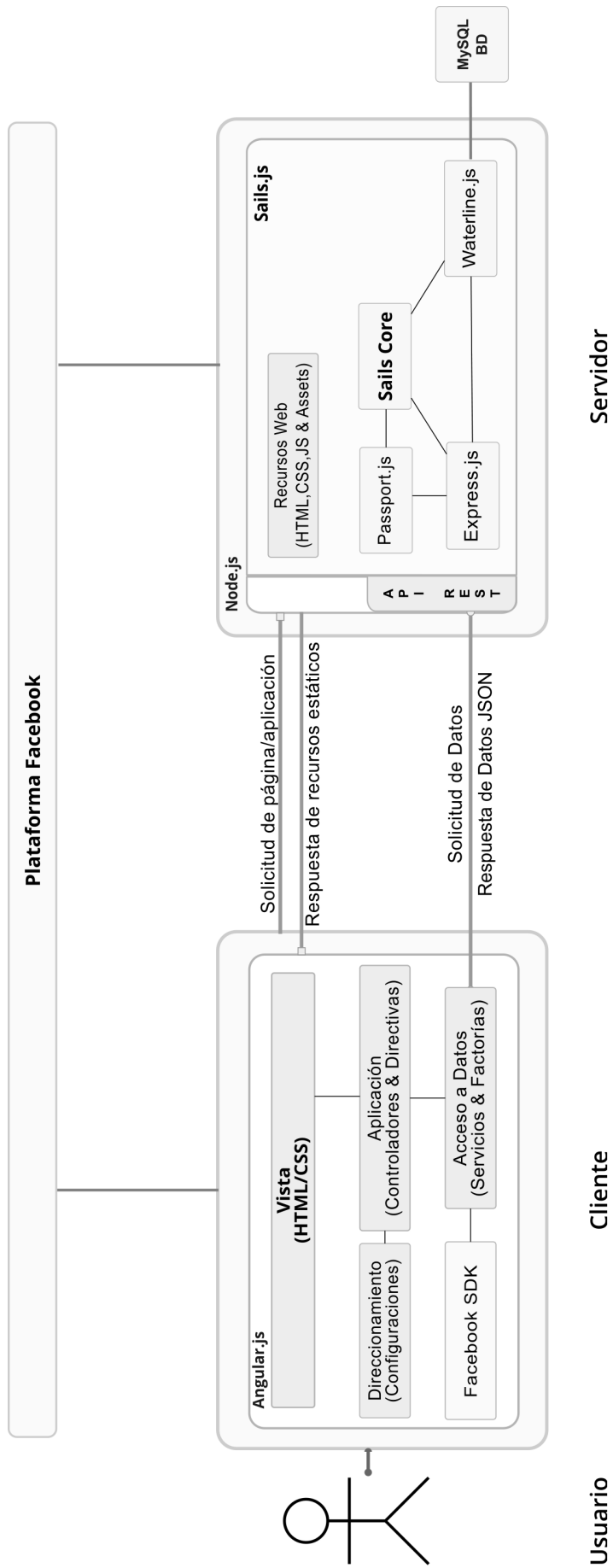


Figura 17: Arquitectura del sistema Sigma

```

|--
|  |-- blueprints
|  |-- controllers
|  |-- models
|  |-- policies
|  |-- responses
|  |-- services
|-- assets
|  |-- angular
|  |   |-- components
|  |-- bower_components
|  |   |-- angular
|  |   |-- angular-animate
|  |   |-- angular-bootstrap
|  |   |-- angular-i18n
|  |   |-- angular-lazy-img
|  |   |-- angular-loading-bar
|  |   |-- angular-local-storage
|  |   |-- angular-resource
|  |   |-- angular-route
|  |   |-- angular-sanitize
|  |   |-- bootstrap-css
|  |   |-- ng-facebook
|  |   |-- webfontloader
|  |   |-- .....
|  |-- fonts
|  |   |-- fontawesome
|  |   |-- opensans
|  |   |-- oxygen
|  |   |-- ubuntu
|  |-- icons
|  |-- images
|  |-- styles
|  |-- templates
|-- config
|  |-- env
|  |-- locales
|-- node_modules
|-- |  |-- .....
|  |-- grunt
|  |-- passport
|  |-- sails -> /usr/lib/node_modules/sails
|  |-- waterline-criteria
|-- |  |-- .....
|-- tasks
|  |-- config
|  |-- register
|-- views
|-- web-scraper
|  |-- app
|  |   |-- controllers
|  |   |-- views
|  |-- config
|  |-- core
|  |-- public
|  |-- files

```

Figura 18: Estructura de árbol de directorios del sistema Sigma

4.4.2.2. Requisitos funcionales

Elaboración diagrama de secuencias

Mediante los diagramas de secuencia es posible representar la comunicación y los mensajes del usuario respecto a la interfaz visual, representar los procesos e interacción a través del tiempo del conjunto de objetos que conforma el sistema.[58]

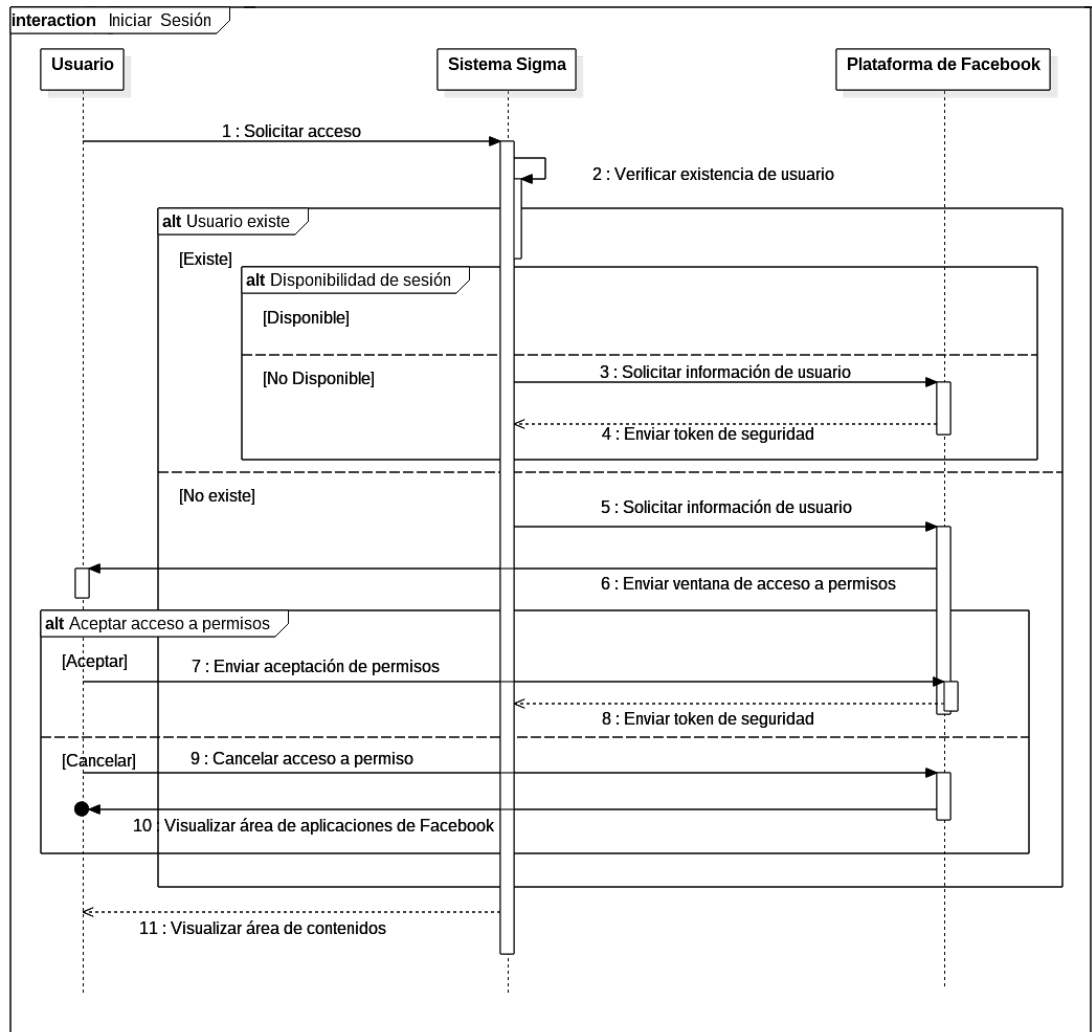


Figura 19: Diagrama de secuencias. Iniciar sesión

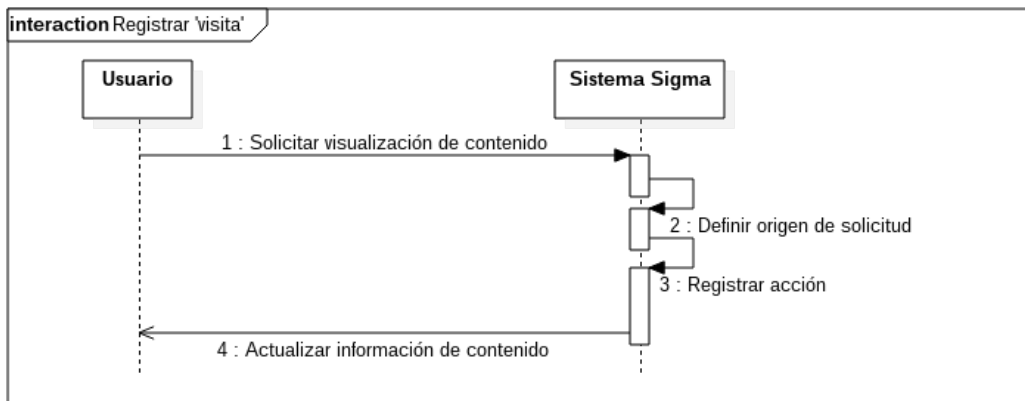


Figura 20: Diagrama de secuencias. Registrar acción «visita»

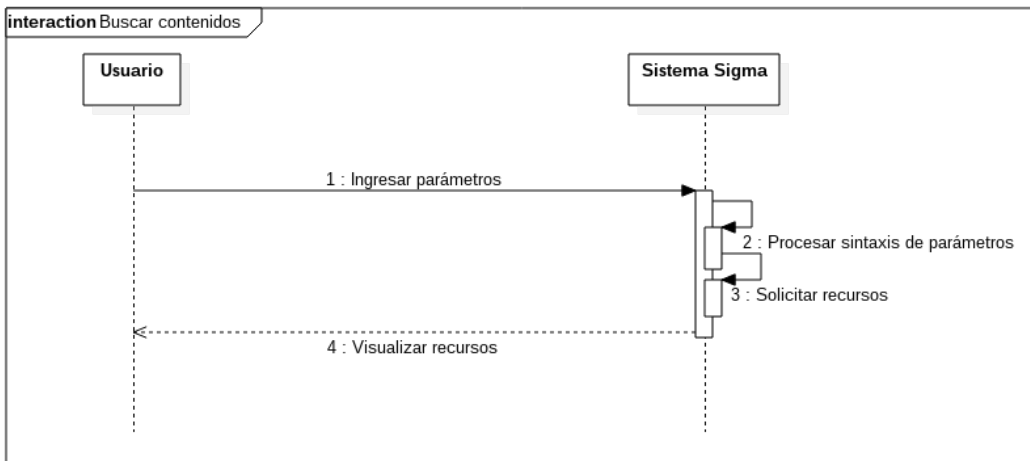


Figura 21: Diagrama de secuencias. Buscar contenidos

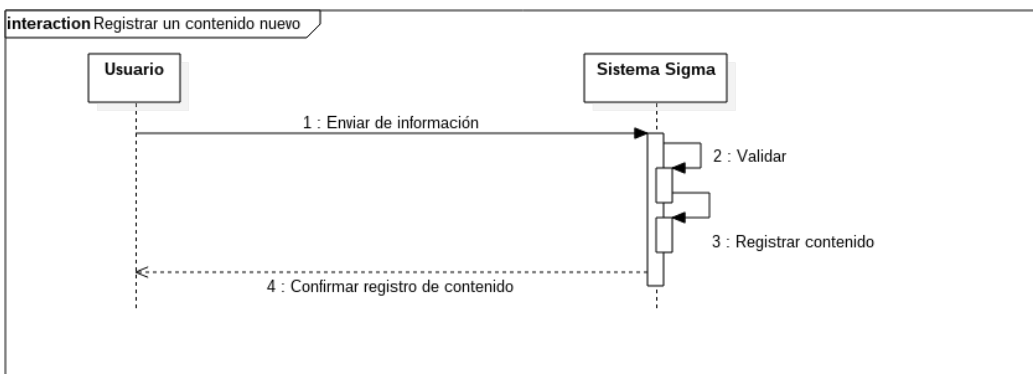


Figura 22: Diagrama de secuencias. Registrar un nuevo contenido

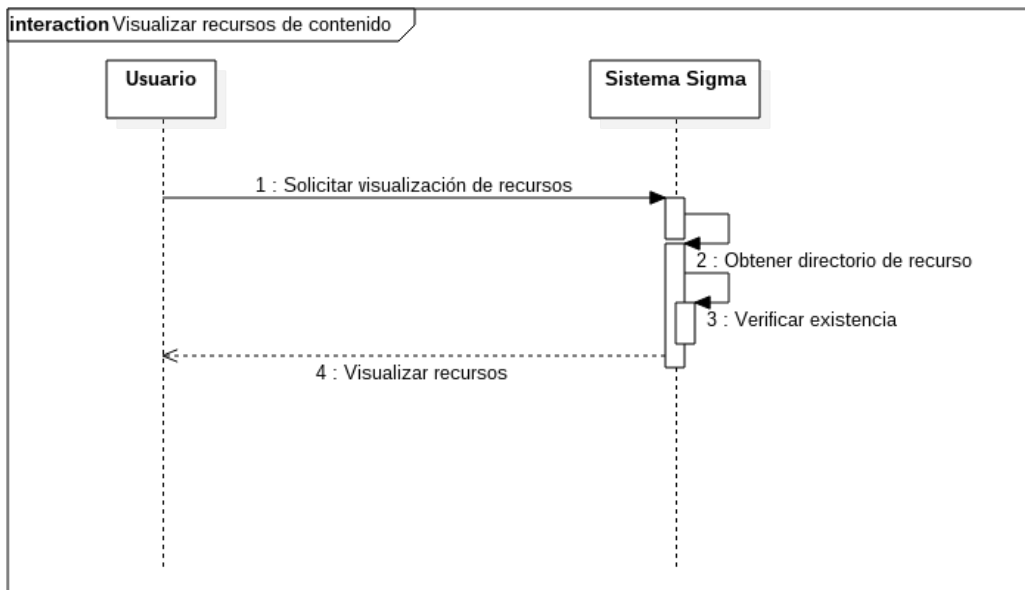


Figura 23: Diagrama de secuencias. Visualizar recursos de contenido

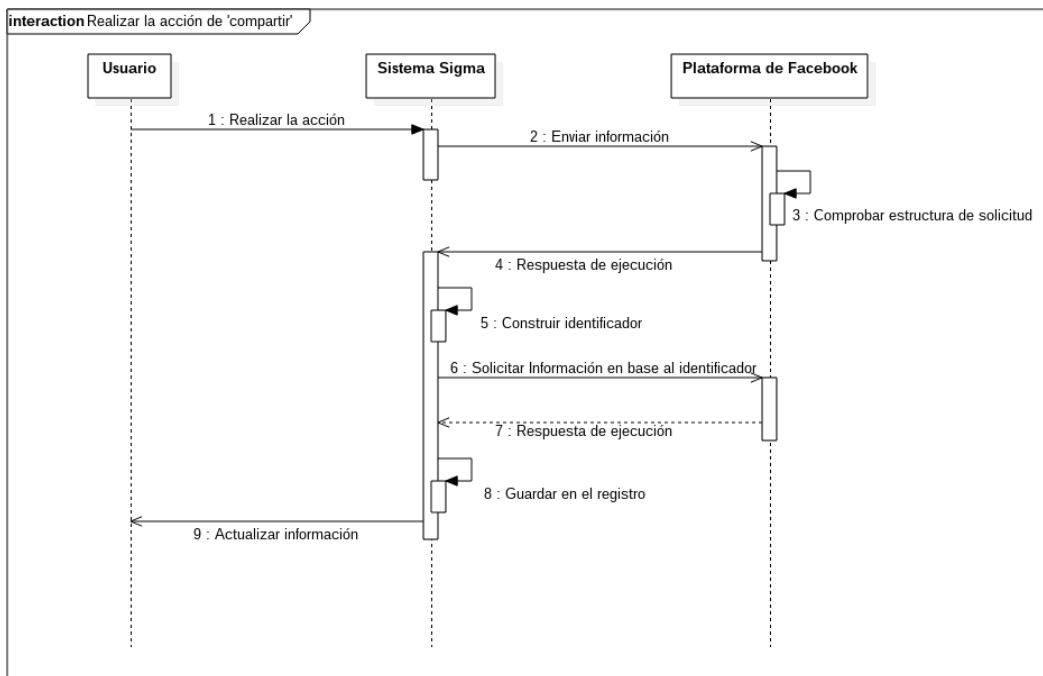


Figura 24: Diagrama de secuencias. Realizar acción «compartir»

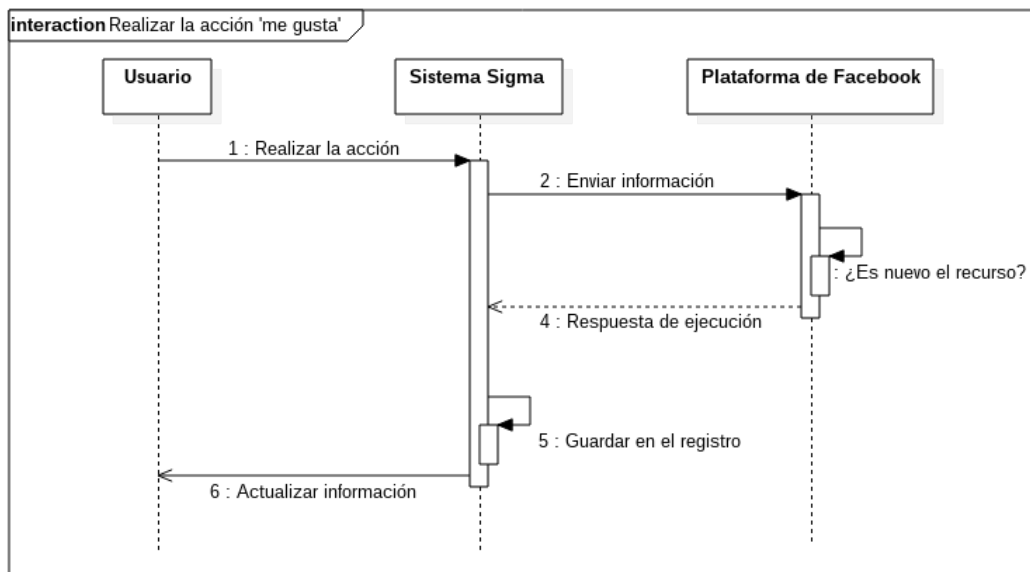


Figura 25: Diagrama de secuencias. Realizar acción «me gusta»

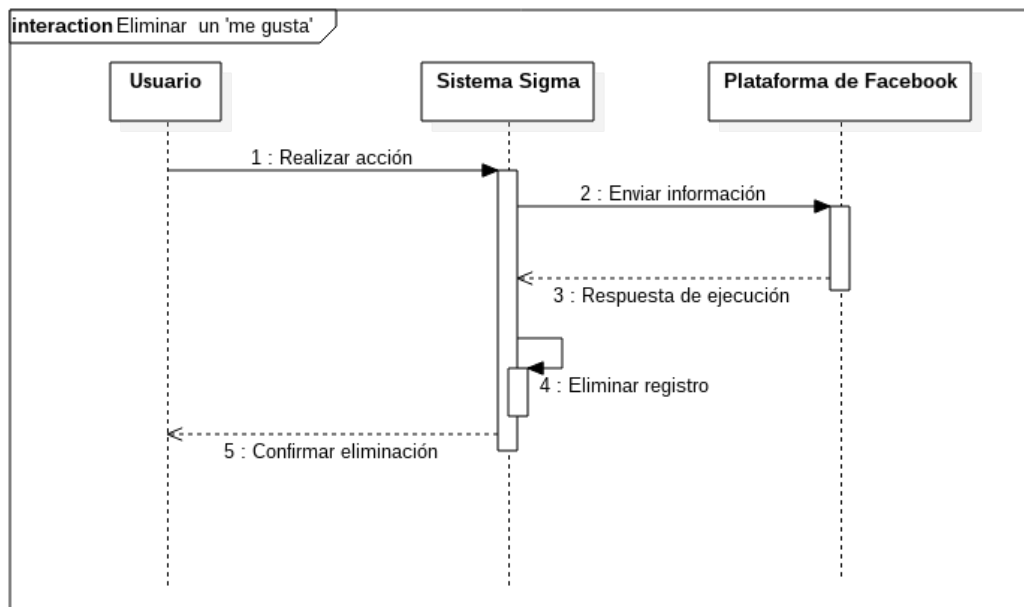


Figura 26: Diagrama de secuencias. Eliminar «me gusta»

Elaboración diagrama de clases

El diagrama de clases permite describir la estructura del sistema mostrando sus clases, atributos y cómo se relacionan unas con otras. El diagrama de clases es una técnica ampliamente utilizada para el modelado de sistemas orientados a objetos [61].

Diagrama de clases de la programación del servidor

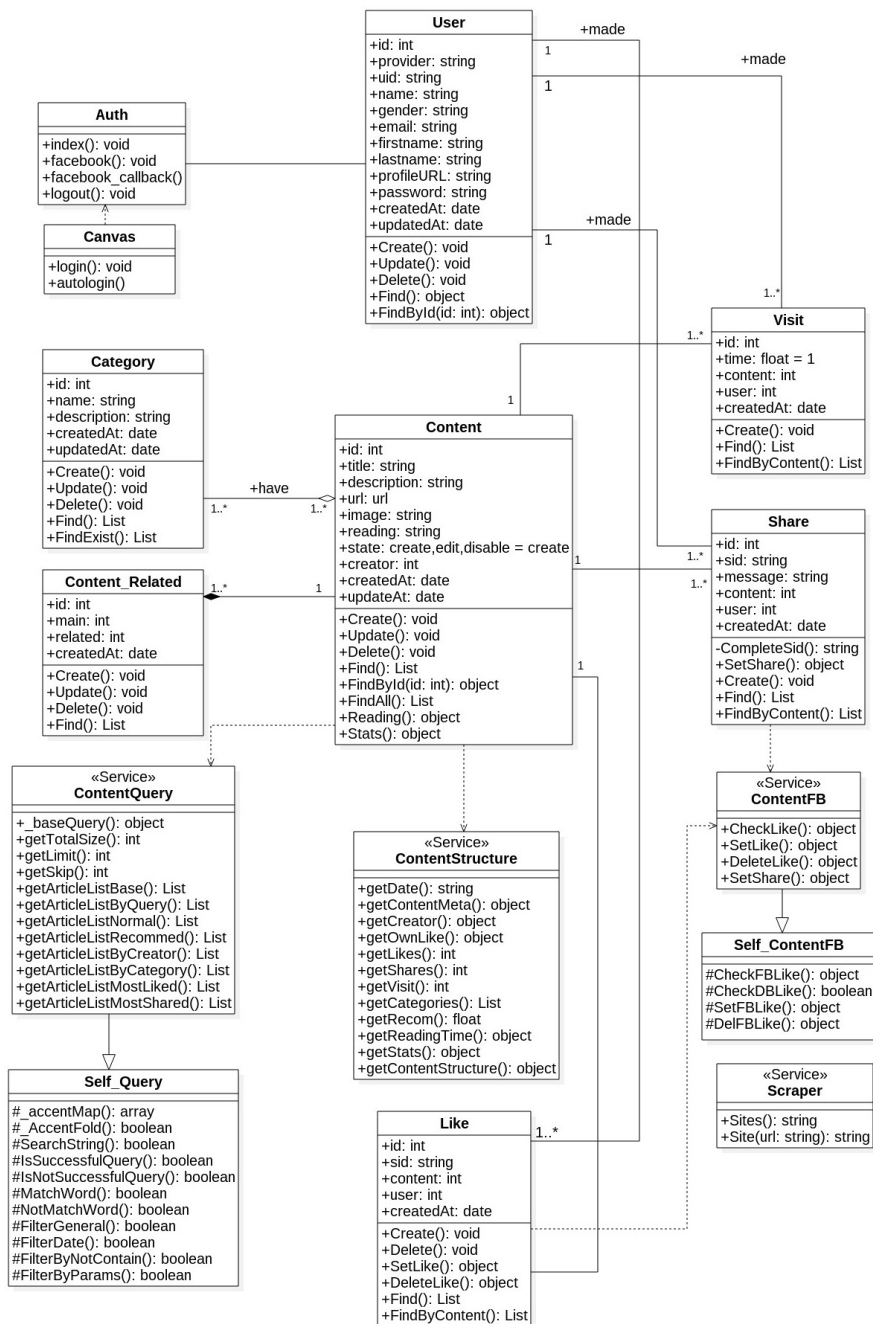


Figura 27: Diagrama de clases. Programación del servidor

Diagrama de clases de la programación del cliente

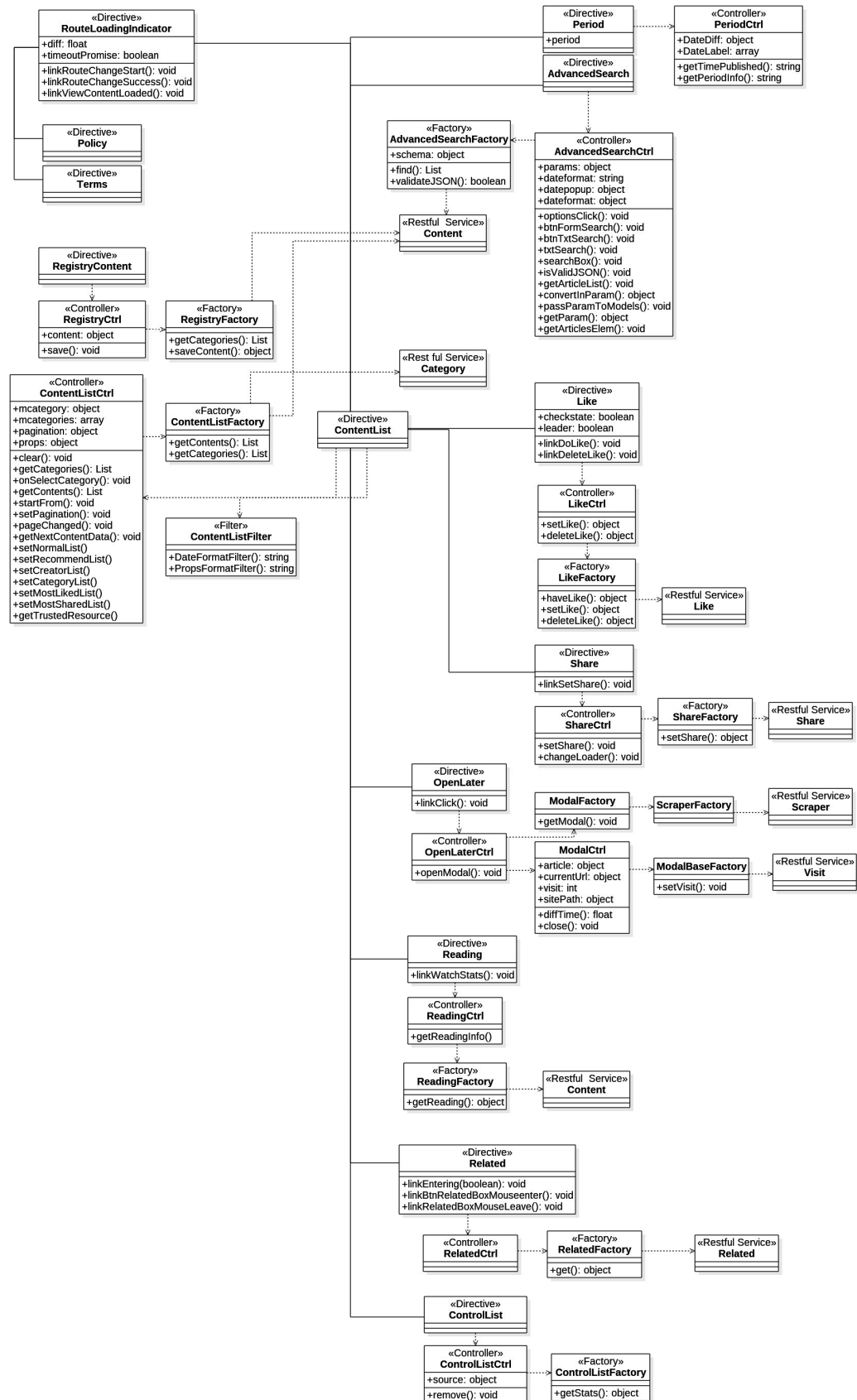


Figura 28: Diagrama de clases. Programación del cliente

Detalle de funcionalidad El sistema Sigma se integra a lo que se conoce como el *canvas* (ambiente de ejecución de aplicaciones) de Facebook, lo que hace que los usuarios la identifiquen como una interfaz que extiende las funcionalidades de la red social. En la figura 29, se puede apreciar la interfaz del sistema, integrado en el entorno de Facebook. En dicha figura se resalta la ubicación de componentes específicos del funcionamiento de Sigma.

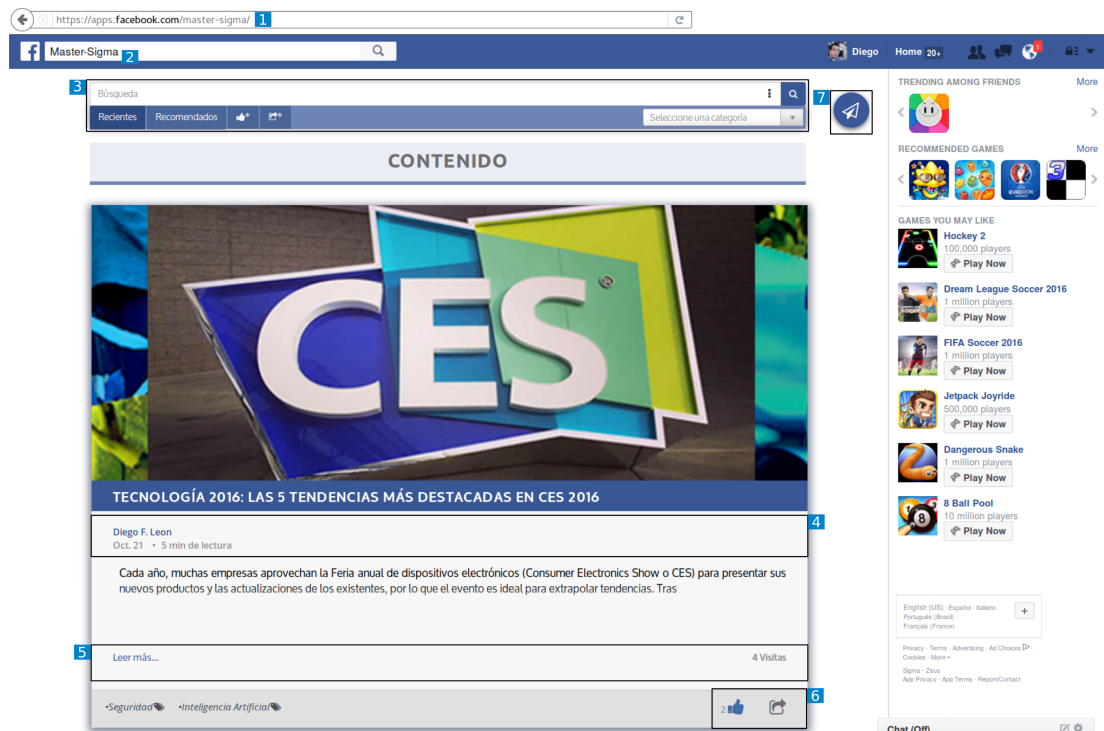


Figura 29: Interfaz de Sigma. 1) Dirección URL. 2) Identificación de aplicación en Facebook. 3) Opciones de filtrado de contenidos. 4) Metadatos. 5) Área de visualización. 6) Acciones sociales disponibles. 7) Agregar un nuevo contenido.

La comunicación entre el sistema implementado y la plataforma Facebook tiene lugar a través de la API de Facebook conocida como el *Facebook Graph*. Esto implica que lo primero que ocurre al querer usar Sigma es un proceso de *instalación* a través del cual el usuario confiere su autorización a los permisos que se requieren para que el sistema registre acciones sociales en la plataforma a nombre del usuario (por ejemplo, cuando este hace “me gusta” sobre determinado objeto de aprendizaje no directamente en Facebook sino a través de Sigma). Ese procedimiento garantiza que el sistema reciba el *token* de acceso a la plataforma (como aplicación Facebook registrada) garantizando que nunca manejará información relacionada con las contraseñas de usuario para la autenticación ante la plataforma[62].

“Me gusta” La acción «me gusta» es denominada dentro del API de Facebook como el objeto *og.likes* [44] [63]. El objeto *og.likes* dispone de una restricción y

es que se permite una única ocurrencia de esta acción social para un usuario actuando sobre determinado recurso. Es decir que determinado usuario puede realizar un solo «me gusta» sobre un contenido específico [63]. El sistema contabiliza los «me gusta» realizados por los usuarios en un determinado contenido al momento de su visualización.

En el momento de la realización de la acción «me gusta» sobre determinado contenido, se envía la dirección URL que dispone a la API de Facebook. Con ello se solicita la creación de un objeto *og.likes* que generará un identificador numérico universal, el que es enviado como respuesta. El sistema, al recibir el identificador, crea un registro que contiene los datos referentes a la acción social llevada a cabo por el usuario a través de la aplicación implementada.

```
angular.module('app.main.content.like',['app.config'])
.factory('LikeFactory',LikeFactory)
.controller('LikeCtrl',LikeCtrl)
.directive('like', function( partial,$log){
return {
restrict: 'EA',
scope: {
stats: '=',
source: '=',
likeSid:=''
},
controller: 'LikeCtrl',
controllerAs: '$like',
templateUrl: partial.main.content+'tpl/like.cmp.html',
link : function (scope, element, attrs, controller) {

scope.checkstate = false;
scope.loader = false;

// Check if exist previous like
if(scope.likeSid)
scope.checkstate = true;

scope.doLike = function(){
scope.loader = true;

if(scope.likeSid)
scope.checkstate = true;

if(!scope.checkstate)
{
controller.setLike(scope.source.id,scope.source.url)
.then(function (response){
if(response.data.created) // created,record
{
scope.likeSid = response.data.record.sid;
scope.checkstate = true;
scope.loader = false;
scope.stats = scope.stats + 1;
}
}
.catch(function(err){
$log.error(err);
scope.loader = false;
});
}
}
else {
controller.deleteLike(scope.likeSid)
.then(function(response){
if(response.data)
{
scope.checkstate = false;
scope.stats = scope.stats - 1;
scope.loader = false;
scope.likeSid = '';
}
}
}
.catch(function(err){
$log.error(err);
scope.loader = false;
});
}
});
}
});
```

Figura 30: Extracto de código. Cliente - Realizar acción «me gusta»

```

.....
setFBLike : function(contentURL){
  return new Promise(
    function (resolve,reject){
      graph.post('me/og.likes',
        {
          object:contentURL
        },
        function(err, response) {
          if(err) reject(err);
          resolve(response);
        });
    });
},
delFBLike : function(sid){
  return new Promise(
    function (resolve,reject){
      graph.del(sid,
        function(err, response) {
          if(err) reject(err);
          resolve(response);
        });
    });
}
.....
module.exports = {
  like : {
    check: function(contentID){
      return self.checkDBLike(contentID).then(function (exist){
        return exist;
      });
    },
    set : function (contentID,contentURL,userID) {
      return self.setFBLike(contentURL)
        .then(function(reslike){ return [reslike]; })
        .spread(function (reslike) {
          return Like.create({
            sid : reslike.id,
            content : contentID,
            user : userID
          })
            .then(function(created){
              sails.log.debug('+ Set like :'+created.sid);
              return ({created : true,record : created});
            });
        })
        .catch(function(err){
          sails.log.warn(err.message);
          sails.log.debug(err);
          throw 'Like: Not set like err';
        });
    },
    delete : function (sid) {
      return self.delFBLike(sid)
        .then(function(resdel){ return [resdel]; })
        .spread(function (resdel) {
          return Like.destroy({
            sid : sid
          }).then(function deleteRecord(){
            sails.log.debug('+ Deleted like :'+sid);
            return ({deleted : true});
          })
            .catch(function(err){
              sails.log.warn(err);
              return ({deleted:false,err : err});
            });
        })
        .catch(function(err){
          sails.log.warn(err);
          throw 'Share: Not set share err';
        });
    }
  }
}

```

Figura 31: Extracto de código. Servidor - Realizar acción «me gusta»

“**Compartir**” La acción dentro de la API de Facebook adquiere el estado de objeto *Post*, que es una publicación del usuario que se visualiza en su muro social. El tipo de esta publicación se denomina *link*, y es una dirección URL referencia hacia determinado sitio web [44][64]. Para la realización de la acción «compartir» sobre un determinado contenido no existe un límite numérico de creación.

El sistema provee la generación de la ventana modal a partir de la dirección URL del contenido a compartir. En dicha ventana, el usuario establecerá el nivel de privacidad y podrá personalizar el contenido del mensaje de la publicación Facebook correspondiente.

La ventana modal se genera a través del SDK de Facebook, por lo que su apariencia y funcionamiento es independiente del sistema. Para el acceso al mensaje e información adicional de la publicación generada en la ventana modal se utiliza el identificador numérico que retorna Facebook al momento de confirmar el éxito del registro del objeto dentro de su grafo social.[64]

En base al identificador numérico en conjunto con el identificador del usuario se realiza, en la programación del cliente, la consulta del objeto de la publicación en la red social (utilizando la API de Facebook), para obtener, en el caso que exista, el mensaje con el que se comparte el contenido en cuestión en el muro social del usuario. Con los datos que devuelve la consulta, se crea el registro de haber realizado la acción «compartir».



Figura 32: Ventana modal para compartir en el muro social del usuario.

```

angular.module('app.main.content.share', [])
.factory('ShareFactory',ShareFactory)
.controller('ShareCtrl',ShareCtrl)
.directive('share', function($log,$facebook,partial,$Session){
  return {
    restrict: 'EA',
    scope: {
      stats: '=',
      source: '=' //content
    },
    transclude: true,
    controller: 'ShareCtrl',
    controllerAs: '$share',
    templateUrl: partial.main.content+'tpl/share.cmp.html',
    link: function(scope, element, attr,controller) {
      scope.setShare = function () {
        scope.loader = true;
        $log.debug('+ Click Share Button');
        $facebook.ui(
          {
            method: 'share',
            href: scope.source.url
          })
        .then(function(response){
          if(response && !response.error_code)
            {
              var user = $Session.get().user;
              $log.debug('+ FB Share user ',user);
              $log.debug('+ FB Share user.uid ',user.uid);
              var completeSID = user.uid+'_'+response.post_id;
              $log.debug('+ FB Share Complete ID',completeSID);
              $facebook.api('/'+completeSID)
                .then(function(share){
                  if(share){
                    $log.debug('+ Share success by adding in DB');
                    controller.setShare(share.id,scope.source.id,share.message);
                    scope.loader = false;
                    scope.stats = scope.stats+1;
                  }
                });
            }
          });
        .catch(function () {
          scope.loader = false;
        });
      };
    }
  });
};

```

Figura 33: Extracto de código. Cliente - Realizar acción «compartir»

```

module.exports = {
  share : {
    set : function (shareSID,contentID,userID,messageShare) {
      return Share.create({
        sid : shareSID,
        content : contentID,
        user : userID,
        message : messageShare
      }).then(function createRecord(created){
        sails.log.debug('+ Set share :'+JSON.stringify(created));
        return ({created : true,record : created});
      },
      function (err) {
        sails.log.warn(err);
        return ({created:false,err : err});
      }
    );
  }
};

```

Figura 34: Extracto de código. Servidor - Realizar acción «compartir»

“Registrar un nuevo contenido” El sistema permite al usuario la creación de un nuevo contenido; esto se lo realiza mediante un formulario que se carga en una pantalla del sistema, a través de la cual se ingresa la información necesaria para el registro de un nuevo contenido en la base de datos del sistema.

El usuario ingresa la información, la que se valida en la capa cliente de la aplicación. Si los datos son válidos, se envían desde la capa del cliente hacia la del servidor a través de la petición correspondiente al servicio web RESTful, implementado con el propósito de guardar los datos en la base de datos del sistema.

```
function RegistryFactory($http){
    savecontent : function(content){
        return $http.get('/me').then(function (response){
            content.creator = response.user.id;
            return $http.post('/api/content/create', content).then(function (response){
                console.log(response);
                return response;
            }).catch(function (err) {
                console.error(err.stack);
            });
        }).catch(function (err) {
            console.error(err.stack);
        });
    }

function RegistryCtrl($scope, RegistryFactory){

    var $registry = this;
    $registry.content = {};

    RegistryFactory.getCategories()
        .then(function (response){
            $registry.categories = response;
        })
        .catch(function (err) {
            console.error(err.stack);
        });

    $registry.reset = function(){
        $registry.content = {};
        $registry.contentForm.$setPristine();
    };

    $registry.save = function(){
        if(JSON.stringify($registry.content) === '{}')
        {
            console.log('Not content info');
        }else{
            RegistryFactory.savecontent($registry.content).then(function(response){
                if(response.status == 201)
                {
                    $registry.reset();
                }
            });
        }
    };
}
}
```

Figura 35: Extracto de código. Cliente Registrar un nuevo contenido

“Visualización de origen de contenidos” El sistema permite la visualización del origen del contenido, es decir la visualización del sitio web original del cual hace referencia el creador y a partir del cual realiza la descripción. Se determina dos formas de acceder al origen del sitio web de un contenido, la visualización dentro del sistema y la apertura directa del sitio en una nueva pestaña. La posibilidad de visualizar el origen del contenido dentro del sistema permite al usuario una mejor interacción con el resto del sistema y el consumo de información que respecto a la visualización en una nueva pestaña en el navegador.

La visualización dentro del sistema se lo realiza en una ventana modal que mediante la utilización de un *iframe*, figura 37 en el que se cargan los recursos necesarios para disponer de la visualización más parecida a la del sitio web de origen. El sistema en la petición de visualización busca el directorio dentro del servidor en el cual se encuentran los recursos del sitio web si no llegará a existir el sistema procede a descargar el sitio web , si ya se encontrará descargados los recursos en su respectivo directorio se retorna la ruta de acceso al directorio.

La acción del requerir la visualización del sitio web de origen de un contenido en el sistema se considera como «visita». La acción «visita» se determina como parte de las acciones social, dado por el valor que añade al conocer si para un usuario el contenido que esta visualizando es de interés siendo así necesario conocer más acerca de ese contenido. Los procesos en los cuales el sistema almacenará como una «visita» son en la pulsación del área de la imagen del contenido y al momento de la pulsación del texto *Leer más* estas zonas se observan en la figura 36.

En la pulsación de la imagen del contenido, se abre una nueva pestaña en el navegador en la que se solicita el sitio web de origen respecto a este proceso no es posible conocer cuanto tiempo el usuario estuvo navegando en el sitio web, por lo que se almacena el registro de la visita con un tiempo de un 1 segundo, se realiza de esta forma para poseer un indicador de que el usuario deseo visualizar directamente un contenido.

En la visualización dentro del sistema que es a partir de la pulsación del texto *Leer más* se trata de evitar la generación de falsos positivos, se almacenará el registro de la visita después de cerrarse la ventana modal siempre y cuando el tiempo que el usuario estuvo navegando dentro de la ventana es mayor a 15 segundos, de esta forma se asegura que el usuario no abrió por error la visualización del contenido.

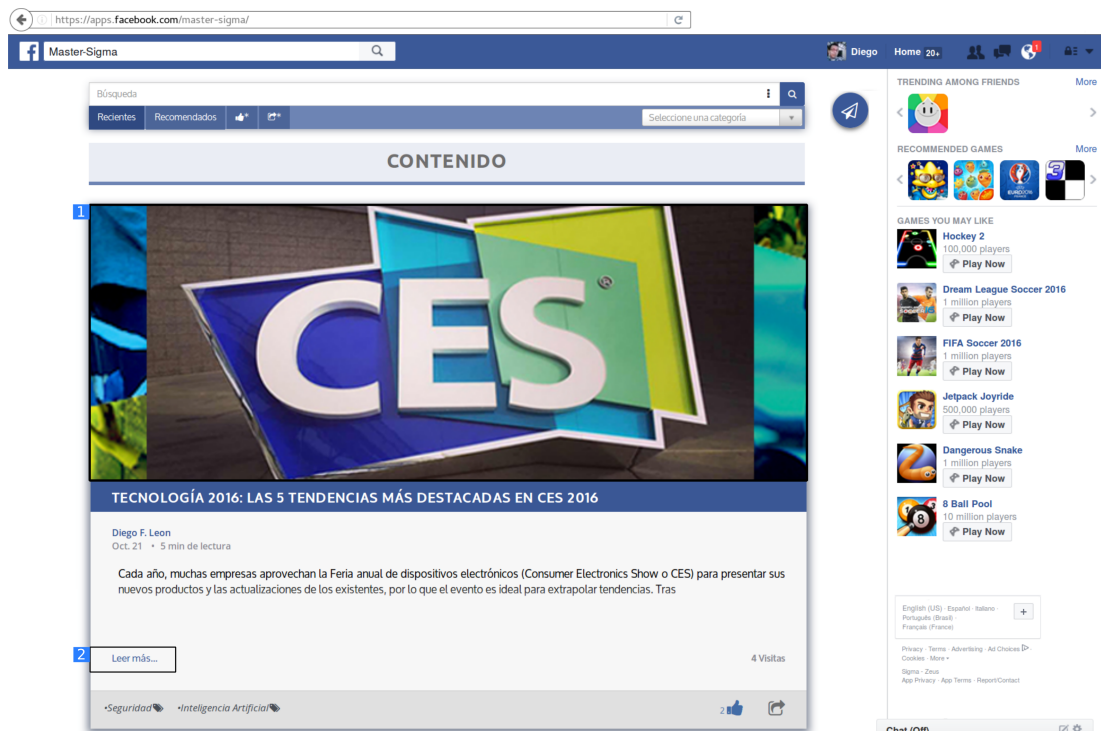


Figura 36: Áreas de acceso a visualización de origen. 1) Visualizar en una nueva pestaña. 2) Visualizar dentro del sistema en una ventana modal.



Figura 37: Ventana modal.

```

<div class="content">
  <div class="col-sm-12 post animate-repeat"
    ng-animate="'animate'"
    ng-repeat="content in $contentlist.data
      |limitTo: $contentlist.perPage
      track by content.id" >
    <a class="image-box"
      ng-href="{{::content.url}}"
      ng-click="$contentlist.setVisit(content)"
      target="_blank">
      
    </a>
    <div id="title" class="title"><span ng-bind="content.title"></span></div>

```

Figura 38: Extracto de código. Cliente.
Visualización de sitio de origen en una nueva pestaña.

```

var service = {
  scrape: function scrape(options) {
    var siteDirname = getSiteDirname(options.url);
    var siteFullPath = getSiteFullPath(siteDirname);

    var scraperOptions = _.extend({}, defaults, {
      urls: [options.url],
      directory: siteFullPath,
      // If defaults object has request property, it will be superseded by options.request
      request: options.request
    });

    return scraper.scrape(scraperOptions).then(function() {
      return Promise.resolve(buildSiteObject(siteDirname));
    });
    return Promise.resolve(true);
  },

  getSite: function getSite(url) {
    var siteDirname = getSiteRawDirname(url);
    var dirname = siteDirname.domain+'-'+siteDirname.path;
    var timeNow = new Date().getTime();
    return getSitesDirectories().then(function (directories) {
      directories = _.sortBy(directories, ['directory']).reverse();
      var found = _.find(directories, function(el) {
        var odirname = String(el).substring(0,el.lastIndexOf('-'));
        var ocreatedAt = el.substring(el.lastIndexOf('-')+1) || new Date().getTime();
        return odirname === dirname && getDiffDays(ocreatedAt,timeNow) <= defaults.refresh ;
      });

      if (!found) {
        return service.scrape({url:url});
      }

      return Promise.resolve(buildSiteObject(found));
    });
  },

  getFullPath: function getFullPath(url) {
    var siteDirname = getSiteRawDirname(url);
    var dirname = siteDirname.domain+'-'+siteDirname.path;
    return getSitesDirectories().then(function (directories) {
      directories = _.sortBy(directories, ['directory']).reverse();
      var found = _.find(directories, function(el) {
        var odirname = String(el).substring(0,el.lastIndexOf('-'));
        return odirname === dirname;
      });

      if (!found) {
        return Promise.reject(getNotFoundError(dirname));
      }

      return Promise.resolve(getSiteFullPath(found));
    });
  }
};

```

Figura 39: Extracto de código. Servidor
Descargar sitio web de origen

```

function ScraperFactory($log,$resource){
    return $resource('/scraper?url=:url', {url:'@url'});
}

function ModalCtrl($scope,$log,$q,$sce,$timeout,$uibModalInstance,ScraperFactory,ModalBaseFactory,content){
    var $modal = $scope;

    $modal.content = content;
    $modal.currentUrl = $sce.trustAsResourceUrl(content.url.substr(content.url.indexOf('/:')+1));
    $modal.visit = 0;
    $modal.sitePath;

    $q.when(ScraperFactory.get({'url':content.url}).$promise,
        function(response){
            $modal.sitePath = $sce.trustAsResourceUrl('/website'+response.previewPath);
        },
        function(err){
            $log.warn(err);
            $modal.siteError = true;
        });

    $uibModalInstance.opened.then(function(){
        $modal.startTime = new Date();
        $timeout(function () {
            $modal.exit = true;
        }, 4000);
    });

    $uibModalInstance.closed.then(function(){
        $log.debug('- Close Modal');
        $uibModalInstance.dismiss('close');
    });

    $modal.diffTime = function(){
        var diff = (new Date() - $modal.startTime)/(1000); //segs
        return Math.round(diff);
    };

    $modal.close = function(){
        $log.debug($modal.diffTime());
        if($modal.diffTime() > 15)
        {
            $log.debug('Time:',$modal.diffTime());
            ModalBaseFactory.setVisit(content,$modal.diffTime());
            $uibModalInstance.close({visit:true,content:content.id});
            content.visits = content.visits + 1;
        }
        $uibModalInstance.dismiss('cancel');
    };
}
}

```

Figura 40: Extracto de código. Cliente Registrar «visita»

```

module.exports = {
    create: function (req,res) {
        var userID = req.user.id;
        var articleID = req.param('contentID');
        var visitTime = req.param('visitTime');

        Visit.create({user: userID,content: contentID,time: visitTime})
        .exec(function createCB(err, created){
            if(err)
                return sails.log.warn(err);

            return res.ok(created);
        });
    }
};

```

Figura 41: Extracto de código. Servidor Registrar «visita»

“Filtrado de contenidos” La filtración de contenidos que el sistema proporciona es a nivel de la capa de servidor y no desde la capa del cliente, siendo esto por la velocidad de selección y proyección de datos que se posee en el ambiente de servidor. Al momento de que el usuario opta por filtrar los datos de los contenidos de una forma determinada, se envía la solicitud de la filtración desde el cliente al servidor, se procesan el conjunto de datos existentes en el servidor de acuerdo a los parámetros de la solicitud enviada, se ordena el conjunto de resultados por orden de creación o en su caso de actualización y son visualizados en la área de contenidos del cliente.

Se poseen las siguientes opciones de filtración como:

- Contenidos recién publicados.
- Contenidos que son recomendados por el número total que se han realizado acciones como «visita», «me gusta» y «compartir».
- Contenidos que poseen mayor número de acciones de «me gusta».
- Contenidos que poseen mayor número de acciones de «compartir».
- Contenidos que pertenecen a una determinada categoría.
- Contenidos que pertenecen a un determinado autor.

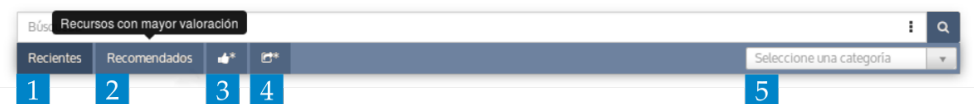


Figura 42: Interfaz. Opciones en componente de filtrado de contenidos.

1) Recientes. 2) Recomendado. 3) Con más acciones de “me gusta”. 4) Con más acciones de “compartir”. 5) Por categoría.

Por ejemplo cuando el usuario solicita la filtración de contenidos con el mayor número de acciones «me gusta». En la solicitud, el sistema en la capa del cliente realiza una petición Ajax al servicio web RESTful que proporciona la conexión con el servidor. Al llegar la petición en el servidor se procesa los parámetros de la paginación (límite de objetos e índice de consulta) y se retorna la lista de resultados ordenados de forma descendente.

```

| getContentListMostLiked : function (contentQuery){
  sails.log.debug('+ List of most liked elements:');
  return new Promise(function(resolve){
    delete contentQuery._criteria['limit'];
    contentQuery.then(function (contents){

      contents.sort(function(a, b) {
        return b.likes.length - a.likes.length;
      });

      contentQueryService.getContentListBase(contents)
        .then(function (response) {
          resolve(response);
        });

    });
  });
},

```

Figura 43: Extracto de código. Servidor
Contenidos con mayor número de acciones «me gusta»

```

.....
$contentlist.getContents = function(props){
  contentListFactory.getContents(props).then(function(response){
    $log.debug("+Get ContentList");
    var contentList = [];
    angular.forEach( response.data.results, function(content) {

      contentList.push(content);
    });

    $contentlist.setPagination(
      contentList,
      response.data.size,
      $contentlist.perPage
    );

  });
};
.....

$contentlist.setMostLikedList = function()
{
  $contentlist.props = {
    'kind': 'liked',
    'limit': $contentlist.perPage
  };
  $contentlist.getContents($contentlist.props);
  $contentlist.liked = true;
  $contentlist.normal = !$contentlist.liked;
  $contentlist.recommend = !$contentlist.liked;
  $contentlist.shared = !$contentlist.liked;
  $contentlist.currentPage = 1;
};

```

Figura 44: Extracto de código. Cliente
Contenidos con mayor número de acciones «me gusta»

“**Búsqueda de contenidos**” Se posee de las siguientes opciones para definir como parámetros de búsqueda, se pueden unir diversos parámetros con el propósito de mejorar la filtración de contenidos.

- Búsqueda de relación con palabras que conforman el nombre del autor.
- Búsqueda de relación con palabras que conforman el título.
- Búsqueda en la descripción de los contenidos.
- Búsqueda de contenidos que tengan relación con el parámetro dentro de su estructura.
- Búsqueda de contenidos que no tengan relación con el parámetro dentro de su estructura.
- Búsqueda de contenidos a partir de una fecha.

La búsqueda de contenidos a nivel del cliente se realiza mediante la sintaxis Sigma. El usuario interactúa con el componente, en el momento cuando realiza la acción de solicitar resultados se procesa los parámetros y se crea una sentencia en base a la sintaxis Sigma y la visualiza con el fin de representar de manera conjunta los campos y parámetros que el usuario ha definido. Para obtener los resultados el sistema construye a partir de la sentencia representada un objeto JSON en cual se envía los parámetros hacia el servidor, el cual retorna los contenidos que posean una relación con los parámetros definidos.

Tabla 11: Sintaxis de búsqueda Sigma.

Búsqueda con una palabra: [<i>parámetro: palabra</i>]
Búsqueda con dos o más palabras: [<i>parámetro: (palabra1 palabra2 ... palabraN)</i>]
<i>Parámetros</i>
title: Título
creator: Creador
description: Descripción
general: Contiene la palabra o las palabras definidas
not: No tiene relación con la palabra o palabras.

Se toma en cuenta con cuantos parámetros un contenido tiene relación para constar en la lista de los resultados a visualizarse, es decir para que un contenido cuente dentro de la lista debe cumplir con un índice de relación que por defecto es del 50%.

En el caso en que los parámetros de búsqueda no tengan ninguna relación con el conjunto de datos de los contenidos. Se retorna un mensaje de la no existencia

de resultados desde el servidor y el cliente al momento de recibir el mensaje como respuesta de la petición enviada , visualizará un mensaje de error al usuario. Figura 45

Tabla 12: Ejemplo de proceso de selección de resultados de búsqueda

Parámetro	P1	P2	P3	P4	Índice	Visualizar
Contenido 1	X	✓	✓	✓	0.75	True
Contenido 2	X	X	X	✓	0.25	False
Contenido 3	✓	✓	✓	✓	1	True
Contenido 4	X	✓	✓	X	0.5	True
Contenido 5	X	✓	X	X	0.25	False

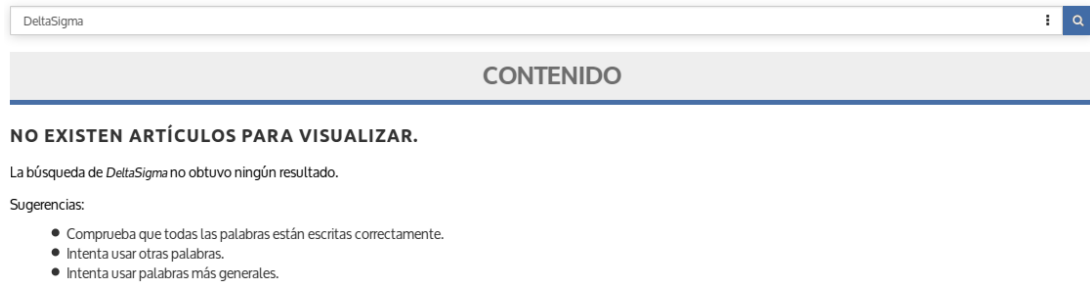


Figura 45: Mensaje de error en la de búsqueda de contenidos

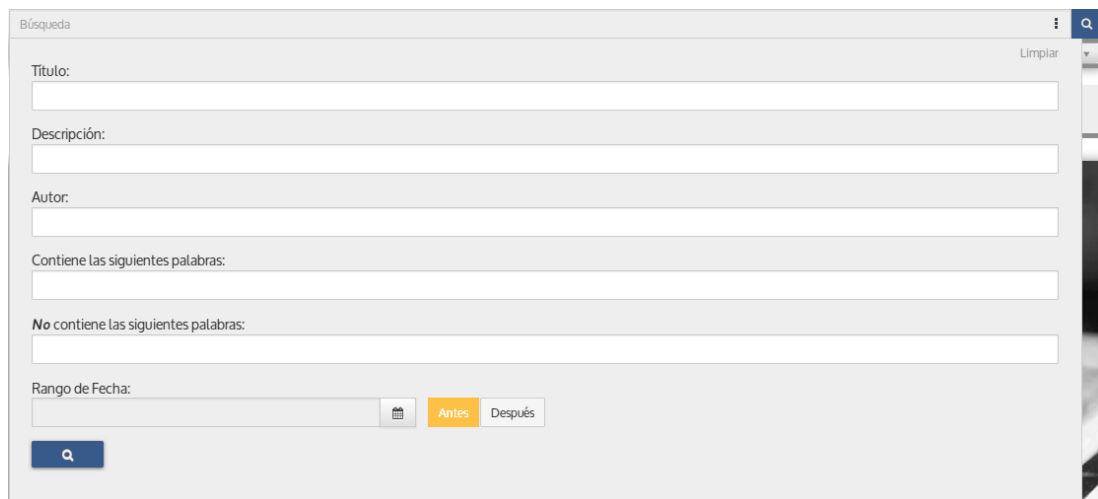


Figura 46: Interfaz de búsqueda de contenidos

4.4.2.3. Requisitos no funcionales

Usabilidad El sistema de información Sigma da un valor de importancia a la usabilidad dado que es el grado en el cual un producto puede ser utilizado por sus usuarios para lograr metas con efectividad, eficiencia y satisfacción en un determinado contexto de uso grado, lo cual es un factor de éxito considerable para un proyecto como para ser ignorado.

La usabilidad se encuentra relacionada íntimamente con la percepción del usuario respecto a la calidad del sistema; los algoritmos internos, procedimientos, o la definición de la arquitectura puede ser excelente y de un gran valor técnico, pero el usuario no tiene visibilidad de eso, sino que es la interfaz con la que interactúa y se relaciona. Es la usabilidad la cual se acerca a las fronteras del alcance de la ingeniería de software al relacionarse con la evaluación cognitiva y los modelos mentales respecto al usuario, tareas que se desarrollan en el área de la pedagogía y la psicología. [65][66]

Fuentes tipográficas En una composición gráfica, como en los sitios y aplicaciones web, los contenidos textuales son los principales responsables de la transmisión del mensaje al espectador final. Es la tipografía la cual hace de intermediaria entre el receptor y la información. Los caracteres matizan las palabras y le aportan o refuerzan el sentido. La mala selección de esas formas pueden interferir negativamente en la comunicación. Junto con el color, la tipografía puede alterar por completo el significado que asociamos a un diseño. El texto puede decir una cosa; las letras, otra muy diferente [67][68].

Fuentes compatibles Windows-Mac Os	
Windows	Mac
Arial	Helvetica
Courier New	Courier
MS Sans Serif	Geneva
MS Serif	New York
Times New Roman	Times

fuentes seguras

Con Windows Font Pack	
Windows	Mac
Arial	Arial
Comic Sans	Comic Sans
Trebuchet MS	Trebuchet MS
Verdana	Verdana
Georgia	Georgia

Figura 47: Fuentes tipográficas seguras

En la elección de las fuentes tipográficas para el sistema «Sigma» , aún a pesar de la recomendación existente en el diseño web de utilizar fuentes web seguras(Web Safe Fonts) en las que existe compatibilidad en la mayoría de sistemas operativos. En un acercamiento real no es posible acatar la recomendación de diseño web por limitantes como: la visualización de iconos, diseño, espacio entre caracteres, legibilidad a diferentes dimensiones.

Al elegir la utilización de fuentes tipográficas web diferentes a las que existen por defecto en los sistemas operativos , se debe optar por la incrustación de las fuentes tipográficas dentro del renderizado de las interfaces del sistema en el navegador. Factores que determinaron en la selección de las fuentes para el sistema fue la legibilidad y diseño que permitieran un mejor grado de visualización.[67]

Ubuntu	0123456789 ABCDFGHIJKL
Open Sans	0123456789 ABCDFGHIJKL
Oxygen	0123456789 ABCDFGHIJKL
Font Awesome	

Figura 48: Fuentes tipográficas del sistema Sigma

Para el proceso de la carga de fuentes se utiliza la biblioteca Web Font Loader [69], que permite tener mejoras en la carga de las fuentes en el ambiente del navegador y poder realizar la técnica de la *disposición de texto sin estilo*(*Flash of Unstyled Text - FOUT*), mediante esta técnica hasta la finalización de la carga de las fuentes por el navegador se visualizará por defecto el contenido con fuentes tipográficas seguras , evitando así la espera de la carga para el usuario ya que el navegador al no disponer de las fuentes no podrá visualizar aún los textos y contenidos [70][71].

Colores La paleta de colores para la utilización en las interfaces del sistema es construida a partir de la paleta base de Facebook. Con el propósito que la apariencia del sistema se ha con predominancia de las tonalidades de azules y grises con lo cual no se crea una desentono respecto a la apariencia de Facebook. Figura 49.

Con la paleta base del sistema Sigma, figura 50 se crean variaciones de tonalidades y se toman en cuenta los colores base al momento de combinar colores en los componentes de las interfaces como se visualiza en la figura 51. La importancia de la definición de una paleta de colores es la creación de interfaces agradables y uniformes para el usuario, en el que se da valor a la conjunción y armonía de colores evitando el resalte abrupto de componentes y la fatiga visual.

Color	Hex	RGB
	#3b5998	(59,89,152)
	#8b9dc3	(139,157,195)
	#dfe3ee	(223,227,238)
	#f7f7f7	(247,247,247)
	#ffffff	(255,255,255)

Figura 49: Paleta de colores de Facebook

Se determina a partir de la investigación de Kaplan en que el valor de los colores en un sitio web determina el grado de éxito que se tendrá en el momento del desarrollo de la relación con el usuario, si se utiliza colores que causan una disconformidad se producirán emociones y reacciones adversas respecto a la utilización del sistema [66][72].

Como por ejemplo podemos definir en base a los paradigmas sociales en el contexto del continente americano en que el color rojo tiene un significado de alerta, precaución o peligro ; si en un sistema tenemos una paleta de colores definido para las interfaces con una predominancia a las tonalidades del color rojo se tendrá como resultado que los usuarios estarán en un estado de alerta ,creando fatiga visual dificultando la creación de la relación e interacción con respecto al sistema [73][74].

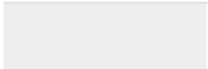
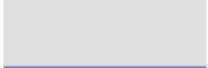



Color	Hex	RGB
	#e0e0e0	(238,238,238)
	#e0e0e0	(224,224,224)
	#6d84b4	(109,132,180)
	#3b5998	(59,89,152)
	#595959	(89,89,89)

Figura 50: Paleta de colores del sistema Sigma.



Figura 51: Colores en la interfaz del sistema Sigma.

Diseño El diseño del sistema es hacia un enfoque minimalista, al no disponer de un múltiple de acciones y de información no resta al propósito del sistema. Cada información, componente, elemento extra compite con la información y elementos necesarios, haciendo que se disminuya el grado de visualización.[75] La información y componentes que ayude en la experiencia del usuario es a la que se le da un mayor grado de visualización en el sistema de información Sigma.

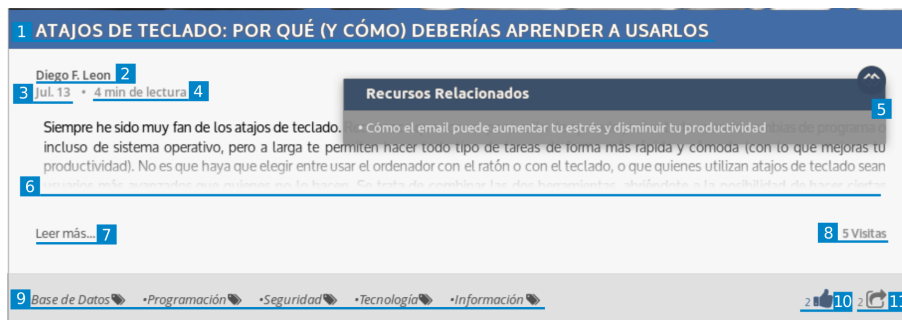


Figura 52: Elementos de los contenidos del sistema Sigma.

- 1) Título.
- 2) Autor de contenido.
- 3) Fecha.
- 4) Tiempo de lectura.
- 5) Contenidos relacionados.
- 6) Descripción.
- 7) Visualizar sitio web de origen.
- 8) Número de Visitas.
- 9) Categorías que se relacionan.
- 10) Me gusta.
- 11) Compartir.

Se detalla a continuación la información de mayor relevancia en los contenidos:

- Tiempo de lectura: Indica el tiempo en promedio que se toma en leer un contenido se basa en la métrica de 250 palabras por minuto.
- Descripción: Información presentada referente al contenido.
- Contenidos Relacionados: Contenidos que comparte una relación.

- Categorización: Abstracciones referente a la información del contenido.
- Visitas: Número de veces que se ha solicitado la visualización de un contenido.
- “Me gusta”: Número de veces en las que los usuarios han indicado que les gusta un contenido.
- “Compartir”: Número de veces que un contenido ha sido compartido por el usuario dentro de los diferentes contextos de Facebook.

Se definió en el diseño del sistema Sigma utilizar retículas para disponer de consistencia en las estructura y proporciones en los contenidos. Es de utilidad utilizar en el diseño la retícula (*grid* en inglés), para asegurarse de poseer un equilibrio visual en la composición. La retícula es una estructura conformada de dos dimensiones construida a partir de líneas verticales y horizontales utilizadas para dar forma y estructura al contenido, sirve como una base para organizar textos, videos o imágenes de una forma que sea fácil de visualizar y que se mantenga un orden racional [76][77]

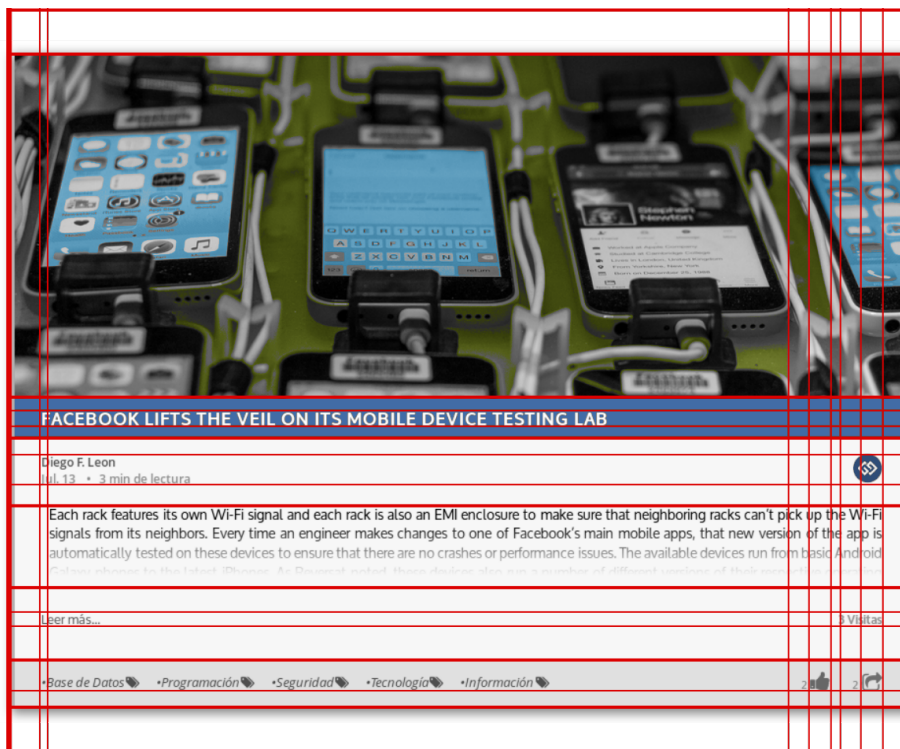


Figura 53: Retícula aplicada a los contenidos del sistema Sigma

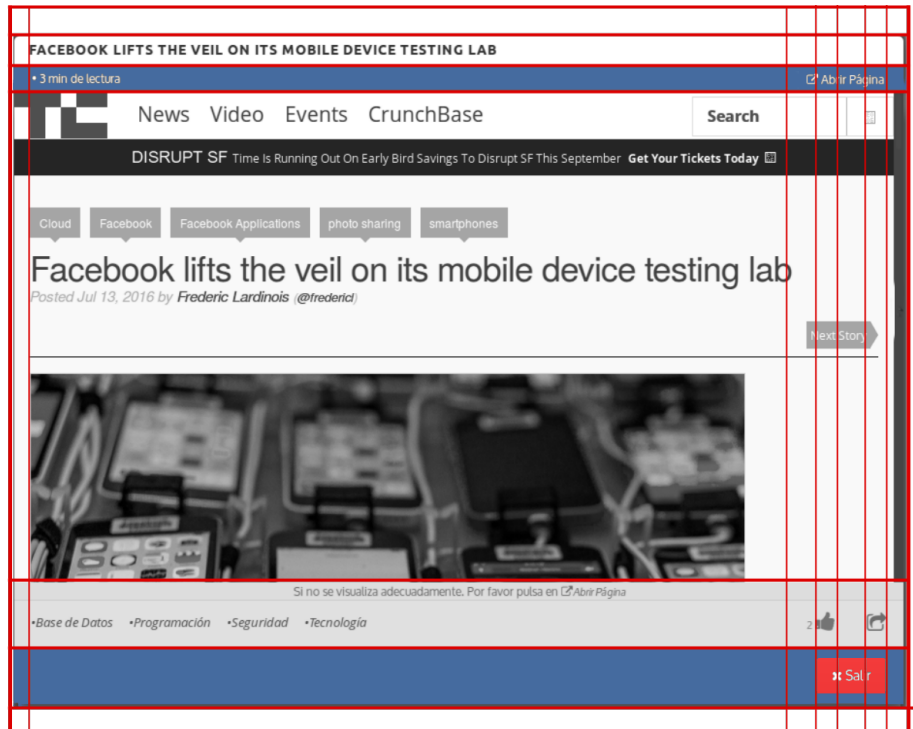


Figura 54: Retícula aplicada en la ventana de visualización de contenidos.

Mantenibilidad En entornos de ejecución para Node.js se difiere dos ámbitos como son de producción (sistemas estables) y de desarrollo (sistema en construcción y en pruebas).

Para iniciar una instancia de un servidor de Node.js, se debe llamar al servicio de ejecución e indicar el nombre del script que se encargará de iniciar el sistema y los servicios necesarios. Respecto a esto existe la limitante que la ejecución del servidor se parará inmediatamente cuando se encuentre con una excepción no controlada. En entornos de desarrollo no involucraría problemas, dado que es parte del proceso de construcción.

Respecto a entornos de producción en los cuales es necesario asegurar la mantenibilidad y funcionamiento de los sistemas, esto lleva a determinar la utilización de otras herramientas con las que se pueda disponer de un control en el despliegue y monitoreo de los procesos en entornos de producción de Node.js. Se visualiza en la figura las opciones con las que se cuenta para el control y monitoreo de sistemas. [78]

Se determina la utilización de PM2 para la gestión de procesos de despliegue, ejecución y monitoreo, se escoge a PM2 por ser una herramienta con evidencia de utilización en el ámbito empresarial. Al ser la empresa KeyMetrics quienes son los contribuidores y administradores principales del proyecto pm2, se cuenta con confiabilidad y acceso a documentación técnica. [79][80]

Tabla 13: Cuadro comparativo entre gestores de procesos en Node.js

Característica	Forever	PM2	StrongLoop
Reinicio en caso de fallo	✓	✓	✓
Inicio con el sistema		✓	✓
Registro de logs		✓	✓
Soporte múltiples aplicaciones		✓	✓
Monitoreo desde terminal		✓	✓
API		✓	✓
Clusterización		✓	✓
Configuración con JSON		✓	✓
Despliegue en contenedores Docker			✓

La interfaz que provee pm2 es de gran utilidad dado que permite acceso a la información relevante como es consumo de memoria, archivos de registros, estado de procesos, lista de procesos, esto permite monitorear todo lo que conlleva la ejecución del sistema.

```
~/D/master-sigma >>> pm2 status
```

App name	id	mode	pid	status	restart	uptime	memory	watching
Sigma	0	fork	616	online	0	12m	189.723 MB	disabled

Figura 55: Estado de la ejecución del sistema Sigma en pm2

```
~/D/master-sigma >>> pm2 show Sigma
Describing process with id 0 - name Sigma
```

status	online
name	Sigma
restarts	0
uptime	13m
script path	/home/dlfinis/DevProjects/master-sigma/app.js
script args	--prod
error log path	/home/dlfinis/.pm2/logs/Sigma-error-0.log
out log path	/home/dlfinis/.pm2/logs/Sigma-out-0.log
pid path	/home/dlfinis/.pm2/pids/Sigma-0.pid
interpreter	node
interpreter args	N/A
script id	0
exec cwd	/home/dlfinis/DevProjects/master-sigma
exec mode	fork_mode
node.js version	5.12.0
watch & reload	X
unstable restarts	0
created at	2016-08-18T23:39:49.739Z

```
Revision control metadata
```

revision control	git
remote url	git@github.com:dlfinis/master-sigma.git
repository root	/home/dlfinis/DevProjects/master-sigma
last update	2016-08-18T23:52:27.539Z
revision	4f557624b4034058cec4b6ecc0774e0b0877c92a
comment	Update info
branch	development

```
Code metrics value
```

Loop delay	0.5ms
------------	-------

Figura 56: Detalles de la ejecución del sistema Sigma en pm2

Diseño de interfaz adaptativa El diseño adaptativo es uno de los estándares que se caracteriza de gran valor en el desarrollo de aplicaciones y sitios web siendo por el valor de la visualización en diferentes dimensiones , adaptándose en el contexto que se encuentre si es en el navegador de un computador de escritorio, en el navegador de una tableta inteligente o en el navegador de un teléfono inteligente. El enfoque al diseño adaptativo se debe a las posibilidades de mejorar la experiencia e interacción del usuario [81][82].

En el sistema Sigma se utiliza como base en el desarrollo de interfaces adaptativas a Bootstrap [51], dado que es un marco de desarrollo que ya contempla varios de los aspectos y elementos necesarios en los procesos de diseño, permitiendo un ahorro en recursos e implementaciones.



Figura 57: Interfaz del sistema Sigma a 1280x890



Figura 58: Interfaz del sistema Sigma a 640x960

Seguridad Para la ejecución en el ambiente de aplicaciones de Facebook , se solicita tener navegación mediante el protocolo HTTPS. No se permite la utilización del protocolo HTTP, esto debido a las políticas de Facebook dado que se esta realizando una incrustación de contenido externo dentro de la plataforma.

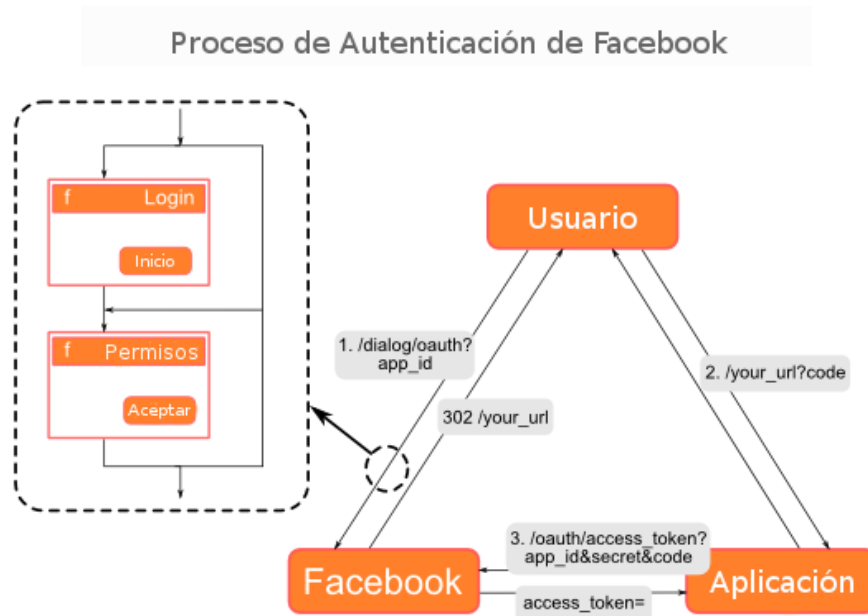


Figura 59: Autenticación OAuth de Facebook [7]

Facebook utiliza autenticación por token el cual es almacenado en una cookie en el cliente, el sistema «Sigma» basa su proceso de autenticación en el proceso de Facebook, siendo la primera instancia en el proceso de validación y control. Procesos esenciales de seguridad dentro del sistema son:

- **El usuario no ha otorgado los permisos:** Esto es posible si el usuario ingresa por primera vez o en el caso de que borre el acceso a los permisos. Cuando se solicite acceso el sistema solicitará a Facebook la visualización de una ventana modal para la solicitud de permisos al usuario.
- **La sesión del usuario ha caducado :** Se realizará una solicitud de acceso a Facebook, se almacena la respuesta del token de acceso en la cookie del usuario. Permitiendo continuar con el proceso de acceso.
- **La sesión del usuario esta disponible :** Cuando el tiempo de vida de la sesión aún no ha terminado, se permite seguir el proceso de acceso.

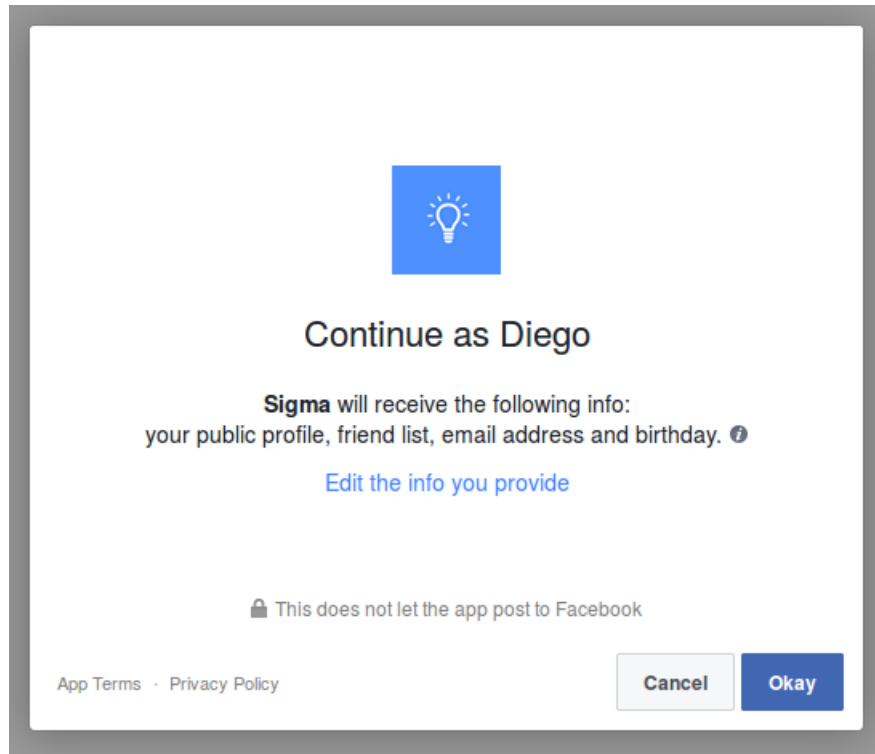


Figura 60: Ventana modal de acceso a permisos a la información del usuario.

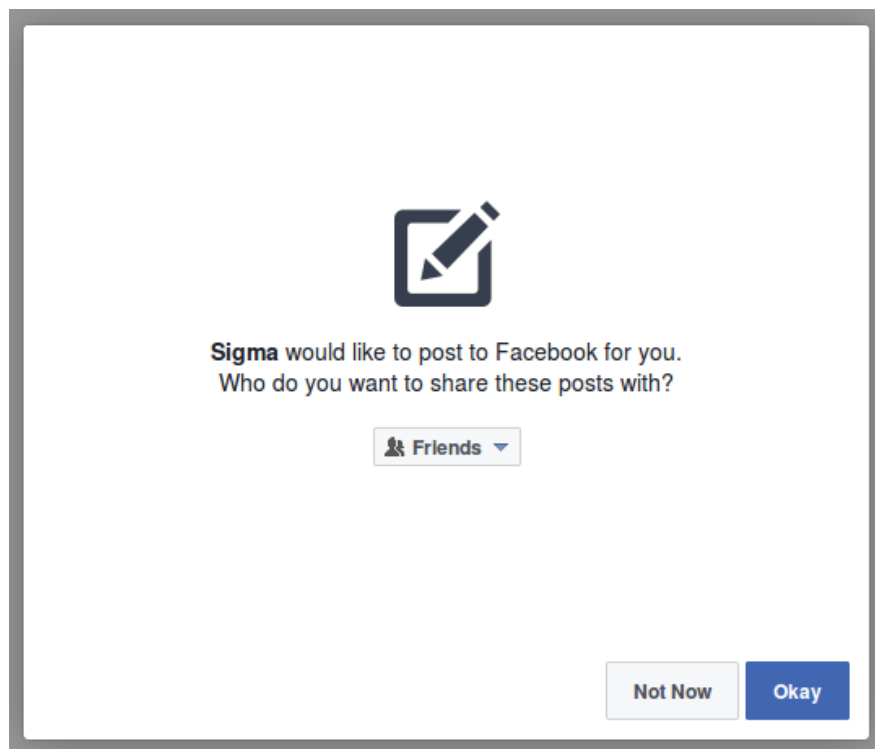


Figura 61: Ventana modal de aceptación de permisos para publicar en nombre del usuario.

Rendimiento Los sitios y las aplicaciones web llegan a tener niveles de retardo que pueden ser significativos en el lado del cliente dado por el tiempo de respuesta y la disponibilidad del servidor. El problema está asociado al número de peticiones y la cantidad de datos a ser transferidos, siendo procesos como la concatenación una ayuda con lo primero y la minimización ayuda con lo segundo [83][84].

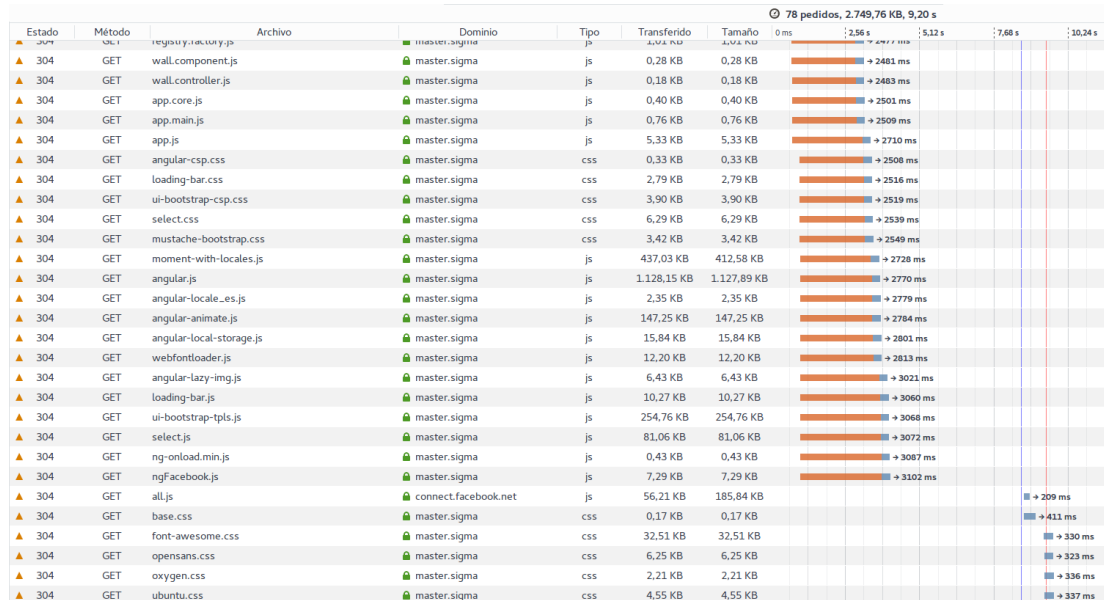


Figura 62: Carga de recursos en entorno de desarrollo.

Respecto a las solicitudes de recursos esenciales para la visualización como son los archivos css y javascript se implementan mejoras en el rendimiento del despliegue de estos. En la siguiente figura se puede observar el resultado de la utilización de la concatenación y minimización de archivos.

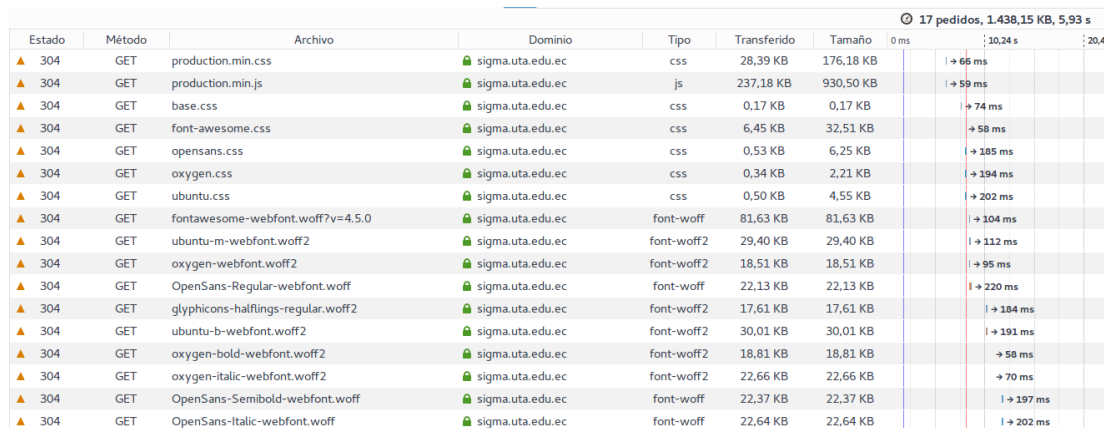


Figura 63: Carga de recursos en entorno de producción.

Concatenación Es el proceso por el cual se unen determinados archivos dando como resultado un solo archivo con identificadores y separadores de acuerdo al

tipo de archivo. Esto permite evitar la múltiple solicitud de archivos al servidor [83].

Minimización Proceso por el cual se remueven todos los caracteres innecesarios del código fuente sin cambiar su funcionalidad. Estos caracteres innecesarios usualmente incluyen espacios en blanco, caracteres de nueva línea, comentarios, y a algunas veces delimitadores de bloque, cuales son usados para mejorar la visibilidad del código pero no son requerido para la ejecución [83].

La minificación es especialmente utilizada para los lenguajes interpretados utilizados en el Internet -en este caso Javascript-, a causa de esto se reduce la cantidad de datos ha ser transferidos en las solicitudes de cada archivo al servidor.

En la minificación de Javascript es posible la creación de «mapas de recursos»(source maps) lo cual permite al código ser leído y depurable aún a pesar de la concatenación y la minificación [83][85].

4.4.3. Transición

4.4.3.1. Pruebas de funcionalidad

Las pruebas de funcionalidad tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema. Por este motivo, estas pruebas son realizadas por el usuario final que, durante este período de tiempo, debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

Para determinar mejoras y recomendaciones respecto a la usabilidad e interacción del sistema con el usuario, así como para comprobar la mantenibilidad y seguridad del sistema, se realizaron pruebas de funcionalidad con un grupo de usuarios potenciales. Se utilizó la encuesta como instrumento de recolección de información (**Anexo F**); esta se aplicó cuando el sistema entró en la primera fase estable de producción.

En el **Anexo F** se analiza la información que se obtuvo en las pruebas de funcionalidad. Mediante gráficos estadísticos se muestra que la aceptación y valoración por parte de los usuarios está en un nivel admisible; a partir de esto se llevó a cabo un proceso de refinamiento de los procesos e interfaces del sistema.

En la evaluación llevada a cabo se evidenció la falta de manuales de ayuda y documentación, como uno de los principios de usabilidad con evaluación menos favorable. A partir de ello, se tomaron medidas para mejorar la ayuda al usuario y se redactó el manual de usuario (**Anexo H**), que constituye una aportación

adicional a la documentación de la aplicación para el usuario.

Para medir el tiempo de reacción y funcionamiento del sistema en un ambiente controlado se diseñó un experimento en el que usuarios potenciales de la aplicación midieron con un cronómetro el tiempo que demoró el sistema en procesar un conjunto predefinido de solicitudes de procesamiento.

En los resultados del **Anexo G** se puede constatar el tiempo empleado en la realización de las acciones.

A partir de ello, se determinó qué procesos se encontraban ralentizando la interfaz del sistema y la experiencia del usuario. Se tomaron medidas de mejora en aspectos como la usabilidad de las acciones sociales y del proceso de carga asincrónica de datos para la visualización de los contenidos.

Como primera opción se decidió que para la visualización en la ventana modal se realizaría un llamado al sitio web de origen. La problemática se da en que no todos los sitios web soportan navegación por HTTPS, lo cual constituye un requisito para cualquier consumo de datos desde una aplicación Facebook.

El sistema solventaba, en primera instancia, detectando si era posible el acceso por HTTPS. Este proceso de verificación y visualización le tomaba cierto tiempo al sistema, algo que se evidenció en los resultados del **Anexo G**, además de que la notificación de que el sitio no podía ser consumido directamente desde la aplicación afectaba la usabilidad de la misma. En base a esto, el proceso se cambió de hacer una llamado al sitio a ser en cambio el sitio descargado a cual el contenido hace referencia, evitando así una espera innecesaria.

Una vez mejorados los procesos, en base a las pruebas y recomendaciones de los usuarios, actualizado el sistema a la versión estable final, con las respectivas pruebas por parte de los usuarios finales, las mismas que fueron exitosas conforme a las aspiraciones de los mencionados.

4.4.3.2. Puesta en producción

También denominada esta fase como implantación, es la última etapa del desarrollo de un software. Para que un software se ponga en producción debe estar correctamente probado y los prototipos aceptados por los usuarios.

La puesta en producción indica que el software se encuentra en estado óptimo para la utilización de los usuarios en un ambiente real.

Para la puesta en producción de la aplicación en cuestión se siguieron los

siguientes pasos:

- Configuración de la red.
- Instalación de gestor de base de datos.
- Instalación de requisitos del sistema.
- Configuración del entorno de la plataforma NodeJS.
 - Adquisición de bibliotecas para el entorno de «back-end» con **npm**.
 - Adquisición de bibliotecas para el entorno de «front-end» con **bower**.
 - Ejecución de tareas de configuración de entorno de producción con **GruntJS**.
- Ejecución del sistema.
- Monitorización de la ejecución del sistema.

4.4.3.3. Estabilización

La entrega del proyecto final se hace conjuntamente con un manual de usuario en el **Anexo H** para la documentación de las funcionalidades del sistema y de un manual técnico en el **Anexo I**, que detalla el proceso del despliegue del sistema en ambientes de producción.

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- Al llevar a cabo el presente trabajo se comprobó que el lenguaje de programación Javascript provee de un marco de desarrollo estable que unifica el lenguaje de programación en las capas cliente y servidor.
- El desarrollo de la aplicación con Javascript permitió obtener una solución creada con software libre y multiplataforma.
- La experiencia de uso del marco de desarrollo Sails.js ha sido satisfactoria; supone una ventaja en la simplificación de la organización y la administración del código respecto al desarrollo en un ambiente como Node.js. Aunque este ya proporciona, por sí mismo, un adecuado nivel de abstracción del servidor web y otros componentes, sin el soporte de Sails.js, el nivel de complejidad aumenta, por lo cual es necesario una mayor experiencia y conocimientos en determinados componentes y procesos requeridos en la construcción de una solución de software.
- Una aplicación web con una arquitectura SPA puede equipararse en funcionalidad como si se tratará de una aplicación tradicional de escritorio. El tiempo de respuesta es menor que el de una aplicación web tradicional puesto que se actualiza solo una parte del contenido de la página web y no se refresca todos los componentes del despliegue necesarios para la visualización de la aplicación.
- Al utilizar la arquitectura de servicios web RESTful se garantiza la independencia de la programación del servidor respecto a la del cliente, lo que permite la evolución y la refactorización.
- Las herramientas y ayuda técnica que aprovisiona la plataforma de Facebook permite un acceso de una manera más sencilla y eficaz en la interacción y manipulación de la información de los usuarios, siendo esto un beneficio al disminuir el tiempo de desarrollo y de implementación.
- Al registrar las acciones sociales sobre los contenidos académicos publicados, es posible obtener información útil sobre la cual evaluar la relevancia de los

contenidos y realizar ajustes en base a la retro-alimentación activa de los usuarios.

- El sistema de información «Sigma» desarrollado para la FISEI constituye una propuesta que permite la integración de los entornos de redes sociales al proceso de aprendizaje que tiene lugar en las instituciones educativas. El impacto potencial de su uso en la comunidad universitaria es mayor por el uso de código abierto y gratuito en su implementación.

5.2. Recomendaciones

- Se sugiere la investigación en cuanto a la implementación de mecanismos para la curación de contenidos , que utilizando componentes de inteligencia artificial de un agente de software se determine la relevancia de los diversos contenidos a partir de la información que es procesada en el sistema «Sigma».
- En futuros trabajos de investigación se recomienda el estudio del rendimiento y escalabilidad del lenguaje de programación Javascript en entornos de producción.
- Como trabajos de análisis a posterior se recomienda el estudio del las capacidades del ORM Waterline para gestionar grandes volúmenes de datos, específicamente en la selección y proyección de datos en ambientes de peticiones concurrentes.

Bibliografía

- [1] B. Explained, “Intermediate Rails: Understanding Models, Views and Controllers.” [Web; accedida el 18-10-2016].
- [2] C. de la Torre Llorente, U. Z. Castro, M. A. R. Barros, and J. C. Nelson, “Guía de Arquitectura N-Capas orientada al Dominio con .NET,” 2010.
- [3] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
- [4] LittleQ, The Department of Computer Science, NCCU, “Introduction to Facebook Python API,” 2011.
- [5] G. Fink and I. Flatow, “Introducing Single Page Applications,” in *Pro Single Page Application Development*, pp. 3–13, Springer, 2014.
- [6] M. S. Mikowski and J. C. Powell, “Single Page Web Applications,” *B and W*, 2013.
- [7] <https://canvas.facebook.com/>. [Web; accedida el 10-05-2016].
- [8] R. M. Ruiz, F. C. López, and J. Millán, “Experimentando con las redes sociales en la enseñanza universitaria en ciencias,” *Revista Eureka sobre enseñanza y divulgación de las ciencias*, vol. 10, no. 3, pp. 394–405, 2013.
- [9] S. R. H. Requena, “El modelo constructivista con las nuevas tecnologías, aplicado en el proceso de aprendizaje,” *RUSC. Universities and Knowledge Society Journal*, vol. 5, no. 2, p. 6, 2008.
- [10] Á. J. Domínguez Daniel, “Redes sociales y espacios universitarios,” *RUSC, Revista de Universidad y Sociedad del Conocimiento*, vol. 9, no. 1, pp. 51–64, 2012.
- [11] Facebook.Inc, “Facebook,” 2015. [Web; accedida el 15-08-2016].
- [12] Moodle, “Moodle,” 2016. [Web; accedida el 15-08-2016].
- [13] LinkedIn, “LinkedIn,” 2016. [Web; accedida el 15-08-2016].
- [14] F. Esteve, “Bolonia y las TIC: de la docencia 1.0 al aprendizaje 2.0,” *La cuestión universitaria*, vol. 5, pp. 59–68, 2009.

- [15] A. G. Sans, “Las redes sociales como herramientas para el aprendizaje colaborativo: una experiencia con Facebook,” *Re-Presentaciones: Periodismo, Comunicación y Sociedad*, no. 5, pp. 48–63, 2009.
- [16] Q. Martín-Moreno, “Aprendizaje colaborativo y redes de conocimiento,” *Actas de las IX Jornadas Andaluzas de Organización y Dirección de Instituciones Educativas*, pp. 55–70, 2004.
- [17] G. Nández and Á. Borrego, “Use of social networks for academic purposes: a case study,” *The Electronic Library*, vol. 31, no. 6, pp. 781–791, 2013.
- [18] C.-J. Chen and S.-Y. Wu, “A Case Study Exploring Junior High School Students’ Interaction Behavior in a Learning Community on Facebook: Day and Time,” *International Journal of Learning, Teaching and Educational Research*, vol. 12, no. 2, 2015.
- [19] V. M. Díaz and J. Cabero-Almenara, “Percepciones de los estudiantes universitarios latinoamericanos sobre las redes sociales y el trabajo en grupo,” *RUSC. Universities and Knowledge Society Journal*, vol. 10, no. 2, pp. 219–235, 2013.
- [20] M. G. Aguilar, S. R. Campos, and P. F. Batlle, “El uso académico de las redes sociales en universitarios,” *Comunicar*, vol. 19, no. 38, pp. 131–138, 2012.
- [21] M. Hayes, K. van Stolk-Cooke, and F. Muench, “Understanding Facebook use and the psychological affects of use across generations,” *Computers in Human Behavior*, vol. 49, pp. 507–511, 2015.
- [22] M. Castells, *The rise of the network society: The information age: Economy, society, and culture*, vol. 1. John Wiley & Sons, 2011.
- [23] R. Judele, D. Tsovaltzi, T. Puhl, and A. Weinberger, “Collaborative learning in Facebook: Adverse effects of individual preparation,” in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 1616–1624, IEEE, 2014.
- [24] D. Tsovaltzi, R. Judele, T. Puhl, and A. Weinberger, “Scripts, individual preparation and group awareness support in the service of learning in Facebook: How does CSCL compare to social networking sites?,” *Computers in Human Behavior*, vol. 53, pp. 577–592, 2015.

- [25] P. A. Kirschner, “Facebook as learning platform: Argumentation superhighway or dead-end street?,” *Computers in Human Behavior*, vol. 53, pp. 621–625, 2015.
- [26] T. O’Reilly, “Web 2.0: compact definition,” *Message posted to http://radar.oreilly.com/archives/2005/10/web_20_compact_definition.html*, 2005.
- [27] R. V. Argüelles, “Las redes sociales y su aplicación en la educación,” *Revista Digital Universitaria UNAM*, vol. 14, no. 4, 2013.
- [28] L. Codina, “¿ web 2.0, web 3.0 o web semántica?: El impacto en los sistemas de información de la web,” in *Congreso Internacional de Ciberperiodismo y Web*, vol. 2, 2009.
- [29] I. Küster and A. Hernández, “De la web 2.0 a la web 3.0: antecedentes y consecuencias de la actitud e intención de uso de las redes sociales en la web semántica,” *Universia Business Review*, vol. 1, no. 37, 2013.
- [30] F. B. Viégas, D. Boyd, D. H. Nguyen, J. Potter, and J. Donath, “Digital artifacts for remembering and storytelling: Posthistory and social network fragments,” in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pp. 10–pp, IEEE, 2004.
- [31] K. C. Laudon and J. P. Laudon, *Sistemas de información gerencial: administración de la empresa digital*. Pearson Educación, 2004.
- [32] C. Ciborra, *The labyrinths of information: Challenging the wisdom of systems: Challenging the wisdom of systems*. OUP Oxford, 2002.
- [33] J. A. Senn, *Análisis y Diseño de sistemas*. Mc Graw Hill, México, 1996.
- [34] A. Silberschatz, H. F. Korth, S. Sudarshan, *et al.*, *Database system concepts*, vol. 4. McGraw-Hill Singapore, 1997.
- [35] R. Barzanallana, “Aplicación Web,” 2012. [Web; accedida el 15-08-2016].
- [36] O. U. Press, “Aplicación,” 2016. [Web; accedida el 11-01-2016].
- [37] S. L. Mora, *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario, 2002.
- [38] W3C, “Relationship to the World Wide Web and REST Architectures.” [Web; accedida el 10-12-2015].
- [39] W3C, “Web Service Glossary.” [Web; accedida el 10-12-2015].

- [40] J. J. Gutiérrez, “¿ Qué es un framework web?,” *Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf* Accessed May, vol. 12, 2014.
- [41] N. Jain, P. Mangal, and D. Mehta, “AngularJS: A Modern MVC Framework in JavaScript,” *Journal of Global Research in Computer Science*, vol. 5, no. 12, pp. 17–23, 2015.
- [42] M. Masse, *REST API design rulebook*. O’Reilly Media, Inc., 2011.
- [43] N. B. Ellison, C. Steinfield, and C. Lampe, “The benefits of Facebook “friends:” Social capital and college students’ use of online social network sites,” *Journal of Computer-Mediated Communication*, vol. 12, no. 4, pp. 1143–1168, 2007.
- [44] Facebook.Inc, “Graph API Facebook,” 2016. [Web; accedida el 15-08-2016].
- [45] Node.js, “Node.js,” 2015. [Web; accedida el 10-12-2015].
- [46] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, *Node.js in Action*. Manning, 2014.
- [47] J. Bermúdez-Ortega, E. Besada-Portas, J. López-Orozco, J. Bonache-Seco, and J. de la Cruz, “Remote Web-based Control Laboratory for Mobile Devices based on EJS, Raspberry Pi and Node.js,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 158–163, 2015.
- [48] SailsJS, “Sail.js,” 2015. [Web; accedida el 10-12-2015].
- [49] A. Mardan, “Sails.js, DerbyJS, LoopBack, and Other Frameworks,” in *Pro Express.js*, pp. 205–214, Springer, 2014.
- [50] T. Inc, “Bootstrap.” [Web; accedida el 10-05-2016].
- [51] T. Firdaus, *Responsive Web Design by Example Beginner’s Guide*. Packt Publishing Ltd, 2013.
- [52] G. Inc., “AngularJS,” 2016. [Web; accedida el 11-08-2016].
- [53] M. García Salinero, “Grid de datos desarrollo con AngularJS utilizando el patrón Modelo Vista-Controlador,” 2014.
- [54] P. Kozlowski, *Mastering Web Application Development with AngularJS*. Packt Publishing Ltd, 2013.

- [55] G. Inc., “Documentation AngularJS,” 2016. [Web; accedida el 11-08-2016].
- [56] J. Nielsen, “Usability Introduction,” 2012. [Web; accedida el 15-08-2016].
- [57] J. Nielsen, “Ten usability heuristics,” 2005. [Web; accedida el 15-08-2016].
- [58] J. Heumann, “Introduction to business modeling using the Unified Modeling Language (UML),” *Rational Software*, 2001.
- [59] H.-E. Eriksson and M. Penker, “Business modeling with UML,” *Business patterns at work*, 2000.
- [60] I. Grigorik, “Latency: The new Web performance bottleneck,” *Retrieved April*, vol. 24, p. 2013, 2012.
- [61] B. Mallick and N. Das, “An Approach to Extended Class Diagram Model of UML for Object Oriented Software Design,” *International Journal of Innovative Technology & Adaptive Management (IJITAM)*, vol. 1, no. 2, 2013.
- [62] <https://canvas.facebook.com/>. [Web; accedida el 10-05-2016].
- [63] <https://developers.facebook.com/docs/opengraph/guides/og.likes>. [Web; accedida el 10-05-2016].
- [64] <https://developers.facebook.com/docs/sharing/reference/share-dialog>. [Web; accedida el 10-05-2016].
- [65] B. Boehm, “A spiral model of software development and enhancement,” *ACM SIGSOFT Software Engineering Notes*, vol. 11, no. 4, pp. 14–24, 1986.
- [66] L. Brady and C. Phillips, “Aesthetics and usability: A look at color and balance,” *Usability News*, vol. 5, no. 1, pp. 2–5, 2003.
- [67] J. Pamental, *Responsive Typography: Using Type Well on the Web*. O’Reilly Media, Inc., 2014.
- [68] S. E. Meyer, *Designing With Type*. New York, 1980.
- [69] TypeKit, “Web Font Loader.” [Web; accedida el 10-05-2016].
- [70] R. Clark, O. Studholme, C. Murphy, and D. Manian, “Improving Web Typography,” in *Beginning HTML5 and CSS3*, pp. 397–433, Springer, 2012.
- [71] B. Tiberi, “FONT LOADING SYSTEM AND METHOD,” Jan. 21 2016. US Patent 20,160,019,189.

- [72] A. Y. KAPLAN, J.-G. Byeon, J.-J. Lim, K.-S. Jin, B.-W. Park, and S. U. TARASOVA, “Unconscious operant conditioning in the paradigm of brain-computer interface based on color perception,” *International journal of neuroscience*, vol. 115, no. 6, pp. 781–802, 2005.
- [73] J. L. Caivano and M. López, “Retórica del negro, blanco y rojo,”
- [74] Z. Pousman and J. Stasko, “A taxonomy of ambient information systems: four patterns of design,” in *Proceedings of the working conference on Advanced visual interfaces*, pp. 67–74, ACM, 2006.
- [75] A. Arpetti, M. C. C. Baranauskas, and T. Leo, “Making design easy: A usability evaluation of latest generation learning design tools,” in *World conference on educational multimedia, hypermedia and telecommunications*, pp. 960–965, 2013.
- [76] J. Müller-Brockmann, *Grid systems in graphic design: a visual communication manual for graphic designers, typographers and three dimensional designers*. Arthur Niggli, 1996.
- [77] P. J. Lynch, *Web style guide*. Yale University Press, 2008.
- [78] StrongLoop, “Compare with pm2 and forever.” [Web; accedida el 10-05-2016].
- [79] K. Inc, “PM2.” [Web; accedida el 10-05-2016].
- [80] K. Inc, “PM2 Repository.” [Web; accedida el 10-05-2016].
- [81] F. X. Li, S. Z. Hai, and L. J. Feng, “Responsive Web Design Based on Different-Sized Devices,” 2015.
- [82] B. Kim, “Responsive web design, discoverability, and mobile challenge,” *Library technology reports*, vol. 49, no. 6, pp. 29–39, 2013.
- [83] E. H. Mjelde, “Performance as design-Techniques for making websites more responsive,” 2014.
- [84] J. R. Lewis and M. Moscovitz, “Optimizing CSS for Performance,” *AdvancED CSS*, pp. 275–289, 2009.
- [85] S. Souders, “High-performance web sites,” *Communications of the ACM*, vol. 51, no. 12, pp. 36–41, 2008.

Anexos y Apéndices

Anexo A

Informe de peticiones DNS

A.1. Información de registro

Fecha: 20 de Noviembre del 2015

Tiempo de Inicio: 14:22:18

Tiempo de Finalización: 16:01:52

Tamaño de archivo(MB): 101

Número de registros: 938332

A.2. Análisis de datos

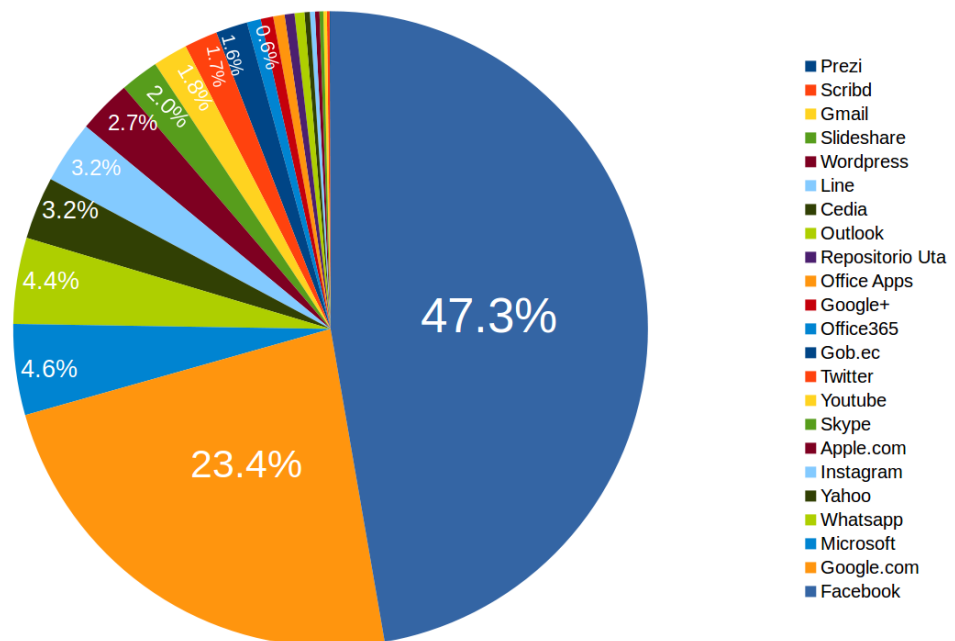


Figura 1: Tráfico hacia los sitios web con mayor predominancia en la Universidad Técnica de Ambato.

Tabla 1: Datos del tráfico web en la Universidad Técnica de Ambato.

Sitios	#Peticiones	%
Prezi	189	0.05 %
Scribd	586	0.14 %
Gmail	756	0.18 %
Slideshare	808	0.19 %
Wordpress	948	0.23 %
Line	1072	0.26 %
Cedia	1146	0.27 %
Outlook	2081	0.50 %
Repositorio UTA	2100	0.50 %
Office Apps	2417	0.58 %
Google+	2676	0.64 %
Office365	2948	0.71 %
Gob.ec	6733	1.61 %
Twitter	7065	1.69 %
Youtube	7382	1.77 %
Skype	8175	1.96 %
Apple.com	11386	2.73 %
Instagram	13271	3.18 %
Yahoo	13292	3.18 %
Whatsapp	18281	4.38 %
Microsoft	19310	4.63 %
Google.com	97470	23.35 %
Facebook	197311	47.27 %
Total	475465	100.00 %

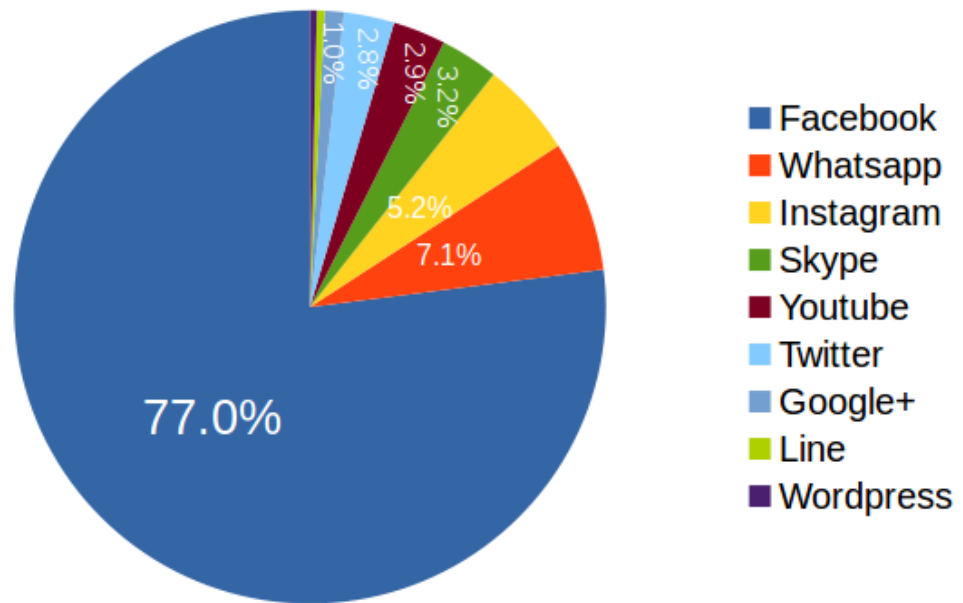


Figura 2: Comparación del tráfico web hacia plataformas de redes sociales.

Tabla 2: Cuadro de la predominancia del tráfico hacia plataformas sociales en la Universidad Técnica de Ambato.

Sitios	#Peticiones	%
Facebook	197311	77.02 %
Whatsapp	18281	7.14 %
Instagram	13271	5.18 %
Skype	8175	3.19 %
Youtube	7382	2.88 %
Twitter	7065	2.76 %
Google+	2676	1.04 %
Line	1072	0.42 %
Wordpress	948	0.37 %
Total	256181	100.00 %

Tabla 3: Cuadro de comparación entre el tráfico dirigido a redes sociales respecto a los sitios académicos en la Universidad Técnica de Ambato.

Sitios	#Peticiones	%
Academia	7777	1.86 %
Otros	153445	36.83 %
Redes Sociales	256181	61.49 %
Total	475465	100.00 %

Tabla 4: Cuadro de comparación entre el tráfico dirigido a sitios de carácter académicos respecto a Facebook en la Universidad Técnica de Ambato

Sitios	#Peticiones	%
Academia	7777	1.86 %
Otros	18281	44.65 %
Facebook	197311	47.36 %
Total	475465	100.00 %

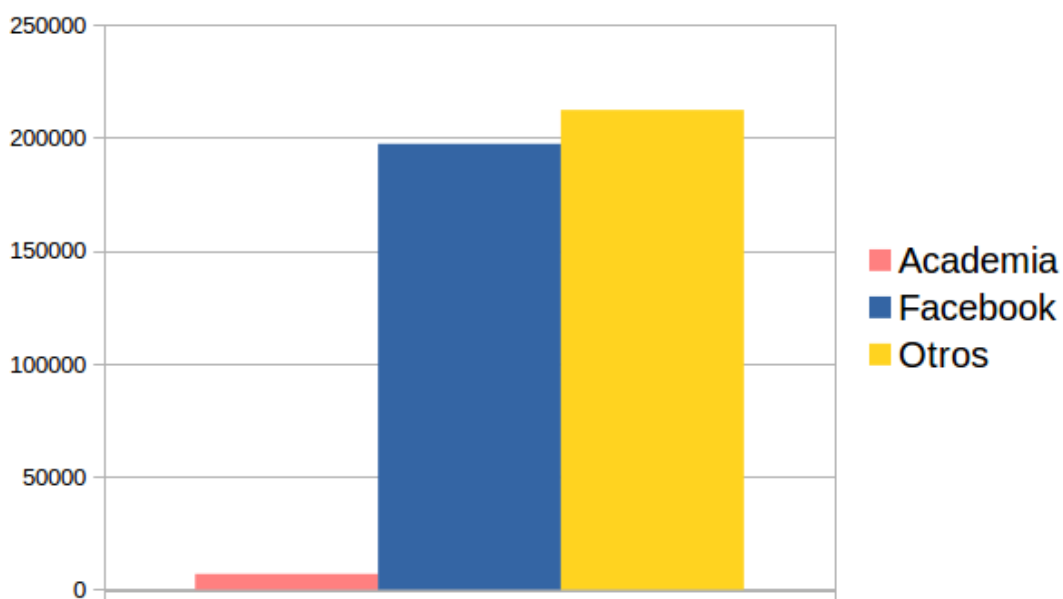


Figura 3: Comparación del tráfico web respecto a Facebook

Anexo B

Encuesta “Facebook vs Academia”

B.1. Preguntas

Cantidad de tiempo dedicada a los siguientes servicios Facebook: *

	no significativa.	poco significativa.	significativa.	muy significativa.	significativa en extremo.
Envío de mensajes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fotos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chat.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muro/noticias.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Videos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eventos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Música.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Notas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enlaces de contenido.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Grupos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Juegos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

El tiempo dedicado a facebook para:

	insignificativo.	poco significativo.	significativo.	muy significativo.	significativo en extremo.
Coordinar encuentro con un grupo de amigos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Conocer qué ocurre en el grupo de mis amigos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Comentarios de fotos, videos y otros.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compartir información, ficheros, fotos, documentos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reencontrarse con conocidos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Conocer las noticias del día.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dar a conocer lo que estoy haciendo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hacer amigos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Utilizar las aplicaciones y los juegos en la red social.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compartir mis sentimientos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Buscar un socio o buscar pareja.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Las actividades a las que les ha dejado de dedicar tiempo desde que hace uso de Facebook: *

	no afectada	afectada poco	afectada	muy afectada	afectada en extremo
Ver televisión.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hacer nada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estudiar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Leer.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dormir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Escuchar la radio.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hacer deporte.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Otras actividades.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ir al cine.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Caminar/dedicar tiempo a amigos y a la familia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Trabajar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Las actividades académicas para las que usa Facebook: *

	no lo uso para esto	poco	sí	mucho	en extremo
Resolver consultas de contenidos y exámenes con otros estudiantes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enterarse de qué han dado en clases en momentos en que no se ha asistido a clases.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hacer trabajo de curso.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mantenerse actualizado sobre lo que sucede en determinada temática.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intercambiar información útil y recursos de una temática tratada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Resolver dudas acerca de la vida en la universidad.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontrar información sobre actividades organizadas por la universidad.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organizar actividades extracurriculares.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consultar recomendaciones de libros o recursos del profesor.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contactar con expertos en los tópicos de un tema particular.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tutoriales y consultas a docentes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

B.2. Resultados

B.2.1. Cantidad de tiempo dedicado a los servicios de Facebook

Tabla 1: Resultados de pregunta acerca de la cantidad de tiempo dedicado a los servicios de Facebook

Servicio	No significativo	Poco significativo	Significativo	Muy significativo	Significativo en extremo
Envío de mensajes	8	69	121	103	36
Fotos	40	143	112	29	13
Chat	9	57	106	108	57
Muro social	33	115	108	65	16
Videos	71	130	85	41	10
Eventos	135	113	61	22	6
Música	93	123	76	34	11
Notas	133	114	65	16	9
Enlaces de contenido	66	106	97	50	18
Grupos	29	70	127	77	34
Juegos	153	101	55	19	9

B.2.2. Actividades afectadas por el uso de Facebook

Tabla 2: Resultados de pregunta respecto a las actividades afectadas por el uso de Facebook

Actividad	No afectada	Afectada poco	Afectada	Muy afectada	Afectada en extremo
Televisión	81	129	69	32	26
Nada	98	112	72	35	20
Estudiar	81	143	88	17	8
Leer	76	112	82	48	19
Dormir	120	122	65	25	5
Radio	116	105	64	31	21
Deporte	141	94	53	34	15
Otros	115	127	70	18	7
Cine	155	92	65	16	9
Amigos Caminar	140	113	55	21	8
Trabajar	168	103	40	18	8

B.2.3. Actividades que se realizan en Facebook

Tabla 3: Resultados de pregunta referente a actividades que se realizan en Facebook

Servicio	No significativo	Poco significativo	Significativo	Muy significativo	Significativo en extremo
Coordinar encuentro	14	75	133	74	41
Qué ocurre con amigos	32	99	130	56	20
Comentar fotos videos	38	124	114	49	12
Compartir	22	87	114	75	39
Reencontrar a conocidos	56	100	121	46	14
Noticias	28	90	118	76	23
Que estoy haciendo	147	123	48	13	2
Hacer amigos	87	134	74	30	9
Usar app games	143	116	47	25	5
Share feelings	184	116	26	9	2
Search partner	226	76	22	8	4

B.2.4. Actividades académicas para las que se utiliza Facebook

Tabla 4: Resultados de pregunta respecto a las actividades académicas para las que se utiliza Facebook

Actividad	No lo uso para esto	Poco	Si	Mucho	En extremo
Consultas	44	90	128	50	25
Qué se ha dado en clases	14	46	152	87	38
Hacer trabajo curso	11	43	125	105	53
Actualizarse	25	78	138	67	29
Intercambiar recursos	8	68	137	85	39
Resolver dudas	40	95	115	65	22
Encontrar información sobre actividades	34	97	127	57	22
Organizar extracurriculares	38	115	124	44	16
Consulta recursos	66	101	111	44	15
Contacto con expertos	97	110	87	33	10
Tutoriales y consulta a docentes	93	99	89	42	14

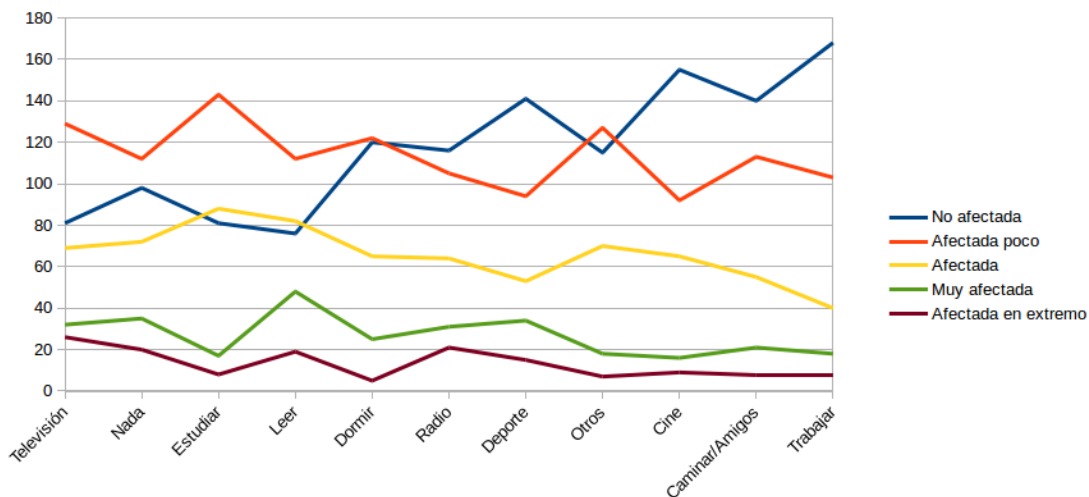


Figura 1: Actividades afectadas por el uso de Facebook

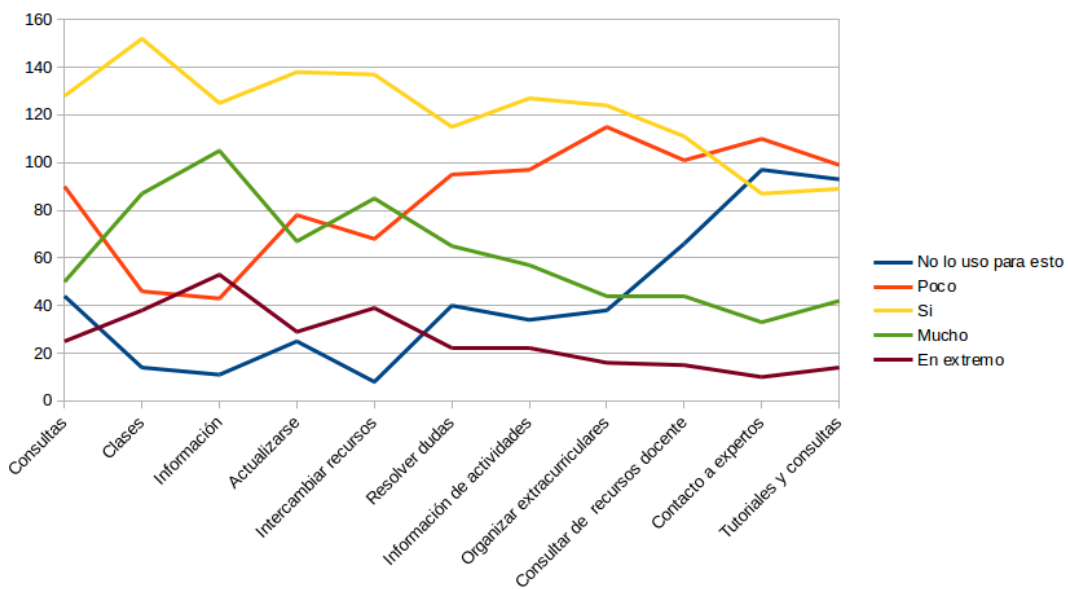


Figura 2: Actividades académicas para las que se utiliza Facebook

B.3. Análisis de información

La presente encuesta refuerza el criterio que la red social Facebook forma una parte importante de los estudiantes. Se usa principalmente para la comunicación a través del servicio de chat y mensajes con el fin de conocer y compartir información con sus amigos.

Se refuerza el criterio de que Facebook es una red social en la que se comparten contenidos. Y es el contenido por lo que los usuarios la utilizan. No es claro la pertinencia dentro de este perfil que las opciones referentes a fotos y videos den un valor adicional importante a la utilización red social. interés en la comunicación con personas conocidas (refuerza el resultado del análisis del primero de los criterios tenidos en cuenta). interés en conocer noticias / información de interés.

El perfil de los usuario que participan en ambientes académicos universitarios y utilizan Facebook, es el que se utiliza la red social para estar en contacto con amigos y familiares, compartir y consumir información. No existe una predominancia de buscar relaciones con nuevos amigos , ni tampoco de utilizar frecuentemente las aplicaciones y juegos dentro de la red social.

Finalmente analizando las tendencias y promedios no se puede definir claramente que sea Facebook un medio que se utiliza como herramienta para la realización de tareas académicas.

Anexo C

Guía de Observación

Tabla 1: Guía de observación para la recolección de requerimientos

Guía de Observación	
Objetivo	Obtener información sobre la interacción de las redes sociales en el contexto del ambiente académico.
Tipo de Observación	Directa
Indicador	Observaciones
Utilización de redes sociales	De los registros DNS del tráfico web de la Universidad Técnica de Ambato se pudo observar que el tráfico hacia redes sociales predominaba en un 61.37 %. Tab 3
Red social de mayor predominancia	Se observa en los registros respecto a tráfico web dirigido a redes sociales el 77 % de las peticiones es hacia la red social Facebook. Fig 2
Difusión del contenido académico	Existe actualmente en determinadas materias grupos de Facebook, en los cuales los estudiantes y docente pueden compartir información, contenido referente a la materia existiendo una valoración del contenido mediante las acciones de la plataforma ,«me gusta»,«compartir» y mediante la escritura de «comentarios» en las publicaciones.
Procesos que se valoran en la difusión de contenido	Disponibilidad de los contenidos. Sencillez de las interfaces. Existencia de un diseño familiar en las interfaces y procesos.

Anexo D

Levantamiento y especificación de requerimientos según estándar de la IEEE 830

D.1. Introducción

En este documento se detalla los requerimientos y especificaciones, en los cuales el proyecto de investigación cumplirá con todo lo que la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato amerite respecto a la gestión del contenido académico publicado en la plataforma de la red social Facebook.

D.1.1. Propósito

El documento de levantamiento y especificación de requerimientos cumple con el fin de establecer el alcance que tendrá el sistema y los requerimientos que necesarios para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial Este documento está destinado para los coordinadores del proyecto «SOMAN» y para el investigador como soporte de los procesos.

D.1.2. Ámbitos del sistema

- El sistema de información se denominara «**Sigma**».
- El sistema utilizará la plataforma de la red social Facebook , gestionará y controlará con lo que conlleva respecto a usuarios, contenido académico, acciones sociales. Por lo que no sule procesos para accionar como un sistema de administración general dado que el objetivo es la visualización de información y adquisición de datos respecto a esta.
- El sistema ayudará en una mejor visualización y acceso de los diversos contenidos académicos dentro del área de aplicaciones de la red social Facebook. El sistema permitirá el registro de las acciones sociales en los contenidos académicos.

D.1.3. Definiciones, Acrónimos y Abreviaturas

- **BD** Base de Datos

- **ORM** Mapeo objeto-relacional (Object-Relational mapping)
- **API** Interfaz de programación de aplicaciones (Application Programming Interface)
- **REST** Transferencia de Estado Representacional (Representational State Transfer)
- **HTTPS** Protocolo seguro de transferencia de hipertexto (Hypertext Transfer Protocol Secure)
- **FISEI** Facultad de Ingeniería en Sistemas, Electrónica e Industrial
- **SOMAN** Sistema de Gestión Académica en las Redes Sociales
- **Open Source** Código abierto
- **Facebook** Plataforma web de redes sociales
- **Canvas** Área de interacción de las aplicaciones dentro de la plataforma de Facebook
- **Node.js** Entorno en tiempo de ejecución multiplataforma para la capa del servidor
- **Usuario** Persona que usará el sistema para gestionar procesos
- **ERS** Especificación de Requisitos Software
- **RF** Requerimiento Funcional
- **RNF** Requerimiento No Funcional

D.1.4. Referencias

Título del Documento	Referencia
Standard IEEE 830 - 2008	IEEE

D.1.5. Visión General del Documento

El presente documento consta de los ámbitos que tendrá el sistema de información «Sigma», una descripción general del sistema con sus respectivas perspectivas, funciones , características de los procesos y restricciones.

Se detalla los requisitos específicos como interfaz, requisitos funcionales, de diseño, también se presenta los atributos del sistema.

D.2. Descripción General

D.2.1. Perspectiva del Producto

El sistema de información Sigma será un producto diseñado con el enfoque de trabajar en el ambiente de aplicaciones de la plataforma Facebook, lo que permitirá que de una forma rápida y eficaz se realice su utilización.

Al vincularse de forma íntegra hacia a las herramientas y servicios que provee la plataforma de Facebook , permitirá interactuar directamente con los usuarios ,poseer un grado mayor de visualización así como de acceso personalizado a estadísticas que genera la interacción del sistema con la plataforma.

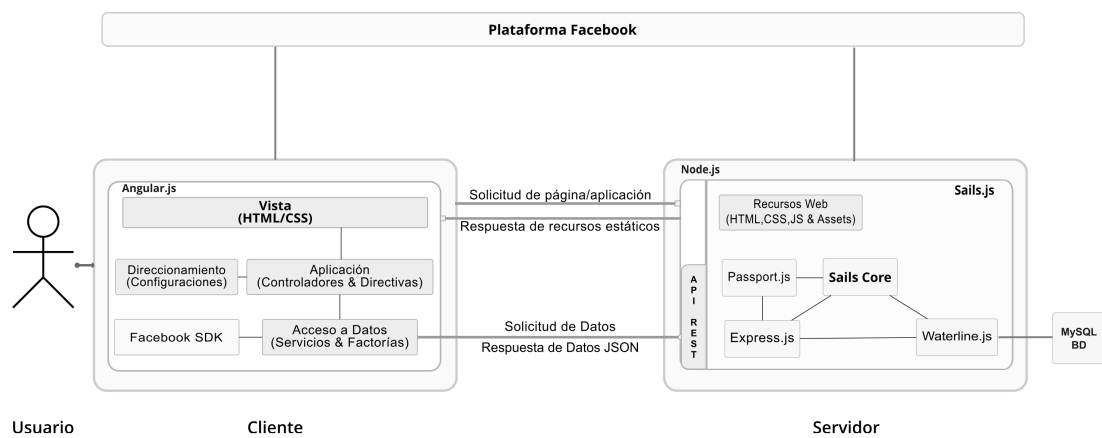


Figura 1: Perspectiva del sistema de información Sigma

D.2.2. Funciones del Producto

El sistema utilizará los recursos y herramientas de Facebook. Gestionará los procesos que conlleva respecto a la administración de usuarios desde Facebook, a la visualización de los contenidos académicos ,sus relaciones ,categorías desde registros de una base de datos como también el registro de las acciones sociales generadas por los usuarios.

D.2.3. Características de los Usuarios

El sistema va dirigido a los estudiantes y docentes de la FISEI. Por tanto, se encamina a un usuario con un conocimiento de nivel medio y de un grado de experiencia alto en la relación con ambientes y plataformas de redes sociales.

D.2.4. Restricciones

- Ejecución en el ambiente de aplicaciones que proporcionada la plataforma de Facebook.
- Utilización del protocolo de seguridad HTTPS.
- Enfoque en la usabilidad del usuario respecto al sistema.
- Relacionar acciones y procesos del sistema con los existentes en la plataforma de Facebook.

D.2.5. Suposiciones y Dependencias

Los requerimientos se asumen para un sistema GNU/LINUX, con la suficiente potencia y capacidad para dar respuesta de las peticiones de recursos y resultados en un tiempo razonable. No existen requerimientos de tiempo de respuesta ni limitaciones de memoria. Se trabaja con una base de datos que la comparten diversos sistemas, si se llegará a cambiar la estructura, el sistema tendrá que estar sujeto a revisión y pruebas de estabilidad.

D.2.6. Requerimientos futuros

Se omite para versiones futuras mejoras en la funcionalidad de los componentes en base a pruebas de usabilidad realizadas a grupos de usuarios. De la misma manera, se sugiere la utilización de un sistema experto u otras técnicas apropiadas de inteligencia artificial para la generación de enlaces y categorías del contenido académico como para la toma de decisiones.

D.3. Requisitos Específicos

D.3.1. Interfaces Externas

- Interfaces de usuario
 - Definición de una paleta general de colores con predominancia a tonos en referencias al color azul.
 - Definición de familias tipográficas de acceso rápido y de una adecuada visualización.
 - Visualización del estado de la ejecución de una acción.
 - Estilo del sistema uniforme.
 - Navegación fluida.

- Interfaces de hardware
 - El sistema no trabajará con dispositivos adicionales al computador, por lo que no existirá interfaces de conexión externas.
 - Será necesario utilizar elementos como el teclado, el ratón, el monitor elementos que conforman un equipo de computación.
- Interfaces de software
 - Implementación de servicios web API REST.
 - Se utilizará la API de la plataforma de Facebook.
 - Se predominará la utilización de software «open source»

D.3.2. Funciones

- El sistema de información «Sigma» se ejecutará dentro del ambiente de aplicaciones de la plataforma Facebook.
 - Registrará los permisos para la utilización de información de los usuarios.
 - Utilizará información proporcionada por la plataforma.
 - Utilizará los procesos de la plataforma en la valorización de contenidos.
 - Utilizará los procesos de la plataforma en la difusión de contenidos.
- El sistema gestionará respecto a los usuarios lo siguiente:
 - Registrar a los usuarios.
 - Procesar el inicio de sesión.
- El sistema gestionará el contenido académico en lo siguiente:
 - Visualizar la información relevante de los diversos contenidos a partir de la base de datos.
 - Visualizar las relaciones del contenido respecto a categorías de relevancia como también hacia otros contenidos existentes.
 - Visualizar los sitios web de origen de los diferentes contenidos a partir de una copia de estos en el sistema.
 - Visualizar los contenidos de mayor relevancia de acuerdo a las acciones realizadas de los usuarios.
- El sistema gestionará las acciones sociales sobre los contenidos de la siguiente forma:

- Registrar la acción de dirigirse al origen del contenido como «visita».
- Registrar la acción de la preferencia de valor de un contenido como «me gusta».
- Registrar la acción del «compartir».

D.3.3. Requerimientos de Rendimiento

Respecto al rendimiento, el sistema suplirá un tiempo de respuesta menor a 5 segundos según la especificación de los estándares, para garantizar que el proceso sea óptimo y de cumplimiento con los objetivos.

D.3.4. Restricciones de Diseño

- Utilización del protocolo de seguridad HTTPS.
- El acceso a la base de datos será mediante la utilización de la técnica de programación ORM.
- Ejecución en un ambiente gráfico de tamaño dinámico.

D.3.5. Atributos del Sistema

- Usabilidad
 - El sistema dispondrá de una curva de aprendizaje baja al relacionarse con el comportamiento y funcionamiento de otros sistemas y plataformas.
 - Se tendrá la visualización y sintetizabilidad del estado de la ejecución de acciones y procesos.
 - El sistema dispondrá en la interfaz de elementos que permitan la disminución de la carga cognitiva.
- Mantenibilidad
 - El sistema tendrá la posibilidad de dejarse en funcionamiento un largo periodo de tiempo.
- Portabilidad
 - La arquitectura del sistema podrá implementarse en cualquier equipo con una instalación del entorno NodeJS con acceso a los repositorios de bibliotecas de código.
- Seguridad

- El sistema utilizará el proceso nativo de autenticación provisto por Facebook.
- El cifrado de la información se dará mediante la utilización del protocolo HTTPS.
- Se utilizará el sistema de autenticación por testigos.

D.3.6. Otros Requisitos

D.4. Apéndices

Anexo E

Diccionario de Datos

Leyenda

Leyenda	
Término	Definición
PK	Primary key
NN	Not null
UN	Unique
AI	Autoincrement

Tabla “category”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
name	VARCHAR(255)		✓				
description	VARCHAR(255)						
createdAt	DATETIME					current_timestamp	
updatedAt	DATETIME					current_timestamp	

Tabla 1: Diccionario de datos. Tabla: **category**

Tabla “user”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
provider	VARCHAR(15)						
uid	VARCHAR(50)		✓				
name	VARCHAR(50)						
gender	VARCHAR(50)						
email	VARCHAR(50)						
firstname	VARCHAR(50)						
lastname	VARCHAR(50)						
profileUrl	VARCHAR(150)						URL Link to the profile
password	VARCHAR(50)						
createdAt	DATETIME					current_timestamp	
updatedAt	DATETIME					current_timestamp	

Tabla 2: Diccionario de datos. Tabla **user**

Tabla “content”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
title	VARCHAR(255)		✓				
description	LONGTEXT		✓				
url	VARCHAR(255)		✓				
image	VARCHAR(255)					NULL	Path of file
reading	LONGTEXT					NULL	Information about time of reading
state	VARCHAR(15)		✓			create	State: create, edit, disable
creator	INT(10)			✓			User who created the registry
createdAt	DATETIME					current_timestamp	
updatedAt	DATETIME					current_timestamp	

Tabla 3: Diccionario de datos. Tabla: **content**

Tabla “content_category”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
content	INT(10)		✓	✓			
category	INT(10)		✓	✓			

Tabla 4: Diccionario de datos. Tabla **content_category**

Tabla “content_related”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
main	INT(10)		✓	✓			
related	INT(10)		✓	✓			
createdAt	DATETIME					current_timestamp	

Tabla 5: Diccionario de datos. Tabla **content_related**

Tabla “visit”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
sid	VARCHAR(50)		✓				
content	INT(10)		✓	✓			
user	INT(10)		✓	✓			
time	FLOAT		✓			1	Time in Seconds.
createdAt	DATETIME					current_timestamp	

Tabla 6: Diccionario de datos. Tabla **visit**

Tabla “like”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
sid	VARCHAR(50)		✓				
content	INT(10)		✓	✓			
user	INT(10)		✓	✓			
createdAt	DATETIME					current_timestamp	

Tabla 7: Diccionario de datos. Tabla **like**

Tabla “share”

Column name	DataType	PK	NN	UN	AI	Default	Comment
id	INT(10)	✓	✓	✓	✓		
sid	VARCHAR(50)		✓				
message	TEXT						
content	INT(10)		✓	✓			
user	INT(10)		✓	✓			
createdAt	DATETIME					current_timestamp	

Tabla 8: Diccionario de datos. Tabla **share**

Anexo F

Encuesta “Usabilidad del sistema de información Sigma”

F.1. Preguntas

Analice cada una de las características de la presente encuesta y asígnele con un ✓ el peso considerando los siguiente valores:

(E) Excelente	10-9
(AD) Admisible	8-7
(A) Aceptable	5-6
(R) Regular	4-3
(NM) Necesita Mejorar	0-2

	Características	E	AD	A	R	NM
1	Visibilidad del estado del sistema					
2	Igualdad entre el sistema y el mundo real					
3	Control y libertad del usuario					
4	Consistencia y estandarización					
5	Ayuda al usuario					
6	Prevención de errores					
7	Visibilidad en lugar de rellamadas					
8	Flexibilidad y eficiencia de uso					
9	Estética y simplicidad en el diseño					
10	Ayuda y documentación					

F.2. Análisis de resultados

	Características	E	AD	A	R	NM
1	Visibilidad del estado del sistema	4	0	0	0	1
2	Igualdad entre el sistema y el mundo real	1	4	1	0	0
3	Control y libertad del usuario	0	4	2	0	0
4	Consistencia y estandarización	0	5	1	0	0
5	Ayuda al usuario	0	4	0	0	2
6	Prevención de errores	0	0	2	3	1
7	Visibilidad en lugar de rellamadas	1	2	2	0	1
8	Flexibilidad y eficiencia de uso	3	3	0	0	0
9	Estética y simplicidad en el diseño	3	2	1	0	0
10	Ayuda y documentación	0	0	1	1	4

Tabla 1: Resultados de encuesta aplicada respecto a la usabilidad del sistema Sigma

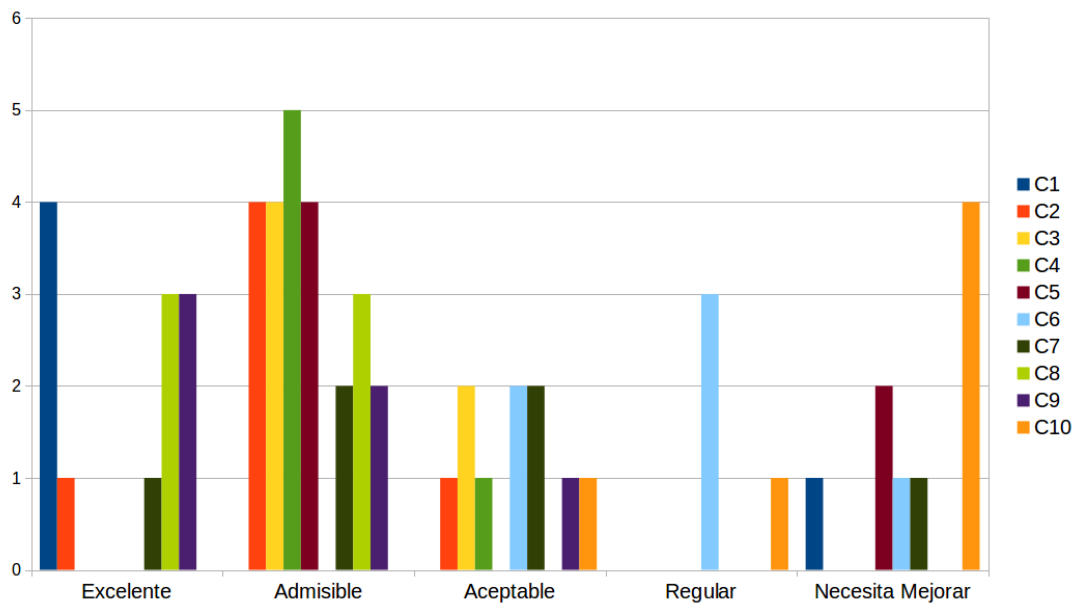


Figura 1: Gráfico de barras de los resultados de la encuesta de usabilidad. Valorización vs Características

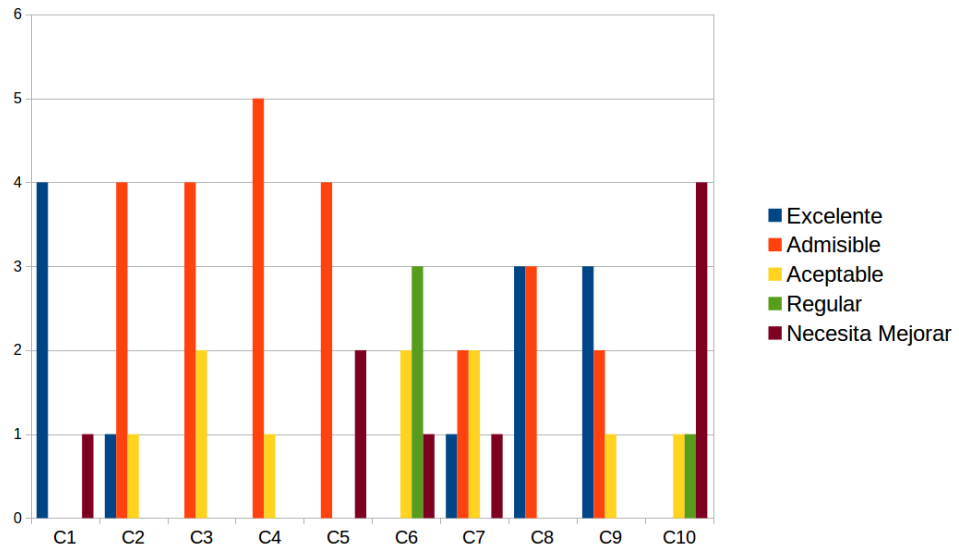


Figura 2: Gráfico de barras de los resultados de la encuesta de usabilidad. Tendencia de los valores

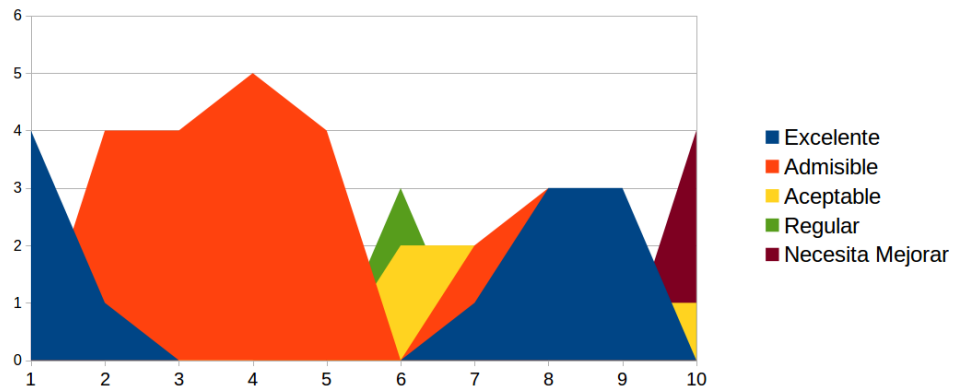


Figura 3: Gráfico de áreas de los resultados de la encuesta de usabilidad. Tendencia de los valores

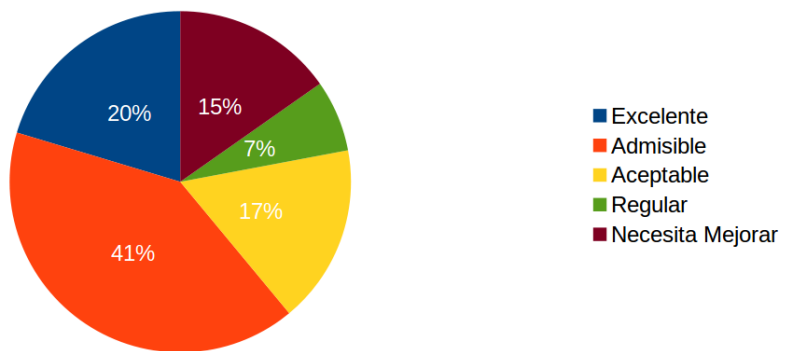


Figura 4: Gráfico en círculo de los resultados de la encuesta de usabilidad. Porcentaje respecto al total de los valores

Anexo G

Prueba de ejecución del sistema de información “Sigma”

G.1. Parámetros

Determine el tiempo de ejecución(seg) que transcurre al realizar las siguientes acciones, con las cuales se evaluará la usabilidad del sistema:

1. Instalación de la aplicación
2. Filtrar por la categoría *Base de Datos*.
3. Filtrar por la categoría *Administración*.
4. Buscar contenidos la palabra *Windows*.
5. Buscar contenidos la palabra *administracion*.
6. Hacer la acción “me gusta” a alguna entrada.
7. Hacer la acción “compartir” a alguna entrada.
8. Filtrar por una de las categorías de una entrada en particular.
9. Visualizar un contenido en detalle al pulsar “Leer más”.
10. Abrir la primera entrada por https.
11. Abrir externamente la entrada en otra ventana, a partir de la ventana modal.
12. Hacer la acción “me gusta” desde la ventana modal.
13. Hacer la acción “compartir” desde la ventana modal.

G.2. Resultados

Ítem	Tiempo de Ejecución(seg)								Sumatoria	Tiempo Promedio
1	15	22	17	7.92	13.32	19	21	24.59	140.83	17.60
2	2	20	10.64	10	25.55	10	7	21.6	108.79	13.60
3	2	4	2.7	2.52	26.02	9	8	25.36	82.6	10.33
4	4	10	7.3	24.77	25.32	7	5	15.02	102.41	12.80
5	9	15	8.64	18.96	32.96	5	5	50.64	150.2	18.78
6	1	1.5	1.2	4.58	20.01	3	15	45.96	98.25	12.28
7	9	8	9.3	5.06	18.49	19	15	19.15	110	13.75
8	2	5	20.2	12.73	18.99	10	9	4.2	90.12	11.27
9	25	4	6.4	4.7	71.22	30	18	100	268.32	33.54
10	6	12	0	3.5	5	4	4	4.4	48.9	6.11
11	2	6	9	2.55	4.5	25	5	2.3	67.35	8.42
12	2	2	0	1.55	4	15	5	3.12	44.67	5.58
13	4	4	0	2.62	6	7	10	6.15	52.77	6.60

Figura 1: Resultados de la prueba de ejecución del sistema Sigma

Anexo H

Manual de usuario del sistema “Sigma”

H.1. Introducción

El sistema de información Sigma, se creó con el fin de ayudar en la difusión de contenidos en el ámbito académico. Según varios estudios el desarrollo de comunidades de aprendizaje en ambiente de redes sociales ayuda al desarrollo cognitivo, mejoras en el aprendizaje entre otros beneficios más. Se tiene hoy una baja participación de la academia en entornos de redes sociales. Tenemos un ejemplo de rápido acceso como son los grupos sociales en Facebook, pero estos tienen sus dificultades como el tiempo de vida de una publicación en específico, componentes para realizar búsquedas de un contenido y temática en específico. Sigma trata de resolver tres problemáticas, principalmente el acceso de la academia a ambientes de redes sociales, el de disponer de valorizaciones y control del contenido académico y el mejorar los procesos de creación y difusión de contenidos.

Se hace énfasis en la usabilidad e interacción con el usuario. Al igual que al diseño y todo lo que conlleva en el proceso como son fuentes tipográficas, colores, distribución, relación y armonía a todo esto se le ha asignado un papel de importancia, no como un añadido que se dispone al final de la implementación sino siendo como una parte de la base de acción y crecimiento de todo lo que conlleva en el sistema de información Sigma.

H.2. Funcionalidades

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. A continuación se detallan las interfaces de las funciones del sistema.

H.2.1. Sistema Sigma en Facebook

Para la utilización del sistema Sigma en Facebook, es necesario realizar la *instalación* del sistema en el área de aplicaciones de Facebook, es el proceso por el cual se permite al sistema Sigma acceder a la información del usuario y

realizar acciones en nombre de este. Facebook maneja el control de la información y de acciones mediante permisos, los permisos que utiliza el sistema son:

<i>public_profile</i>	Información que por defecto que se comparte.
<i>email</i>	Se accede a la dirección de correo electrónico.
<i>birthday</i>	Acceso a la fecha de nacimiento.
<i>publish_actions</i>	Publicar contenidos en nombre del usuario.

Tabla 1: Manual de Usuario. Permisos utilizados por el sistema Sigma

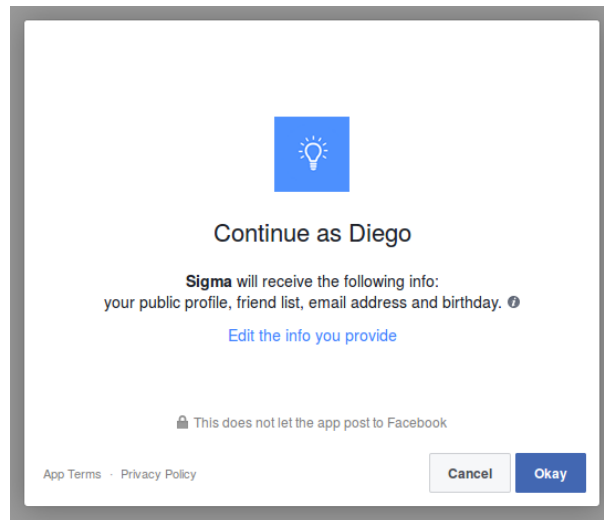


Figura 1: Manual de Usuario. Ventana para permitir el acceso a permisos de la información del usuario

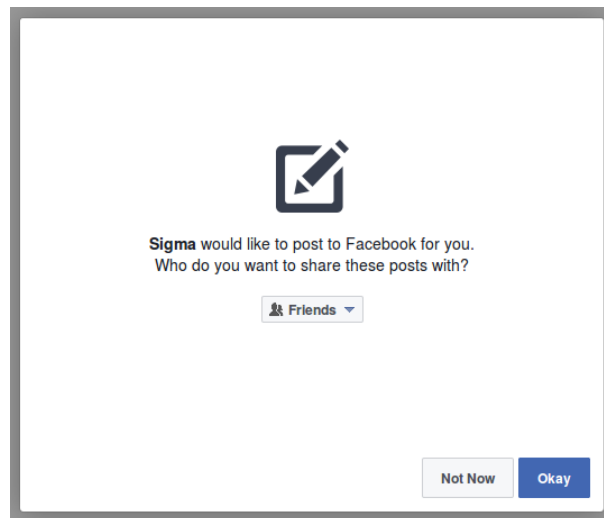


Figura 2: Manual de Usuario. Ventana para permitir publicar en nombre del usuario

H.2.2. Área de contenidos

El área de contenidos es donde se visualizan las entradas disponibles a las que tiene acceso el sistema. Los contenidos se construyen en base a registros de una base de datos.



Figura 3: Manual de Usuario. Interfaz. Área de contenidos.

- 1) Barra de búsqueda avanzada.
- 2) Opciones de filtrado rápido.
- 3) Lista de contenidos.
- 4) Paginación. Control del número de contenidos.

H.2.3. Contenido

Es en los contenidos donde se efectúan las acciones sociales. Cuando un contenido tiene un mayor número de interacciones en el sistema tendrá una mayor relevancia. En cada contenido se dispone de un estado, que es un identificador que indica si a un contenido se le permite visualizarse en el sistema.

Cada contenido hace referencia a un sitio web, desde el cual mediante la descarga se obtiene un directorio de recursos necesarios para la visualización del sitio web de origen dentro del sistema, y esta opción visualización se la realiza en una ventana modal, esto con el fin de que se pueda visualizar y consultar de una manera rápida la información deseada.

H.2.3.1. Detalle de elementos de un contenido

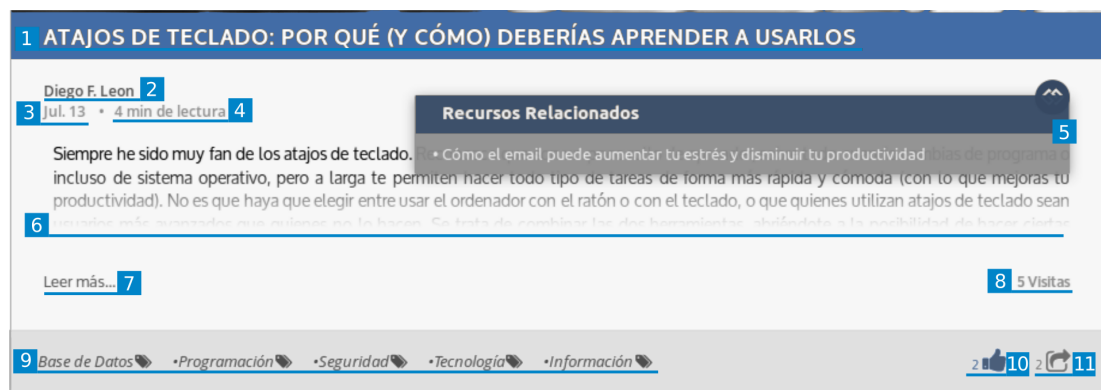


Figura 4: Elementos de los contenidos del sistema Sigma.

- 1) **Título.** 2) **Autor de contenido.** 3) **Fecha.** 4) **Tiempo de lectura.** 5) **Contenidos relacionados.** 6) **Descripción.** 7) **Visualizar sitio web de origen.** 8) **Número de Visitas.** 9) **Categorías.** 10) **Me gusta.** 11) **Compartir.**

1. **Título** Conjunto de palabras que denominan a un contenido.
2. **Autor de contenido** Usuario que creo dentro en el sistema el contenido. Al realizar una pulsación en el, se le solicita al sistema los contenidos que ha creado el usuario.
3. **Fecha** Indicación del periodo tiempo que ha transcurrido desde la publicación.
4. **Tiempo de lectura** Indicación del tiempo aproximado que se tomará la lectura.
5. **Contenidos relacionados** Lista de contenidos con los que existe una relación.

6. **Descripción** Información que ayuda en la definición del contenido.
7. **Visualizar sitio web de origen**
8. **Número de Visitas** Indicación de la cantidad de veces que se ha solicitado la visualización del contenido.
9. **Categorías** Identificadores que un contenido posee respecto a una relación de determinados tipos o clases. Al realizar una pulsación en alguna de las categorías, se le solicita la sistema los contenidos que estén identificadas con esa categoría.
10. **Me gusta** Componente para realizar la acción de “me gusta” y en el que se visualiza el número de veces que se ha realizado la acción.
11. **Compartir** Componente para realizar la acción de “compartir” y en el que se visualiza el número de veces que se ha realizado la acción.

H.2.4. Acciones sociales

Es mediante las acciones sociales que se realizan los procesos de valorización y difusión de contenidos. La interacción de las acciones sociales con Facebook es mediante la URL del sitio web a la cual el sistema hace referencia. Las acciones se registran tanto en el grafo social de Facebook como en los registros de la base de datos del sistema.



Figura 5: Manual de Usuario. Interfaz. Disposición de botones de acciones sociales

H.2.4.1. “Me gusta”

La acción “me gusta” dispone de dos estados el *realizado* y *no realizado*, el estado por defecto es *no realizado* inmediatamente cuando se ejecuta la acción se cambia el estado a *realizado* y viceversa. Al realizar esta acción se actualiza el contador aumentando o disminuyendo respecto al estado en que se encontraba el componente previamente.

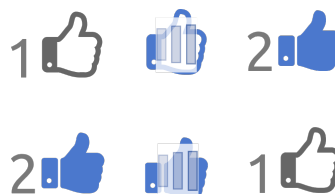


Figura 6: Manual de Usuario. Interfaz. Estados de la acción “me gusta”

H.2.4.2. “Compartir”

La acción de “compartir” no tiene una limitate respecto al número de veces que se puede realizar en un contenido. En la ventana modal que se generará al pulsar el botón , se define el mensaje y nivel de privacidad que tendrá la entrada que hace referencia al contenido. Al realizar esta acción se publicará una entrada en el muro social del usuario y se actualizará el contador.



Figura 7: Manual de Usuario. Interfaz. Estados de la acción “compartir”

H.2.5. Contenidos Relacionados

La lista de contenidos relacionados, es un componente que se visualiza cuando existan contenidos con los que se dispone una correlación mediante el cual se accede de una forma rápida a estos contenidos. Al pulsar en el título de un contenido se generará una ventana modal donde se visualizará el sitio web de origen del contenido.

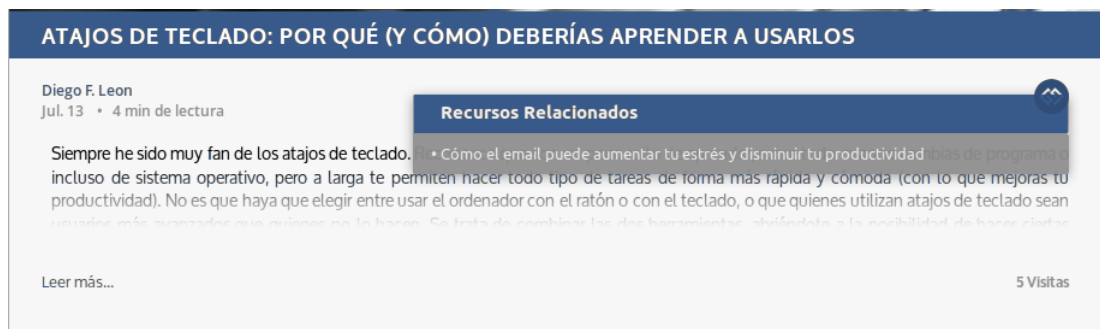


Figura 8: Manual de Usuario. Interfaz. Despliegue de menú de contenidos relacionados

H.2.6. Filtrado de contenidos

La filtración de contenidos se realiza mediante un componente que dispone de opciones para el acceso y selección de resultados de una forma rápida.

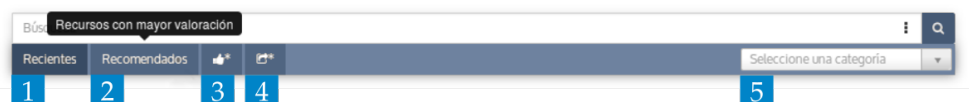


Figura 9: Interfaz. Opciones de filtrado.

- 1) Recientes.
- 2) Recomendado.
- 3) Con más acciones de “me gusta”.
- 4) Con más acciones de “compartir”.
- 5) Por categoría.

H.2.6.1. Detalle de opciones de filtrado

1. **Recientes** Contenidos que son nuevos o se han editado recientemente.
2. **Recomendado** Contenidos ordenados de acuerdo a su valoración en base al número total de las acciones que se han realizado como «visita», «me gusta» y «compartir».
3. **Con más “me gusta”** Contenidos que poseen mayor número de acciones de «me gusta».
4. **Más compartido** Contenidos que poseen mayor número de acciones de «compartir».
5. **Por categoría** Contenidos que pertenecen a una determinada categoría.

H.2.7. Búsqueda de contenidos

La búsqueda de contenidos es un componente del sistema que permite al usuario definir parámetros para seleccionar que contenidos desea visualizar. El usuario definirá los parámetros y el sistema buscará los contenidos que cumplan en su mayoría con el criterio de selección. Un contenido se visualizará si cumple con al menos la mitad de los parámetros de búsqueda.

Si llegará a no disponer de resultados que cumplan con la búsqueda el sistema visualizará un mensaje indicado que no se ha podido encontrar contenidos que cumplan con los parámetros de búsqueda.

H.2.7.1. Sintaxis de búsqueda

El sistema utiliza en el proceso de búsqueda la sintaxis Sigma. El usuario tiene la opción de generar parámetros de forma manual al escribir directamente en la barra de búsqueda o generarlos mediante la interfaz del componente. En ambos casos la sentencia de búsqueda se construye con la sintaxis Sigma.

Tabla 2: Manual de Usuario. Definición de la sintaxis Sigma de búsqueda

Búsqueda con una palabra: [<i>parámetro: palabra</i>]
Búsqueda con dos o más palabras: [<i>parámetro: (palabra1 palabra2 ... palabraN)</i>]
<i>Parámetros</i>
title: Título
creator: Creador
description: Descripción
general: Contiene la palabra o las palabras definidas
not: No tiene relación con la palabra o palabras.

Búsqueda

Limpiar

Título:

Descripción:

Autor:

Contiene las siguientes palabras:

No contiene las siguientes palabras:

Rango de Fecha:

Antes Después

🔍

Figura 10: Manual de Usuario. Interfaz. Búsqueda avanzada

DeltaSigma

CONTENIDO

NO EXISTEN ARTÍCULOS PARA VISUALIZAR.

La búsqueda de *DeltaSigma* no obtuvo ningún resultado.

Sugerencias:

- Comprueba que todas las palabras están escritas correctamente.
- Intenta usar otras palabras.
- Intenta usar palabras más generales.

Figura 11: Manual de Usuario. Mensaje de búsqueda sin resultados

H.2.8. Visualización de contenido

La visualización de contenidos es el proceso por el cual se hace referencia a los recursos necesarios para mostrar el sitio web de origen dentro del sistema mediante una ventana modal.

Si un contenido es nuevo en la primera petición de visualización se realizará un proceso de descarga del sitio web de origen, las siguientes peticiones de visualización después de esta se realizarán de una manera más rápida dado que el sistema ya tiene a disposición los recursos necesarios.

En la ventana modal se dispone la opción para el usuario de abrir el sitio web de forma externa, es decir se abrirá una nueva pestaña en el navegador solicitando el sitio web. Adicionalmente en la ventana modal se cuenta con la posibilidad de realizar acciones sociales.



Figura 12: Manual de Usuario. Interfaz. Diálogo de visualización de contenido

Tabla 3: Manual de Usuario. Preguntas Frecuentes

Preguntas	Respuestas
¿Es necesario aceptar los permisos para usar el sistema?	Si, es necesario para que el sistema pueda acceder a los datos en tu información y crear una cuenta. De igual forma si no aceptas los permisos Facebook te redirigirá hacia su tienda de aplicaciones.
¿El nombre que aparece en los contenidos, es quien creó ese contenido ?	El nombre de la persona que consta en un contenido es quien creó ese contenido, en el contexto que lo creó dentro del sistema, pero no hay que confundir con la persona que realizó ese contenido, esa persona se le denomina autor. En Sigma no se toma la propiedad intelectual de los contenidos , sería un medio más por el cual se puede difundir contenidos de una mejor manera.
¿No me aparecen resultados después de realizar una consulta ?	Debe existir por lo menos una relación de la estructura e información de los contenidos con las palabras y parámetros de la búsqueda que se ha solicitado. Una recomendación es revisar los parámetros de búsqueda, si se escribe incorrectamente eso afectará a los resultados de la búsqueda.
¿Por qué cuando registro un nuevo contenido no aparece enseguida en el área de contenidos?	Se hace un control en el registro de nuevos contenidos. Aunque todo contenido que es registrado es guardado, pero no todos los contenidos son publicados. Se pasa un control de contexto, coherencia y validez antes de que un contenido pueda pasar a ser visualizado.
¿Trato de hacer un «me gusta» en un contenido , pero no se cambia.?	Probablemente se caduco el identificador de seguridad de la sesión. Con un refresco a la página se arreglaría la situación.
¿Por qué no se ve igual los sitios web en la ventana modal como en el sitio original.?	Esto se debe que el sistema Sigma se descarga el sitio web de origen y trata de visualizar lo más cerca al original. Pero en algunos sitios aún existen problemas de seguridad por lo que se bloquean ciertos contenidos y recurso. Si al no ver adecuadamente en la ventana modal el sitio web de un contenido, lo más recomendable es abrir el sitio externamente.

Anexo I

Manual técnico del sistema “Sigma”

En el presente manual se detalla los procesos necesarios para implementar el sistema de información Sigma. Para la configuración del sistema se deben seguir los siguientes pasos. Los cuales se describen de forma global y general. El sistema de información Sigma es multiplataforma dado por la interoperatividad de Node.js, en este manual las configuraciones se hace con enfoque a sistemas operativos GNU/Linux.

Puertos Utilizados:	<ul style="list-style-type: none">▪ Sails.js: 1337▪ Web Scraper: 3100▪ MySQL: 3306▪ Redis: 6379
Versión de Software	<ul style="list-style-type: none">▪ MySQL: 5.6.0▪ Redis: 3.2.0▪ Git: 1.8.0▪ Node.js: 4.5.0▪ PM2: 1.1.3▪ Npm: 2.15.6▪ Sails.js: 0.11.2▪ AngularJS: 1.5.8

I.1. Preparación de entorno

I.1.1. Código

- Verificar que se disponga de una versión instalada igual o superior a la asignada a Git.

- Poner los archivos y directorios en la carpeta desde la cual se ejecutará el sistema.
- Ubicarse en el directorio del sistema Sigma.
- Inicializar los submódulos Git de código.


```
git submodule update --init --recursive
```
-
- Verificar la descarga de archivos en los directorios asignados a los submódulos.

I.1.2. Instalación

- **Node.js**
 - Instalar la versión determinada del paquete de Node.js
 - Crear enlace simbólico, si fuera necesario para utilizar el comando *node*.
- **Redis**
 - Instalar o definir instancia de Redis disponible.
 - Verificar ejecución y configuración de la instancia definida.
- **MySQL**
 - Instalar
 - Instalar con las configuraciones por defecto.
 - Definir contraseñas para el super usuario.
 - Para una compatibilidad de las bases de datos en diferentes sistemas operativos se debe añadir o editar en el archivo de configuración *my.cnf - my.ini* con el siguiente código:


```
lower_case_table_names = 2
```
 - Configuración
 - Crear o definir la base de datos y el usuario que utilizará para acceder el sistema.
 - Realizar la carga del esquema base del modelo de datos a la base de datos definida.


```
mysqldump -u user -h password --no-data -p database > ./init/migration-data-mastersigma.sql
```

I.2. Configuración de entorno

I.2.1. Paquetes y bibliotecas

Adicionalmente del código del código del sistema es necesario la descarga de las bibliotecas, módulos y paquetes que hace uso el sistema pero son de origen de terceros. Sin esta obtención el sistema no podría llegar a funcionar adecuadamente, dado que algunas bibliotecas y paquetes forman parte esencial de la estructura del sistema.

- Instalar los paquetes para las utilización del lado del servidor con la herramienta npm (Node Package Manager)

```
npm install
```

- Es el archivo *package.json* el que indica que paquetes y que versiones de estos se encuentran instalados o en su caso se instalarán.
- Para guardar en *package.json* la instalación o desinstalación de un paquete se debe incluir el indicador *--save*.

```
npm install paquete --save | npm uninstall paquete --save
```

- Instalar las bibliotecas para la utilización del lado del cliente con la herramienta Bower.

```
bower install
```

- Es el archivo *bower.json* es el que indica cuales librerías y recursos se encuentra instalados o se instalarán.
- Para guardar en *bower.json* la instalación o desinstalación de un paquete se debe incluir el indicador *--save*.

```
bower install recurso/biblioteca --save | bower uninstall recurso/bibl
```

- Si es necesario realizar modificaciones en Bower en aspectos como almacenamiento, acceso y conexión para estos fines se utiliza el archivo *.bowerrc*.

I.2.2. Base de Datos

- Abrir el archivo de configuración del sistema Sigma

```
./config/env/production.js
```

- Editar en la sección [application_auth] la configuración respecto a la autenticación con Facebook como dirección, claves, identificadores, permisos.

- Editar la configuración respecto a la conexión a la base de datos definida para MySQL en la sección de [connection].

```
connection: {  
.....  
sigmaPrd:{  
.....  
    host: 'localhost',  
    user: 'user',  
    password: 'password',  
    database: 'database'  
.....}  
}
```

- Editar la configuración respecto a la conexión a la base de datos definida para Redis en la sección de [session].

```
session: {  
.....  
host: '127.0.0.1',  
port: 6379,  
ttl: 1800,  
pass: 'password',  
cookie: { maxAge: 6 * 60 * 60 * 1000 }  
..... }
```

- Editar la sección de [host] si la dirección ip es diferente a la ip local privada.
- Editar la sección de [port] si se desea iniciar en un puerto de escucha diferente.
- Editar la sección de [environment] si se cambiará la definición de la identificación del entorno.
- Editar la sección de [scraper] si se dispondrá del servicio *Web Scraper* en una diferente dirección o puerto de escucha.
- Editar la sección de [log] si se desea tener un nivel diferente de registro de eventos y actividades del sistema.

I.3. Despliegue en entorno de producción

I.3.1. Inicio del sistema

- Instalar el gestor de procesos PM2.

```
sudo npm install pm2 -g
```

- Verificar con el comando de netcat la disposición de los puertos 1337 y 3100.

```
nc -zv {ip o dominio} {puerto}
```

- Ubicarse en el directorio del sistema Sigma

```
pm2 start app.js --name Sigma -x -- --prod
```

- La ejecución de los comandos de gestión de pm2 son independientes del directorio base.

- Mostrar registros de actividad y monitoreo(logs) de una aplicación.

```
pm2 logs Sigma | pm2 logs {id | nombre de aplicación}
```

- Listado de aplicaciones que se encuentran gestionadas por PM2.

```
pm2 status
```

- Descripción de un proceso de una aplicación específica.

```
pm2 show Sigma | pm2 show {id | nombre de aplicación}
```

- Monitorizar el consumo de memoria.

```
pm2 monit
```

I.3.2. Configuración servidor proxy reverso

El sistema estará escuchando en una dirección privada y por el puerto 1337, por lo que es necesario crear una vía para que los usuarios accedan al sistema.

- Disponer de la instalación de un servidor web siendo este Apache o Nginx.
- Instalar módulos para conexiones de proxy reverso.
- Activar conexiones por ssl en el servidor web.
- Creación de un *Virtual Host*.

```

<VirtualHost *:80>
    ServerName sigma.company.com
    Redirect permanent / https://sigma.company.com/
</VirtualHost>

<VirtualHost sigma.company.com:443>

    ServerName sigma.company.com

    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    #SSL Stuff
    SSLEngine On
    SSLProxyEngine On
    SSLCertificateFile /opt/secrets/sigma/sigma.crt
    SSLCertificateKeyFile /opt/secrets/sigma/sigma.key

    Header set Access-Control-Allow-Origin "*"

    ProxyPreserveHost On
    ProxyRequests off
    ProxyPass / http://sigma.intranet.com:9000/
    ProxyPassReverse / http://sigma.intranet.com:9000/

</VirtualHost>

```

Figura 1: Configuración de proxy reverso en Apache/Httpd

```

server {

    listen 80;
    listen 443 ssl;

    ssl_certificate /opt/secrets/sigma/sigma.crt;
    ssl_certificate_key /opt/secrets/sigma/sigma.key;

    server_name sigma.company.com;

    location / {
        add_header 'Access-Control-Allow-Origin' '*';
        proxy_pass http://127.0.0.1:1337;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }
}

```

Figura 2: Configuración de proxy reverso en Nginx