

# Universidad de las Ciencias Informáticas



**Título:** Subsistema de almacenamiento y recuperación de tuits

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autores:** Yanet Hing Gómez

Carlos Cecilio Parra Peña

**Tutores:** MSc. Damián Cervantes Rodón

Ing. Miguel Angel Chavez Alfonso

La Habana, junio de 2015

“Año 57 de la Revolución”

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Autorizamos a dicho centro para que haga el uso que estime pertinente de este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Yanet Hing Gómez

---

Autor

Carlos Cecilio Parra Peña

---

Autor

MSc. Damián Cervantes Rodón

---

Tutor

Ing. Miguel Angel Chavez Alfonso

---

Tutor



***“SEAMOS REALISTAS Y HAGAMOS LO IMPOSIBLE”***

ERNESTO GUEVARA DE LA SERNA

## AGRADECIMIENTOS

*A todos los profesores que de una u otra forma contribuyeron a nuestra formación como ingenieros. A los tutores, por su paciencia y dedicación. A los miembros del tribunal y al oponente que influyeron en el perfeccionamiento de esta investigación. A la revolución por darnos esta oportunidad.*

### **Yanet:**

*Agradecer a mis padres, a los cuales les debo todo lo que soy, y de los cuales siempre será poco lo que pueda decir. Por todo lo que han hecho por mí, por enseñarme a mantener y defender mis principios e ideales “contra viento y marea”, por demostrarme que no importa lo que hayamos hecho siempre se podrá hacer más, por tantas, tantas cosas que no soy capaz de describir. Por todo su amor y cariño, por su constante preocupación, por demostrarme que son capaz de hacer lo posible y lo prácticamente imposible para lograr la felicidad de todos sus seres queridos, por acompañarme siempre en cada instante de mi vida, por ser las estrellas alrededor de la cual gira mi vida. Mami, Papí, si pudiese pedir un deseo, desearía que fueran eternos, porque no puedo siquiera imaginar cómo sería la vida sin ustedes. A mis ángeles de la guarda, mis reinas, mis abuelas mima y nena por malcriarme, por protegerme por ser también mis guías. A mis segundos padres, mis tíos Alain y Yole, por acogerme como la hija hembra que no tuvieron. Gracias por traer al mundo a esos dos que llamo mis hermanos, que desde que dieron a luz son mis ojos y por ellos también este esfuerzo. A mi hermana Yeimy que a pesar de la distancia que nos separa he sentido que siempre he tenido su apoyo incondicional. A mi prima Elizabeth por ser mi hermana también. A Yotsan por estos 4 años juntos compartiendo alegría y tristezas y a sus padres por tanto amor y cariño que han depositado en mí. A mis amigos, que saben quiénes son, porque me han apoyado en*

*alguna etapa de mi vida y me han enseñado el significado de la amistad (Lachi, Ale, Lili, Lisi, Doris). A quien hizo posible que este trabajo se desarrollara al ingeniero Victor M. Herrera, no tenemos palabras para agradecerte lo que hiciste por nosotros. Gracias de todo corazón. A mi compañero de tesis Carlos por soportarme en todo este tiempo, por su amistad, comprensión y paciencia. A toda mi linda familia, aquellas que no están conmigo, las que están más cerca de mí y las que veo poco, pero que siempre las recuerdo.*

**Carlos:**

*A mis tías Elida y Mary por tratarme como a un hijo, gracias por su apoyo en todo momento. A mi tío Cesar, que ha sido un padre para mí siempre, gracias. A mi mamá-abuela por el amor que siempre tiene para mí, gracias mamá. A toda mi familia, mi papá, mi tía Zenaida, a mis primos Efraín, Darien, Yudi, mi tío Senen. A todos, gracias por ayudarme. A Yanet, mi compañera de tesis, por su apoyo en todo, por su amistad, por permitirme ser parte de sus amigos. A mi amigo y hermanito Alejandro, por su amistad sincera e incondicional, por siempre estar en las buenas pero también las malas. A mis amigos Jorge Fraga y Gretel por compartir día a día los buenos y malos ratos. A mi amigo Víctor por su ayuda todos estos años, gracias Vic. A mis amigos Lisandra, Yotsan, Lázaro, Roberto, Jorge Enrique y Leonardo por su apoyo. A todos los que de una forma u otra me ayudaron estos cinco años.*

## DEDICATORIA

### *Yanet*

*Dedico esta tesis a esos que me dieron la vida, a mis padres Níurka Gómez Tejeiro y Carlos Hing Arguiz y a mis abuelas, míma y nena.*

### *Carlos*

*A la persona más importante en mi vida, mi hermana.*

## RESUMEN

Internet es uno de los principales medios de información que existe hoy en día, pero cuenta con un gran volumen de información, muy dispersa. Por ello, el almacenamiento y recuperación de tuits de la red social Twitter para sus posteriores análisis y las herramientas que sirven de soporte a dicha actividad, son de vital importancia, sobre todo cuando se está bajo constante ataque mediático como lo está Cuba. En la Universidad de las Ciencias Informáticas (UCI) específicamente en el Centro de Ideoinformática (CIDI) se hace necesaria la creación de una herramienta capaz de almacenar y recuperar los tuits de la red social Twitter para facilitar el trabajo con esta y el acceso inmediato y permanente de los tuits. El presente trabajo de diploma tiene como objetivo desarrollar un subsistema de almacenamiento y recuperación de tuits que facilite el trabajo con la red social Twitter haciendo uso de técnicas de almacenamiento y recuperación, que permita economizar los recursos de red y aumentar la disponibilidad de los tuits. El desarrollo de la solución propuesta está guiado por la metodología de desarrollo OpenUP e implementado con el uso de Python como lenguaje de programación. Se utiliza como sistema gestor de base de datos a MongoDB, PyCharm como entorno de desarrollo y Apache Solr como servidor de indexación para el almacenamiento y la extracción de tuits. Entre los beneficios que aportó la solución se encuentran el almacenamiento y recuperación de tuits para su posterior procesamiento y brindar servicios a terceras aplicaciones para la consulta de los tuits almacenados.

**Palabras Claves:** almacenamiento, recuperación, tuits, Twitter.

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>7</b>
1.1 INTRODUCCIÓN.....	7
1.2 REDES SOCIALES.....	7
1.3 TWITTER .....	8
1.4 ALMACENAMIENTO Y RECUPERACIÓN DE INFORMACIÓN .....	10
1.5 SOLUCIONES EXISTENTES .....	11
1.6 METODOLOGÍAS, HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO .....	13
1.7 CONCLUSIONES PARCIALES .....	26
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN.....</b>	<b>27</b>
2.1 INTRODUCCIÓN.....	27
2.2 PROPUESTA DE SOLUCIÓN.....	27
2.3 MODELO DE DOMINIO.....	28
2.4 REQUERIMIENTOS DEL SISTEMA.....	29
2.5 MODELO DE CASOS DE USO DEL SISTEMA.....	32
2.6 ARQUITECTURA DEL SISTEMA.....	39
2.7 PATRONES DE DISEÑO .....	41
2.8 MODELO DE DISEÑO.....	44
2.9 MODELO DE BASE DE DATOS.....	46
2.10 CONCLUSIONES PARCIALES .....	47
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>48</b>
1.1 INTRODUCCIÓN.....	48
1.2 MODELO DE IMPLEMENTACIÓN DEL SISTEMA.....	48
1.3 VALIDACIÓN DEL SISTEMA .....	52
1.4 PROTOTIPO DE INTERFAZ DE USUARIO .....	57
1.5 CONCLUSIONES PARCIALES .....	59
<b>CONCLUSIONES .....</b>	<b>60</b>
<b>RECOMENDACIONES.....</b>	<b>61</b>

<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>62</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>68</b>

## ÍNDICE DE FIGURAS

Figura 1: Propuesta de solución del sistema. Elaboración propia.....	28
Figura 2: Modelo de dominio .....	29
Figura 3: Diagrama de Casos de Uso del sistema.....	33
Figura 4: Modelo N-Capas del sistema .....	40
Figura 5: Ejemplo de utilización del patrón Experto en información.....	43
Figura 6: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Autenticar usuario .....	44
Figura 7: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Gestionar sistema externo.....	45
Figura 8: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Consultar tuits ....	45
Figura 9: Modelo físico de datos para MongoDB .....	46
Figura 10: Modelo físico de datos para Solr .....	47
Figura 11: Diagrama de componentes del subsistema de almacenamiento y recuperación de tuits .....	49
Figura 12: Diagrama de despliegue.....	51
Figura 13: Resultados obtenidos en las pruebas funcionales al subsistema de almacenamiento y recuperación de tuits.....	56
Figura 14: Resultados de pruebas con Jmeter con 50 usuarios concurrentes .....	57
Figura 15: Resultados de pruebas con Jmeter con 150 usuarios concurrentes .....	57
Figura 16: Prototipo de interfaz. Autenticar usuario .....	58
Figura 17: Vista de la herramienta descargando tuits.....	58
Figura 18: Vista de la herramienta mostrando los tuits solicitados por un cliente.....	59

## ÍNDICE DE TABLAS

Tabla 1: Comparación de sistemas existentes. Elaboración propia .....	12
Tabla 2: Actores del sistema. Elaboración propia .....	33
Tabla 3: Descripción del caso de uso del sistema "Gestionar sistemas externos" .....	34
Tabla 4: Descripción del caso de uso del sistema " Consultar tuits" .....	36
Tabla 5: Descripción del caso de uso del sistema "Gestionar roles" .....	37
Tabla 6: Descripción de las variables correspondiente al caso de prueba para el caso de uso Autenticar usuario .....	53
Tabla 7: Descripción del caso de prueba para el caso de uso Autenticar usuario.....	53
Tabla 8: Descripción de las variables correspondiente al caso de prueba para el caso de uso "Gestionar usuario" .....	54
Tabla 9: Caso de prueba para el caso de uso Gestionar usuario: Sección 1: "Crear usuario" .....	54

## Introducción

A partir del desarrollo que ha alcanzado la informática, el pensamiento de las personas ha ido también evolucionando, lográndose implantar grandes paradigmas que han permitido establecer nuevos retos y modelos que contribuyen al beneficio de la sociedad. Uno de esos paradigmas fue la creación de Internet, una red de redes basada en los protocolos TCP/IP, una comunidad de usuarios que usan y desarrollan estos protocolos y el conjunto de recursos accesibles desde ella [1].

Para que Internet se extendiera por todo el orbe fue necesario crear la web 1.0, que está estructurada básicamente sobre documentos y enlaces de hipertexto, fue diseñada para la lectura humana y no para ser procesada de forma automática. Esta web no estaba centrada en la comunicación bidireccional, ni en el usuario. Por estas razones surge la web 2.0, que está centrada en los elementos anteriores y que contiene variadas vías para que los usuarios interactúen entre ellos y con la propia red [2].

Dentro de la web 2.0 las Redes Sociales constituyen una parte, quizás la más visible y prometedora [3]. La investigación de José Luis Orihuela Colliva demuestra que las redes sociales digitales son definidas como servicios basados en la web que facilitan a sus usuarios relacionarse, compartir información, coordinar acciones y en general, mantenerse en contacto. Son espacios que permiten reconstruir o mantener, en el mundo virtual, los vínculos que alguna vez se establecieron en el mundo físico [4].

Entre las principales redes sociales de Internet se encuentran: Facebook, Twitter, LinkedIn, Pinterest, Google Plus+, Tumblr, Instagram, Flickr. Twitter, como red social, surgió en el año 2006 en Estados Unidos. Su objetivo principal es proporcionar a los usuarios la capacidad de transmitir y consumir fácilmente pequeñas informaciones sobre el mundo. Estas informaciones, o "tuits", tienen un límite máximo de 140 caracteres, que se muestran en la página principal del usuario. Los usuarios pueden suscribirse para recibir los tuits de otros usuarios, a esto se le llama "seguir" y a los usuarios abonados se les denominan "seguidores" [5].

Los tuits pueden ser texto, fotos, videos o direcciones electrónicas que remiten a otros sitios web. Hasta noviembre de 2009 la pregunta clave del servicio era "¿Qué está pasando?" y actualmente es: "¿Qué

estás haciendo?". En ese mismo año, dicha plataforma alcanzó su popularidad internacional con más de 75 millones de usuarios registrados [6]. A lo largo de los años la cantidad de información se va haciendo cada vez más amplia y gracias al desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), hoy se cuenta con los medios necesarios para poner a disposición de los usuarios la información que se requiere [7].

La ampliación de los servicios de conectividad en el país está en consonancia con la estrategia planteada por Cuba de facilitar cada vez más el acceso de la población a las nuevas tecnologías, en dependencia de la disponibilidad de recursos y con un enfoque que favorezca su uso social. Esto es posible, además por estar ya en funcionamiento el cable de fibra óptica entre Cuba y Venezuela, el cual garantiza una mayor calidad y estabilidad en las comunicaciones internacionales de la isla, torpedeadas por las limitaciones que impone el bloqueo económico, comercial y tecnológico de Estados Unidos [8], vigente desde 1992, permitiendo solo el acceso de Cuba a Internet a través del satélite, lo que resulta que la capacidad real de conexión sea limitada, lenta y de alto costo [9].

Las líneas de teléfonos y las computadoras personales (con módem) son elementos esenciales para el acceso y por tanto, para la conexión de un país a Internet. El bloqueo impuesto por el gobierno norteamericano a Cuba le obliga a adquirir parte del equipamiento necesario a través de terceros países, que encarecen estas compras, esto trae consigo que la inversión en la modernización de las computadoras y las telecomunicaciones, tiene sentido desde el punto de vista económico por sus altos precios y restricciones de compra. A pesar de estas limitaciones, el estado cubano busca soluciones por fortalecer el acceso a las TIC, dentro de un marco económico de limitaciones.

En Cuba se han construido numerosas universidades en función de la formación de profesionales para la sociedad, ejemplo la Universidad de las Ciencias Informáticas (UCI), la cual brinda a los estudiantes y profesores un alto rigor de estudio, con la misión de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, al mismo tiempo producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática [10].

La UCI fue creada en el año 2002 por el Comandante en Jefe Fidel Castro Ruz y cuenta con disímiles proyectos trabajando sobre diversos temas, entre estos se encuentra el Centro de Ideoinformática, que tiene como una de sus misiones proveer soluciones, productos y servicios relacionados con las tecnologías de Internet, en función de la defensa de la ideología socialista en la red de redes.

El crecimiento exponencial del uso de las redes sociales en Internet, trae consigo que muchas de las aplicaciones y servicios implementen estrategias para integrarse con estas. En estos últimos años se ha visto un aumento en el uso de la red social Twitter. Luchas como la de la liberación de los 5 Héroes Cubanos injustamente encarcelados en los Estados Unidos y la divulgación de noticias nacionales e internacionales por parte de prestigiosos medios de prensa cubanos encuentran espacio en esta red social.

Teniendo en cuenta el volumen de información que genera y las limitaciones de conexión a Internet en Cuba, la información almacenada será de gran importancia y utilidad para los usuarios. Por tanto con el dinamismo con que se publican y borran mensajes en Twitter se hace necesario utilizar aplicaciones para consultar o almacenar tuits de forma periódica. El hecho de no contar con un almacén común de tuits para su posterior procesamiento provoca un consumo innecesario de ancho de banda descargando la misma información mediante el uso de estas aplicaciones, pues cada una necesita implementar una comunicación directa con la red social Twitter y por tanto es mayor consumo del ancho de banda de Internet.

Por lo antes expuesto se plantea como **problema de investigación**:

¿Cómo garantizar el acceso a los datos de la red social Twitter en Cuba teniendo en cuenta las limitaciones existentes de conectividad?

Para solucionar el problema planteado se define como **objetivo general** de la investigación:

Desarrollar un subsistema de almacenamiento y recuperación de tuits que facilite el trabajo con la red social Twitter haciendo uso de técnicas de almacenamiento y recuperación, que permita economizar los recursos de red y aumentar la disponibilidad de los tuits.

De este se derivan los siguientes **objetivos específicos**:

- ✓ Valorar el marco teórico conceptual y el estado del arte respecto a las tecnologías y funcionalidades de los sistemas de almacenamiento y recuperación de información.
- ✓ Implementar un subsistema que permita la descarga de tuits desde Twitter a partir de consultas parametrizadas, para facilitar su almacenamiento y distribución a aplicaciones de terceros.
- ✓ Validar el correcto funcionamiento del subsistema desarrollado.

Se tiene como **objeto de estudio**: los procesos de almacenamiento y recuperación de tuits. Centrándose específicamente en el **campo de acción**: la automatización de los procesos de almacenamiento y recuperación de tuits de la red social Twitter. Se plantea como **idea a defender** que el desarrollo de un subsistema de almacenamiento y recuperación de tuits, permitirá economizar los recursos de red y aumentar la disponibilidad de los tuits, para garantizar el acceso a los datos de la red social Twitter en Cuba.

Para lograr el objetivo anteriormente planteado se establecen las siguientes **tareas de investigación**:

- ✓ Estudio de los conceptos asociados al marco teórico de la investigación.
- ✓ Definición de los mecanismos de intercambio de información para proveer servicios a aplicaciones de terceros.
- ✓ Determinación de la estructura de base de datos a utilizar para el almacenamiento de la información.
- ✓ Desarrollo de las funcionalidades que permitan la descarga y almacenamiento de tuits para su posterior consumo.
- ✓ Diseño de los mecanismos de prueba que serán aplicadas en la propuesta de solución.
- ✓ Validación del subsistema desarrollado teniendo en cuenta su funcionamiento.

El **aporte de la investigación** está dado porque el subsistema de almacenamiento y recuperación de tuits, estará enfocado en facilitar el acceso a los tuits, diseñado de manera tal que aumente la disponibilidad de estos para su posterior procesamiento. Además brinda la posibilidad de almacenar y recuperar la información (tuits) de forma dinámica.

La realización de esta investigación está guiada por los **métodos científicos**, dentro de estos se emplearon **métodos empíricos y teóricos** de la investigación científica. De los empíricos se utilizó la **entrevista** con el objetivo de identificar las principales funcionalidades del sistema, a través de una conversación planificada con el cliente principal (Ver Anexo 1).

El método teórico **analítico-sintético** permite la descomposición mental del objeto o fenómeno en sus múltiples relaciones o componentes para facilitar su estudio. Establece mentalmente la unión entre las partes previamente analizadas, a parte de descubrir sus características generales y las relaciones esenciales entre ellas [11]. El uso de este método en la investigación permitió analizar la teoría y documentos para la extracción de los elementos más importantes que se relacionan con los sistemas de almacenamiento y recuperación de tuits.

También se utilizó como método teórico el **histórico-lógico** que analiza la trayectoria completa del fenómeno, su condicionamiento a las diferentes prioridades de la historia, revelan las etapas fundamentales de su desenvolvimiento y las conexiones históricas fundamentales [11]. La utilización de este permitió conocer las tendencias actuales de los sistemas de almacenamiento y recuperación existentes. Además recopilar información para describir las tendencias tecnológicas a utilizar.

Por otra parte el uso del método teórico **modelación** facilitó elaborar los diagramas y modelos de la solución propuesta que posibilitó el desarrollo de la misma.

La presente investigación consta de introducción, tres capítulos, conclusiones y recomendaciones. El Capítulo 1 titulado “Fundamentación teórica” ofrece los conceptos básicos asociados al dominio del problema. Al mismo tiempo brinda el estado del arte respecto al tema de investigación, así como las técnicas, tecnologías, metodologías y herramientas en las que se apoya la solución del problema.

El Capítulo 2 denominado “Análisis y diseño de la propuesta de solución” describe las capacidades o funciones que la aplicación debe cumplir (requisitos funcionales) y las propiedades o cualidades que el producto debe tener (requisitos no funcionales). Se realiza una descripción de los casos de uso implementados a partir del diagrama de casos de uso del sistema, se realiza el diseño de la aplicación a partir del diagrama de clase del diseño y se describe la arquitectura del sistema, así como los patrones de diseño.

En el Capítulo 3 “Implementación y validación de la solución propuesta” se muestra el diagrama de componentes y su descripción, con el fin de comprender los elementos de *software* del subsistema y sus relaciones. Se visualizan los prototipos de interfaz principales del sistema desarrollado donde se evidencian las funcionalidades implementadas. También se explica la validación del diseño realizado mediante las pruebas de rendimiento y de funcionalidad.

## Capítulo 1: Fundamentación teórica

### 1.1 Introducción

En el presente capítulo se brinda una visión general de los aspectos relacionados con los sistemas de almacenamiento y recuperación de información. Igualmente se mencionan las características de cada tipo de herramientas o técnicas, así como la descripción de las principales definiciones asociadas al dominio del problema, que son necesarias para entender el negocio y la propuesta de solución. Se definen los conceptos más importantes a manipular en el resto del documento, se hace una valoración de aplicaciones similares existentes en el mundo y una comparación entre estas. Se emite una breve descripción de la metodología de análisis y diseño escogida y de las herramientas de desarrollo usadas para la confección del subsistema.

### 1.2 Redes sociales

Muchos han sido los criterios expuestos de lo que se conoce como red social, todos de una forma u otra contribuyen a ampliar la definición. Según Zamora: “Las redes sociales en Internet son formas de interacción social, definida como un intercambio dinámico entre personas, grupos e instituciones en contextos complejos. Un sistema abierto y en construcción permanente que involucra a conjuntos que se identifican en las mismas necesidades y problemáticas y que se organizan para potenciar sus recursos” [12].

Por otro lado Manetto dice “En el mundo físico todos están inmersos en distintas redes sociales, desde la familia, que es la primera red social, los vecinos, los amigos, los compañeros de colegio, de trabajo, de deporte... Solamente que nuestras relaciones en esas redes sociales están establecidas a partir de los vínculos de primer nivel” [13].

Otro autor como Sueiras refiere que “Son sitios web que ofrecen servicios y funcionalidades de comunicación para mantener en contacto a los usuarios de la red. Se basan en un *software* especial que integra numerosas funciones individuales: blogs, wikis, foros, chat, mensajería, correo electrónico, chat de voz, transferencia de ficheros, grupos de discusión, en una misma interfaz y que facilita la conectividad entre los diversos usuarios de la red” [14].

Dadas las definiciones antes mencionadas, los autores de la investigación concluyen que una red social digital es un sitio web donde se propicia la interacción entre las personas e instituciones, según sus propias necesidades, problemáticas o intereses. Permite mantener, potenciar o crear nuevas conexiones entre los usuarios de la red.

El uso de las redes sociales aportan varios beneficios como: construir la identidad personal y/o virtual debido a que facilita a los usuarios establecer lazos y relaciones con personas que comparten los mismos intereses, preocupaciones y necesidades, y se evitan todo tipo de barreras tanto culturales como físicas. Posibilita el reencuentro con conocidos. Diluyen fronteras geográficas y sirven para conectar personas sin importar la distancia [15].

De igual importancia proporciona información actualizada acerca de temas de interés y la comunicación puede ser en tiempo real. Favorecen la participación y el trabajo colaborativo entre las personas, es decir, permiten a los usuarios participar en un proyecto de forma *online* desde cualquier lugar. Por otro lado facilidad de conseguir la información que se requiere o precise en cada momento, debido a la actualización instantánea de la información [15].

En la actualidad existen diversas redes sociales con millones de usuarios, una de las más importantes es Twitter. Esta red ha crecido en los últimos años, llegando a 560 millones de usuarios activos en el mundo [16]. Hoy en día formar parte de una red social ya no es cuestión de gusto pues ha empezado a ser parte de la necesidad de estar actualizados y presentes socialmente.

### **1.3 Twitter**

Twitter es un servicio de microblogging, con sede en San Francisco, California, en Estados Unidos. La idea original surgió dentro de la compañía Odeo. El nombre original del servicio de mensajería iba a ser "twittr", inspiración de Noah Glass en alusión a la red social "Flickr". La versión definitiva se lanzó el 15 de julio de 2006. En octubre de ese año, Biz Stone, Evan Williams y Jack Dorsey formaron Obvious Corporation, adquirió los derechos sobre Odeo, luego de que los inversores originales perdieran su interés ante el fracaso inicial del proyecto [17].

La plataforma de Twitter ofrece el acceso a los datos almacenados a través de varias interfaces de programación. Cada Interfaz de Programación de Aplicaciones (por sus siglas en inglés API) representa una faceta de Twitter que facilita a los desarrolladores construir y ampliar sus aplicaciones en formas nuevas y creativas [18]. La plataforma de Twitter incluye una serie de herramientas tales como:

- ✓ *Twitter for Websites* (TfW): es un conjunto de productos que integra aplicaciones a Twitter de manera fácil, por ejemplo se puede integrar el botón de “*Twitter*”, con este los usuarios publican un tuit desde su aplicación directamente a su página. También se puede integrar el botón “*Follow*” que facilita a los usuarios seguir su cuenta en Twitter, sin necesidad de visitar directamente esta red social.
- ✓ *Search API*: posibilita que las aplicaciones consulten datos en Twitter, esto incluye búsquedas de tuits por palabras clave y por usuario.
- ✓ *REST API*: esta API proporciona a los desarrolladores el acceso a funcionalidades del núcleo de Twitter como: obtener líneas de tiempo, actualizaciones de estado, información del usuario y el grafo de seguidores de Twitter. Con el uso de esta se puede publicar y contestar los tuits.
- ✓ *Streaming API*: esta API provee actualizaciones en tiempo real de la información de la red social. Permite que variadas palabras clave sean seguidas, de la misma manera que recupera los tuits geo-etiquetados de una determinada región y los mensajes de estado de un conjunto de usuarios.

Para realizar la comunicación con REST API y Streaming API, es necesario registrar una aplicación en la red social. La plataforma de Twitter brinda bibliotecas para varios lenguajes tales como: PHP, C# y Java, que facilitan el proceso de integración de las aplicaciones a la red social, incluyendo el proceso de autenticación con el API a través del protocolo OAuth.

En cuanto al uso de la red social Twitter, el autor Del Moral [19] clasifica su uso principalmente en:

- ✓ *Mantenimiento de amistades*: significa seguir en contacto con amigos, colegas o excompañeros de trabajo, conocidos de verano.
- ✓ *Creación de nuevas amistades*: si bien las redes mantienen el contacto entre personas que se conocen, cada una de las personas que participa, relaciona de una forma u otra, a sus contactos

con segundas o terceras personas, que pueden a su vez interactuar y conocerse. Así el amigo de un amigo, puede llegar a ser contacto y posteriormente amigo de un tercero.

- ✓ Entretenimiento: aunque las redes sociales sirven para interactuar y acrecentar las relaciones, también hay un perfil de usuarios donde se usa como portal de entretenimiento, sin más pretensiones. Exploran las actualizaciones del estado de algunos usuarios, se ponen al día sobre vidas ajenas, descubren los nuevos colegas de antiguos compañeros de clase. Es el recurso de observar lo que acontece sin ser visto.
- ✓ Gestión interna de organizaciones empresariales: sin duda, este uso está limitado a empresas donde en su estructura crean redes sociales privadas para agilizar trámites, comunicaciones, conferencias, informes o se crean otras redes simplemente para poder estar en contacto con profesionales del sector, tanto a nivel laboral como personal.

Twitter es un medio de información útil, de rápido crecimiento, fácil de utilizar y de gran cobertura, que ofrece múltiples oportunidades para mantener actualizados a los usuarios en temas de salud, educación e investigación. Cuando se actualiza habitualmente una cuenta de Twitter se cae en la problemática de que, en ocasiones, se quiere buscar algo que se ha publicado en días anteriores y debido a la cantidad de actualizaciones hechas se convierte en una tarea ardua y compleja, la búsqueda de la información que genera esas actualizaciones se facilitaría si se pudiera almacenar y recuperar los tuits para su posterior procesamiento.

#### **1.4 Almacenamiento y recuperación de información**

Una de las tendencias con mayor auge en el entorno informático es la disponibilidad inmediata y permanente de la información. Siguiendo esta línea, el **almacenamiento de información** o datos es una tarea indispensable que apunta en esta dirección. Los procesos de almacenamiento digital ayudan a reducir los costos operativos. El almacenamiento de información, datos o archivos digitales se utiliza para guardar datos, procesarlos y usarlos como programas o información de cualquier tipo, el mismo posee varias ventajas que a continuación se describen [20]:

- ✓ Permite duplicar y enviar información a través de medios digitales (sin sacarlo de su ubicación).
- ✓ Conserva la información intacta a lo largo del transcurso del tiempo, es decir no se deteriora.

- ✓ Acceso inmediato a la información según distintos criterios de búsqueda. Por ejemplo, por año, por usuario, por tema.
- ✓ Acceso simultáneo de varios usuarios a la misma información.
- ✓ Establece criterios de acceso de forma flexible y fácil.

La **recuperación de información** constituye una parte fundamental en las aplicaciones de *software* que se desarrollan, ya que facilita a las entidades o aplicaciones acceder de forma rápida y objetiva a los datos almacenados. Gracias al desarrollo de las TIC, hoy se cuenta con los medios necesarios para poner a disposición de los usuarios todo tipo de recursos, tanto físicos como digitales y en los más disímiles formatos [7].

Sobre este tema Michel López Camino en su trabajo de diploma plantea que la diversidad de recursos disponibles se convierte en el principal obstáculo para obtener la información deseada en el menor tiempo posible y una de las variantes más utilizadas para resolver este problema es la creación de herramientas de recuperación. El proceso conocido como recuperación de información es la base de las herramientas de búsquedas y trata los aspectos relacionados con el almacenamiento, organización y representación de la información, algoritmos de búsquedas y acceso a la misma [7]. En principio, la recuperación de información engloba las acciones encaminadas a identificar, seleccionar y acceder a los recursos de información útiles para el usuario.

## 1.5 Soluciones existentes

Se realizó un análisis detallado de varias soluciones existentes en el ámbito internacional que abarcan las temáticas de la investigación, estas permiten buscar, archivar y visualizar tuits para poder acceder a ellos mediante una conexión a la plataforma Twitter.

**SnapBird** es un innovador motor de búsqueda en Twitter que realiza búsquedas de palabras claves en una cuenta determinada, es decir, aparte de buscar en la línea de tiempo público, busca en los mensajes directos enviados, en los recibidos, en los tuits de los usuarios a los que siguen y en los tuits favoritos de un usuario. Busca los últimos 100 tuits de un usuario y encuentra tuits con más de 10 días de antigüedad. Para utilizar este servicio es necesario autorizar la aplicación accediendo con la cuenta en Twitter lo que conlleva el acceso directo a Internet para utilizar dicha aplicación, este no almacena el resultado

encontrado y no cuenta con un servicio de integración con terceras aplicaciones [21]. En el Anexo 2 se muestra una vista de esta herramienta.

**TweetBackUp** es un servicio web que archiva todos los tuits diariamente y guarda el registro para luego descargarlos. Realiza la búsqueda de una determinada etiqueta o *hashtag* y guarda hasta 3200 tuits diarios que se emitan con ella. Se accede a este servicio registrándose mediante Twitter OAuth, tras el registro comenzará a descargarse los tuits. Permite la descarga de los tuits archivados en formato RSS (.xml), formato texto o html, mientras el formato CSV es reservado para cuentas premium, con otros permisos de administración de la cuenta si se abona \$1 al mes [22]. Necesita acceso directo a Internet para utilizar dicha aplicación y no facilita la integración con terceras aplicaciones. En el Anexo 3 se muestra una vista de esta herramienta.

**Twimemachine** es una herramienta que visualiza los tuits más antiguos directamente en la web. Se accede a esta mediante la cuenta de Twitter, la misma despliega un listado con los últimos 3.200 tuits publicados de un usuario. Cuenta con un buscador para encontrar tuits dentro de toda la lista que se ha generado, ofrece la opción de indicar otro usuario para mostrar sus tuits [23]. En el Anexo 3 se muestra una vista de esta herramienta.

**Twitter API** es una herramienta creada para Twitter que provee a los programadores desarrollar sus propias aplicaciones. REST API permite publicar microentradas en las aplicaciones, seguir a alguien o crear listas. Search API sirve para buscar tuits en un índice de tuits recientes, no se pueden recuperar tuits más allá de una semana y sólo se recuperan los tuits considerados relevantes dentro de la búsqueda. Streaming API es utilizada por los desarrolladores que quieren toda la secuencia de tuits en tiempo real, en el mismo momento en que se publican en Twitter [24].

En la siguiente tabla se muestra una comparación entre las soluciones existentes, los criterios de comparación están en función de las necesidades por las cuales se decide desarrollar el subsistema de almacenamiento y recuperación de tuits.

*Tabla 1: Comparación de sistemas existentes. Elaboración propia*

---

Twitter Api	SnapBird	Twimemachine	TweetBackup
-------------	----------	--------------	-------------

---

<b>Disponibilidad de los tuits</b>		1 semana	Más de 10 días	Últimos tuits	3.200	Diariamente
<b>Acceso a Internet</b>	a	Sí	Sí	Sí		Sí
<b>Integración con terceras aplicaciones</b>	con	No	No	No		No
<b>Búsqueda por términos</b>	por	Sí	Sí	Sí		Sí

Los sistemas analizados no cuentan con la totalidad de las características necesarias para su utilización en el ámbito nacional pues necesitan de una conexión directa a Internet para el almacenamiento, recuperación o visualización de los tuits, por otro lado tienen un tiempo o cantidad de tuits límites para ser recuperados, además no cuentan con un servicio para que terceras aplicaciones interactúen con el sistema. Los sistemas antes mencionados no serán utilizados por lo que se hace necesario la implementación del subsistema de almacenamiento y recuperación de tuits. Para satisfacer los problemas actuales de esta temática se contará con un ambiente de trabajo que le posibilite al equipo llevar a cabo el proceso de desarrollo del *software*.

## 1.6 Metodologías, herramientas y tecnologías de desarrollo

Elegir correctamente las herramientas, agiliza el proceso de desarrollo de *software*, la selección de estas se realiza según las necesidades del trabajo a realizar. En esta sección se mencionan y describen las herramientas, metodologías y técnicas a utilizar.

### 1.6.1 Metodología de desarrollo

Una metodología de desarrollo de *software* es una guía que muestra paso a paso las tareas y actividades que se deben ejecutar para obtener un producto informático con buena calidad. Indica cuál es el personal que debe participar en el desarrollo de las distintas actividades así como el papel que deberán enfrentar. Muestra toda la información necesaria para culminar o iniciar alguna actividad que se genere como resultado de los diferentes procesos por los cuales transcurre la construcción de un *software* [25].

Estas se pueden clasificar en dos grupos; metodologías ágiles y metodologías pesadas. Las metodologías ágiles están orientadas a la interacción con el cliente y el desarrollo incremental del *software*. Muestran al cliente versiones parcialmente funcionales del producto en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios según se desarrolla. Estas consideran más importante crear una aplicación que funcione, que escribir mucha documentación y la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan [26].

Sin embargo, las metodologías pesadas están orientadas a controlar los procesos y establecer de manera rigurosa las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Están centradas en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requieren una extensa documentación ya que pretenden prever todo de antemano [26]. En la actualidad existen diversas metodologías orientadas al proceso de desarrollo del *software*, entre las más considerables se pueden destacar *Rational Unified Process (RUP)* y *Open Unified Process (OpenUP)*.

#### **1.6.1.1 Proceso Unificado de Desarrollo**

*Rational Unified Process (RUP)* es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es un proceso de ingeniería de *software*, una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) en una organización o equipo de desarrollo de *software* [27].

Es dirigido por casos de uso, estos definen lo que el usuario desea a partir de la captura de los requisitos y la modelación del negocio. Por otro lado está centrado en la arquitectura, característica que brinda una visión completa del sistema y se describen los procesos del negocio. También es iterativo e incremental donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos para mejorar su desarrollo y comprensión [27].

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semiordenadas y con un ciclo de vida que organiza las tareas en fases e iteraciones. Divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. También define los roles que constituyen el papel que desempeña una persona en un

determinado momento. La metodología RUP es apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas [28].

### 1.6.1.2 Proceso de Desarrollo Unificado Abierto

*Open Unified Process* (OpenUP) mantiene las mismas características de RUP, pues está dirigido por casos de uso, centrado en la arquitectura y proceso iterativo e incremental [29].

El OpenUP es un proceso mínimo y suficiente, lo que significa que sólo el contenido fundamental y necesario es incluido. Por lo tanto, no provee lineamientos para todos los elementos que se manejan en un proyecto, pero tiene los componentes básicos que pueden servir de base a procesos específicos. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances [30].

OpenUP se caracteriza por cuatro principios básicos que se soportan mutuamente. Primero la colaboración para alinear los intereses y un entendimiento compartido. Por otro lado balance para confrontar las prioridades (necesidades y costos técnicos) para maximizar el valor para los stakeholders. Enfoque en articular la arquitectura para facilitar la colaboración técnica, reducir los riesgos y minimizar excesos y trabajo extra. Por último evolución continua para reducir riesgos, demostrar resultados y obtener un *software* que cumpla sus necesidades [31].

En correspondencia con el análisis realizado, se evidencia que cada metodología presenta una serie de características particulares para diferentes situaciones. Se seleccionó como apoyo en el desarrollo del *software* la metodología OpenUP con el objetivo de lograr una uniformidad de documentación en todos los componentes del subsistema de almacenamiento y recuperación de tuits. Se centra en una arquitectura temprana para reducir al mínimo los riesgos y organizar el desarrollo. Este detecta errores tempranos a través de un ciclo iterativo.

El equipo de desarrollo está compuesto por dos personas, por tanto la selección de una metodología ágil es recomendable ya que esta se aplica a proyectos de corta duración y está diseñada para pequeños equipos de trabajo. Como parte de la estructura del Centro de Ideoinformática se utiliza esta metodología

para lograr uniformidad en todas las aplicaciones a desarrollar. Esta es la que más se acoge a las necesidades del equipo de desarrollo.

## **1.6.2 Lenguaje de modelado**

El lenguaje de modelado está formado por un conjunto de símbolos que permitirán modelar parte de un diseño de *software* orientado a objetos. En la informática, como en las demás ramas de la ingeniería se necesita organizar, planificar y diseñar el trabajo antes de ejecutarlo, para así reducir las probabilidades de error en la ejecución del mismo. Los diseños se entienden mejor si se representan gráficamente, por lo que surgió la necesidad de representar los diseños del *software* que se fueran a desarrollar. Para poder estandarizar estos diseños y que fueran entendibles para todos, surge Lenguaje Unificado de Modelado (UML), dando paso a una de las premisas de la ingeniería del *software*, la reutilización [32].

### **1.6.2.1 Lenguaje Unificado de Modelado**

Se centra en la representación gráfica de un sistema. Está pensado para modelar sistemas complejos, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluye aspectos conceptuales tales como: procesos de negocio y funciones del sistema. Además, presenta aspectos concretos como: expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables [32].

UML es un método formal de modelado, que permite realizar una verificación y validación del modelo realizado. Se pueden automatizar determinados procesos y generar códigos a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto posibilita que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño de más alto nivel de la estructura de un proyecto [32].

### 1.6.3 Herramientas CASE

Las herramientas de Ingeniería de *Software* Asistida por Computadora (CASE) son aplicaciones informáticas utilizadas para apoyar las actividades del proceso de desarrollo de *software*, con el objetivo de reducir el costo en términos de tiempo y de dinero [33]. Se selecciona la herramienta Visual Paradigm dado su amplio uso internacional, sus funcionalidades y su peculiar característica de tener un entorno de creación de diagramas usando UML. Por lo antes expuesto es que se decide analizar solo esta herramienta CASE.

#### 1.6.3.1 Visual Paradigm

Es una herramienta *Unified Modeling Language* (UML) profesional que soporta el ciclo de vida completo del desarrollo del *software*: análisis, diseño, construcción, pruebas y despliegue. Este *software* de modelado ayuda a una rápida construcción de aplicaciones de calidad y menos costosas. Permite modelar todos los tipos de diagrama de clases, código inverso, genera código desde diagramas y documentos [34].

Visual Paradigm constituye una herramienta robusta. Utiliza UML como lenguaje de modelado, es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto. Permite control de versiones y realizar ingeniería tanto directa como inversa en diferentes lenguajes. Se puede representar todos los tipos de diagramas UML para las distintas fases que propone OpenUP como la captura de requisitos, análisis, diseño e implementación. Esta herramienta facilita la comunicación, ya que utiliza un lenguaje estándar común para el equipo de desarrollo [35].

Se seleccionó Visual Paradigm para el modelado de los artefactos de OpenUP porque es una herramienta multiplataforma, aparte de poseer un entorno de creación de diagramas UML. Soporta una gama de lenguajes en la generación de código e ingeniería inversa. Dentro de esta gama se encuentra el lenguaje de desarrollo Python, la integración de ambos facilitará el desarrollo de la aplicación. Es de fácil uso y genera código desde diagramas, así como generar documentación, para agilizar el trabajo del desarrollador. Por tanto Visual Paradigm se convierte en una herramienta potente para el modelado y desarrollo del producto.

## 1.6.4 Lenguaje de programación

Los lenguajes de programación son notaciones para describir la interacción entre las personas y las máquinas. Constituyen el elemento esencial en la construcción de un *software*.

Un lenguaje de programación es un idioma artificial diseñado para controlar el comportamiento de una computadora. Está compuesto por un conjunto de símbolos y reglas; la unión de estos dos componentes expresan instrucciones que luego serán interpretadas. Los lenguajes de programación pueden clasificarse según el paradigma que usan en: orientado a objetos, funcionales, lógicos y estructurales [36]. Para el desarrollo de esta aplicación se usará un lenguaje orientado a objetos, por la facilidad de uso que estos permiten.

### 1.6.4.1 Java

Es un lenguaje de programación orientado a objeto desarrollado por Sun Microsystem a principio de los años 90. El lenguaje en sí mismo toma mucho de la sintaxis de C y C++, pero tiene un modelo de objetos simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria [37].

La versatilidad de la tecnología Java, así como la portabilidad de su plataforma, y la seguridad que aporta le ha valido el aumento de su popularidad para la implementación de aplicaciones orientadas a la web. Es un lenguaje moderno, de alto nivel, que recoge los elementos de programación que típicamente se encuentran en todos los lenguajes de programación [38].

### 1.6.4.2 Python

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, compatible con la licencia GPL (*General Public License*). Es un lenguaje de scripts interpretado, de alto nivel y de propósito general, es uno de los lenguajes que más está influyendo en la escena de los programadores. Es de código abierto, multiplataforma y resulta adecuado para manipular archivos de texto. Posee infinidad de bibliotecas que permiten trabajar desde diseño dinámico de sitios web hasta la programación de escritorio y diseño de videojuegos [39].

Las principales ventajas de este lenguaje radican en la simpleza, claridad y sencillez de su sintaxis, así como en su portabilidad, flexibilidad, la cantidad de bibliotecas disponibles, su potencia y la comunidad que se mueve tras él. Es por esto que se decide utilizar Python como lenguaje de programación principal por ser potente en el manejo de archivos de texto y es el lenguaje propuesto por el cliente. Su uso garantiza la eficiencia, rapidez y el alto rendimiento de la aplicación, teniendo como basamento su potencia.

Es un lenguaje interpretado, es decir que no necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo. En todos los lenguajes de programación existen bibliotecas que sirven para hacer diversas funcionalidades y muy repetitivas a la hora de programar, que ayudan a realizar tareas habituales sin necesidad de programarlas desde cero y de esto no se excluye a Python.

### **1.6.5 Biblioteca**

Grupo o colección de funciones, clases y algoritmos que proporciona al programador facilidades para la construcción eficiente de aplicaciones [40]. Las bibliotecas de los lenguajes de programación ahorran la tarea de escribir las funciones comunes que por lo general pueden necesitar los programadores.

#### **1.6.5.1 Twython**

Twython es una biblioteca para acceder a los datos de Twitter con Python [41]. Soporta búsquedas de información de usuario, lista de Twitter, líneas de tiempo (*timeline*), mensajes directos (direct messages), otros elementos que se encuentren en la documentación del API de Twitter. Permite la actualización de imagen o actualización del estado de un usuario con una imagen, cambiar el avatar de un usuario, cambiar imagen de fondo así como cambiar el *banner*. Soporte de sólo lectura para OAuth2, para el API de stream de Twitter y para Python [42]. Por las características mencionadas anteriormente se decide utilizar esta biblioteca durante el desarrollo del trabajo.

### **1.6.6 IDE de desarrollo**

Para agilizar la construcción de aplicaciones, los desarrolladores se apoyan de un entorno de desarrollo integrado IDE (*Integrated Development Environment*). Estos programas contienen un conjunto de

herramientas de programación que permiten el desarrollo de otras aplicaciones en determinado lenguaje. Estos pueden realizar las funciones de un editor de código, un compilador, depurador y hasta un constructor de interfaz gráfica [43].

#### **1.6.6.1 JetBrains PyCharm 3.4**

Es un IDE inteligente para Python con asistencia y análisis de código para el desarrollo productivo a todos los niveles. Es una herramienta avanzada para el desarrollo profesional en la web y Python. Soporta marcos de trabajo modernos para el desarrollo web, tales como Django, Flask, Google App Engine, Pyramid y web2py. Asimismo para tecnologías multilenguaje, cubriendo a Python, JavaScript, CoffeeScript, TypeScript, HTML/CSS y Cython. Permite ejecutar, depurar y probar aplicaciones en ordenadores remotos o máquinas virtuales. Brinda soporte para bases de datos con SQL, clases y diagramas de modelos de bases de datos. Está bajo la licencia de Apache [44].

PyCharm busca incrementar la productividad de los desarrolladores dando asistencia en el código, señalamiento de errores, auto-correcciones, así como herramientas de depuración y prueba. Facilita la reestructuración de código fuente de manera automática, así como facilidades para la navegación en el mismo [45].

Se seleccionó como IDE de desarrollo a PyCharm de la suite de JetBrains por su trabajo con el lenguaje de programación Python. Este provee auto completamiento inteligente, chequeo de errores, correcciones rápidas y facilidades para la navegación en el proyecto. Se encarga de todo el trabajo rutinario, enfocándose en elementos más importantes y disfrutar mientras se escribe el código.

#### **1.6.7 Marco de trabajo para servicios web**

En el mundo el desarrollo de aplicaciones informáticas con el uso de los marcos de trabajos se vuelve inminente, ya que estas estructuras conceptuales y tecnológicas de soporte definido, constituyen la base con la cual otro proyecto de *software* puede ser fácilmente organizado y desarrollado [46].

Debido a la necesidad de estandarizar las prácticas comunes, así como los criterios y conceptos a la hora de desarrollar aplicaciones de *software*, surgen los marcos de trabajo (en inglés, *frameworks*), que

definen una estructura conceptual y tecnológica de soporte definido, a partir de artefactos o módulos de *software* previamente desarrollados, que sirven de base para la organización y construcción de un *software* [47].

En la actualidad, son muchos los marcos de trabajo que existen y que implementan en diversos lenguajes los protocolos que se usan para la creación de los servicios web. El uso y popularidad de estos están relacionados con el lenguaje sobre el que están escritos y la cantidad de protocolos que implementan.

### 1.6.7.1 Django REST Framework

Es una herramienta desarrollada en Python, para el desarrollo de servicios web basado en la arquitectura REST, de forma sencilla y rápida. Cuenta con las siguientes características [48]:

Flexible y potente.

- ✓ Proporciona acceso a las APIs creadas desde el navegador web, mejorando la usabilidad a los desarrolladores.
- ✓ Cuenta con políticas de autenticación para el acceso a los servicios web implementados.
- ✓ Extensa documentación y una gran comunidad de desarrollo.

### 1.6.7.2 Flask

Es un microframework escrito en Python bajo la licencia de *software* libre permisiva BSD (*Berkeley Software Distribution*). Flask provee, a partir de la flexibilidad ofrecida por Python, una arquitectura simple para el desarrollo web. Al ser un microframework, su núcleo no provee una capa de abstracción de bases de datos, validación de formularios u otros componentes que a menudo forman parte del núcleo de los marcos de trabajo tradicionales. En contraste con esto, Flask ofrece soporte para extensiones de todo tipo que permiten la inclusión de estas características [49].

Entre sus principales bondades se puede destacar:

- ✓ Cuenta con un servidor de desarrollo y un depurador integrado.
- ✓ Soporte integrado para pruebas unitarias.
- ✓ Soporte a peticiones utilizando la arquitectura REST.

- ✓ Extensiva documentación.
- ✓ Gran número de extensiones que añaden nuevas funcionalidades a la aplicación.

Se selecciona Flask para la implementación de la solución propuesta por su flexibilidad, ligereza y sencillez. El uso de este marco de trabajo hace que el código de la aplicación sea más compacto, al instalar solamente las extensiones necesarias para el correcto funcionamiento de la solución, lo cual se traduce en un mayor rendimiento.

### 1.6.8 JSON

Notación de Objetos de JavaScript (JSON de sus siglas en inglés *JavaScript Object Notation*) es un formato ligero de intercambio de datos, a las máquinas les resultan simple de interpretarlo y generarlo. Está basado en un subconjunto del lenguaje de programación JavaScript. Es un formato de texto completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C++, C#, Java, JavaScript, Perl, Python [50].

JSON está constituido por dos estructuras, una colección de pares de nombre/valor y una lista ordenada de valores. En varios lenguajes la colección de pares de nombre/valor es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo. Por otro lado una lista ordenada de valores, en la mayoría de los lenguajes, se implementa como arreglos, vectores, listas o secuencias.

Entre las ventajas se definen que su procesamiento por parte de los ordenadores es rápido; se necesitan bibliotecas muy pequeñas para trabajar con él (siendo posible incluso procesarlo sin biblioteca). Destacar que JSON es fácil de entender. No es más que escribir pares del tipo "nombre: valor" y asignarlos a un objeto. Al escribir en JSON no ocasiona conflictos con ningún otro script que sea llamado en la misma página [51]. Para el desarrollo de la aplicación, JSON jugará un papel fundamental pues será el formato de datos tanto en el almacenamiento como devolución de la información solicitada.

### 1.6.9 Sistema Gestor de Base de Datos

Sistema Gestor de Base de Datos (SGBD) “es un *software* diseñado para asistir al mantenimiento y utilización de extensas colecciones de datos” [52]. Es una aplicación que permite a los usuarios crear, definir y mantener las bases de datos. Un SGBD es una aplicación que facilita la gestión y mantenimiento de forma segura de la información almacenada en las bases de datos.

De manera general, los SGBD son un tipo de *software* dedicado a servir de interfaz entre las bases de datos, los usuarios y las aplicaciones que las utilizan, permitiendo así la creación y manipulación de los objetos y las propias bases de datos. Por su parte, los almacenes de datos pueden ser relacionales y no relacionales, a estos últimos también se les conoce comúnmente como bases de datos NoSQL. Dentro de los sistemas de gestión de bases de datos relacionales usados a nivel internacional se encuentran: PostgreSQL, MySQL, ORACLE y SQL-Server. Actualmente los almacenes de datos NoSQL que más sobresalen son: MongoDB, Cassandra y BigTable, todos usados en aplicaciones que manejan grandes volúmenes de datos como son: Twitter, Facebook y Google respectivamente [53].

#### 1.6.9.1 MongoDB

Es una base de datos documental, no una base de datos relacional. Es una de las bases de datos que han surgido del movimiento NoSQL (no presenta estructura de tablas y relaciones en las bases de datos) y que intenta incrementar la escalabilidad del sistema. La idea básica es sustituir el concepto de fila por el concepto de documento [54].

Un estudio enfocado en las bases de datos no relacionales arroja que MongoDB constituye una solución escalable y de alto rendimiento de almacenes de datos NoSQL. De código abierto y escrito en C++, este sistema tiene orientado el almacenamiento de datos en documentos al estilo JSON con esquemas dinámicos, que ofrecen potencia y simplicidad. Se destaca por conservar los índices de todos los atributos y hacer mucho más flexible la agregación y procesamiento de datos. Entre sus características fundamentales se encuentran [55]:

- ✓ Libre de esquemas: las claves de un documento no están predefinidas o fijas en ninguna forma, esto provee las migraciones masivas de datos y ofrece a los desarrolladores flexibilidad para el trabajo con los modelos de datos.
- ✓ Fácil escalado: su modelo de datos orientado a documentos permite que los datos sean divididos a través de la red, ofrece balanceo de datos y de carga a través del clúster.
- ✓ Almacenamiento de funciones JavaScript, la colección es de tamaño fijo, almacenamiento de ficheros y un buen rendimiento.

Presenta interfaz para varios lenguajes de programación como C++, Java, PHP, C#. Utiliza un fichero asignado en memoria (ocupa una porción de la memoria virtual) como mecanismo de almacenamiento y consulta de la información. Lo anterior hace que delegue gran parte del manejo de la memoria al sistema operativo donde se ejecuta, porque el código de MongoDB para gestionar la memoria es pequeño y claro. Las operaciones de entrada y salida son mucho más rápidas con esta variante utilizada por MongoDB. Cuenta con una comunidad estable y amplia, de experiencia en el desarrollo con la herramienta, que brinda un espacio para consultar y solucionar los problemas que se presentan en el desarrollo con MongoDB [56].

Dada las características planteadas se selecciona MongoDB para el almacenamiento de los datos relacionados con el subsistema, así como para los parámetros fundamentales del tráfico que se genere, con fines estadísticos. La selección de este sistema de almacenamiento de datos se justifica por lograr un mejor manejo de la información frente a un volumen considerable de datos.

#### **1.6.10 Servidor de indexación**

La indexación tiene como propósito ejecutar la elaboración de un índice que contenga de forma ordenada la información, proporciona la base para un acceso rápido a los registros, por lo que se convierte en una parte importante para elementos con gran volumen de información. Un índice, es una estructura de almacenamiento físico empleada para acelerar la velocidad de acceso a los datos [57].

### 1.6.10.1 Apache Solr

Es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas". Sus principales características incluyen en búsquedas de texto completo, resaltado de resultados y manejo de documentos (como Word y PDF). Solr es escalable, realiza búsquedas distribuidas y replicación de índices [58].

También posee propiedades tales como, la replicación es independiente del sistema operativo para que pueda reproducir el mismo índice en una plataforma Windows, así como en Linux. Los índices se puede configurar para operar en sincronía es decir si alguno de los índices se modifica, las demás copias de dicho índice se actualiza automáticamente. De igual importancia Solr apoya la búsqueda de indexación distribuida por la realización de diversas máquinas y la fusión de los resultados. Puede indexar y realizar búsquedas al mismo tiempo. El mismo cuenta con un esquema basado en XML para la gestión de campos indexados [58].

Permite realizar peticiones HTTP para indexar o consultar documentos es decir que tiene un API estilo REST, aunque no hace uso de todos los verbos, la recuperación de documentos es en formato XML y JSON. Incluye caches internas para devolver con mayor rapidez el resultado de las consultas y una administración web que consulta estadísticas de rendimiento incluyendo el uso de cache, realizar búsquedas mediante un formulario, navegar por los términos más populares del índice y visualizar un desglose detallado de las matemáticas de puntuación y las fases de análisis de texto. Integra el plugin de Lucene para la recuperación de contenido con resaltado de sintaxis en las coincidencias, dispone de un plugin de "*spell check*" o revisión gramatical, para realizar recomendaciones de búsqueda [59].

Está escrito en Java, pero se puede usar en cualquier lenguaje, simplemente usando las peticiones GET para realizar búsquedas en el índice, y POST para agregar documentos. Fácil de configurar y usar. La principal característica de Solr es su API estilo REST, ya que en vez de usar *drivers* para comunicarse con Solr se puede realizar peticiones HTTP y obtener resultados en JSON. Solr utiliza la biblioteca de Java Lucene, para la indexación de texto completo y de búsqueda [60].

## 1.7 Conclusiones parciales

El estudio de los principales elementos teóricos sobre las redes sociales, de Twitter, así como la importancia del almacenamiento y recuperación de tuits aportan los elementos necesarios para el trabajo y se arriba a resultados satisfactorios. Los sistemas existentes a nivel internacional no satisfacen las necesidades requeridas para el trabajo con los almacenes de tuits; todas necesitan ser accedidas mediante una conexión a Internet lo que implica el mismo uso de ancho de banda que si se utilizara Twitter directamente. El almacenamiento y recuperación de tuits en estos sistemas se realiza por un tiempo relativamente corto y cantidad limitada. La utilización de MongoDB garantiza una adecuada gestión de configuración y almacenamiento de la información. El uso de índices mediante Apache Solr como un sistema de recuperación de información permite realizar consultas para las búsquedas de tuits asociados a frases en un gran volumen de información de forma eficiente. El uso de Visual Paradigm 5.0 como herramienta CASE y la metodología de desarrollo de *software* OpenUP garantizan diseñar y elaborar la documentación necesaria para guiar el desarrollo del sistema de forma ágil y metódica. El empleo del lenguaje de programación Python y el uso de twython garantiza una conexión a Twitter de alto nivel, así como el desarrollo de las funcionalidades necesarias para almacenar información en diferentes bases de datos, y el desarrollo de servicios a terceros para establecer mecanismos de recuperación de información.

## **Capítulo 2: Análisis y diseño de la propuesta de solución**

### **2.1 Introducción**

En este capítulo se hace una descripción general de la propuesta de solución del sistema que expone esta investigación. Se plantea la especificación de los requisitos funcionales y no funcionales, se realiza el modelado de casos de uso del sistema y las descripciones textuales de estos. Se describe la arquitectura y patrones empleados. Este capítulo tiene el objetivo de diseñar el sistema a implementar y que esta contribuya a dar la solución deseada a las necesidades que dieron su origen.

### **2.2 Propuesta de solución**

El subsistema de almacenamiento y recuperación de tuits, estará enfocado en permitirle a sistemas externos interactúen con él, recibir peticiones que incluirán consultas parametrizadas donde permitirá la recuperación de tuits de la base de datos del servidor de indexación Solr. A partir de la solicitud, la aplicación generará un mensaje de respuesta con la información requerida.

Un sistema externo es la aplicación que necesita acceder a la información almacenada (tuits) en la base de datos de la aplicación (SART). La utilización de un cliente REST permitirá el intercambio de información entre el sistema externo y el subsistema de almacenamiento y recuperación de tuits (SART). Haciendo uso de este, se envía una petición a la aplicación a través del protocolo HTTPS, recibiendo como respuesta la información solicitada en formato JSON. El API REST se encarga de procesar las consultas hechas a la aplicación. Evalúa los parámetros recibidos, accede al servidor de indexación, siendo este el que contiene en su base de datos los tuits. Esto se representa en la figura 1.

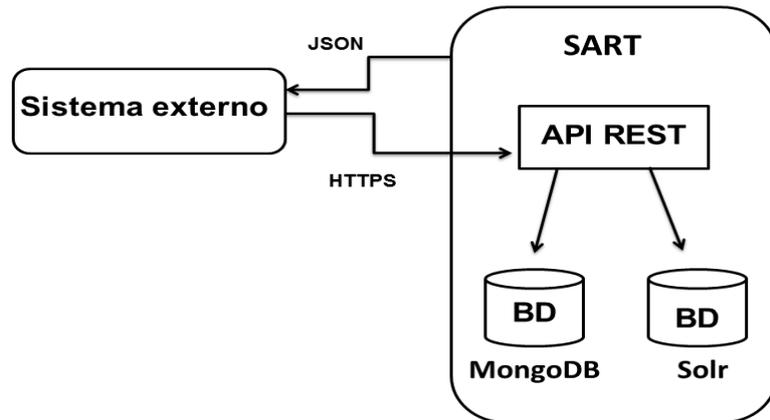


Figura 1: Propuesta de solución del sistema. Elaboración propia

### 2.3 Modelo de dominio

Es una representación visual estática del entorno real del proyecto. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis, como paso previo al diseño de un sistema, ya sea de *software* o de otro tipo. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema y ayudar a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común [61]. El modelo de dominio se representa en UML con un diagrama de clases en los que se muestra:

- ✓ Conceptos u objetos del dominio del problema: clases conceptuales.
- ✓ Asociaciones entre las clases conceptuales.
- ✓ Atributos de la clase conceptuales.

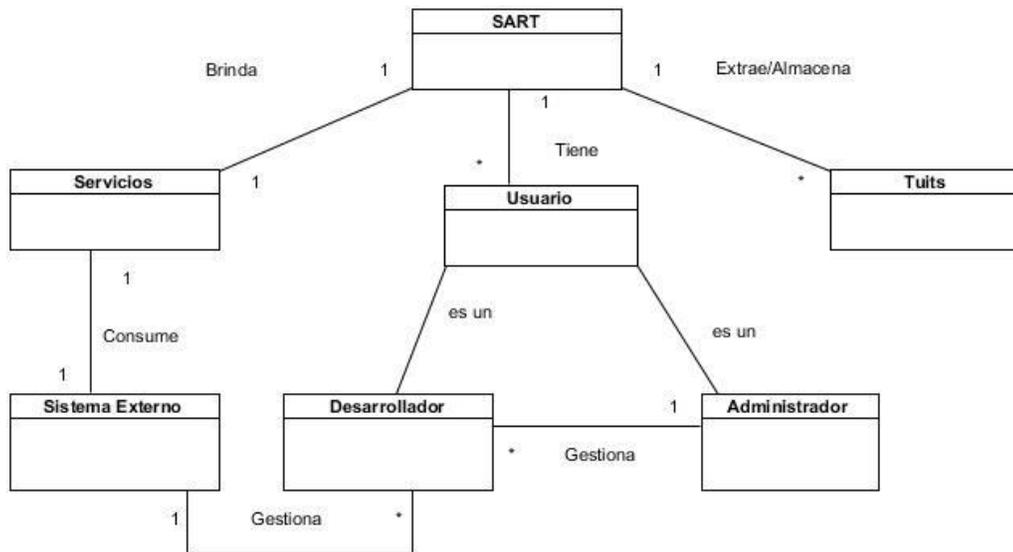


Figura 2: Modelo de dominio

SART es el subsistema de almacenamiento y recuperación de tuits el cual brinda servicios a los sistemas externos, este a su vez tiene a desarrollador y administrador como usuarios, el desarrollador es gestionado por el administrador y este es el que gestiona los sistemas externos. SART extrae los tuits de la red social Twitter y los almacena. Un sistema externo es la aplicación informática que va a consumir los servicios que brinda SART.

## 2.4 Requerimientos del sistema

Requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto [62]. A continuación se presentan los requerimientos que debe cumplir la aplicación a desarrollar. Estos se han clasificado en funcionales y no funcionales.

### **2.4.1 Requisitos funcionales**

Un requerimiento funcional es una condición o capacidad que debe ser alcanzada o poseída por un sistema o componente de un sistema, para satisfacer un contrato, estándar u otro documento impuesto formalmente [61]. A partir del estudio y las investigaciones que se realizaron de los procesos, se obtuvieron una serie de requerimientos funcionales:

- RF1: Autenticar usuario
- RF2: Registrar usuario
- RF3: Modificar usuario
- RF4: Eliminar usuario
- RF5: Mostrar usuarios
- RF6: Recuperar cuenta de usuario
- RF7: Registrar rol
- RF8: Eliminar rol
- RF9: Mostrar roles
- RF10: Modificar rol
- RF11: Consultar tuits
- RF12: Registrar sistema externo
- RF13: Eliminar sistema externo registrado
- RF14: Modificar sistema externo registrado
- RF15: Mostrar sistema externo registrado
- RF16: Mostrar catálogo de servicios
- RF17: Mostrar página de contacto
- RF18: Mostrar página sobre nosotros

### **2.4.2 Requisitos no funcionales**

Los requerimientos no funcionales son capacidades o cualidades que el producto debe tener, o sea, características que hagan al producto atractivo, rápido, usable o confiable. Están estrechamente vinculados a los requisitos funcionales, puesto que una vez que está definido lo que el sistema debe

hacer, es necesario especificar como ha de hacerlo. Pueden llegar a marcar la diferencia entre un producto bien aceptado por los clientes y usuarios o uno de poca o ninguna calidad y aceptación [61].

### **Requerimientos de Apariencia o Interfaz Externa**

- ✓ Diseño orientado para que terceras aplicaciones se integren con el sistema.
- ✓ La interfaz debe tener un diseño uniforme para todos los usuarios.

### **Requerimientos de Seguridad**

- ✓ Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del sistema.
- ✓ Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.
- ✓ Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de acceso del usuario activo.
- ✓ Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- ✓ Verificación sobre acciones irreversibles (eliminaciones).

### **Requerimientos de Usabilidad**

- ✓ Los requisitos de hardware para la ejecución de la aplicación serán: CPU Core2Duo 2.0 GHz o superior, 2 Gb RAM o superior, 2TB de espacio libre en disco.
- ✓ Para la ejecución de la aplicación deberán estar instaladas las siguientes aplicaciones: Ubuntu Server 14.04 LTS, Python 2.7.6, Apache Solr 4.10.3, MongoDB 2.4.9.

### **Requerimientos de Soporte**

- ✓ El sistema deberá ser fácil de instalar.
- ✓ Se permitirá realizar modificaciones posteriores para adaptar mejoras al sistema o en caso que cambien las necesidades de los clientes.
- ✓ El sistema estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

### **Requerimientos de Disponibilidad**

- ✓ El sistema deberá estar disponible las 24 horas del día para trabajar en el momento deseado, aunque puede ser a consideración de los clientes.

## **Requerimientos de Confiabilidad**

- ✓ Se deben establecer distintos roles que limiten el nivel de accesibilidad de los usuarios.
- ✓ Solamente los usuarios registrados pueden modificar la configuración de la aplicación.
- ✓ El acceso a los servicios provistos por la solución será a través del protocolo HTTPS.

## **Restricciones de diseño**

- ✓ La plataforma se debe desarrollar sobre el marco de trabajo que requiere el lenguaje Python en su versión 2.7.
- ✓ La aplicación se desarrolla sobre el sistema operativo Ubuntu 14.04 LTS.
- ✓ Como herramienta para la gestión, procesamiento, almacenamiento y consulta de la base de conocimiento se debe utilizar MongoDB 2.4.9.
- ✓ El sistema se organiza utilizando el marco de trabajo de desarrollo para servicios web Flask 0.10.1.
- ✓ Uso de la biblioteca twython para establecer conexión con Twitter.
- ✓ Para agilizar las búsquedas y el almacenamiento de los tuits en la base de datos se utilizará el servidor de indexación Apache Solr 4.10.3.

## **Requisitos legales, de Derecho de Autor y otros**

- ✓ Las herramientas que se utilizan para el desarrollo del producto están basadas en licencias GNU/GPL.

## **Requisitos para la documentación de usuarios en línea y ayuda del sistema**

- ✓ El sistema cuenta con un manual de usuario y de instalación.

## **2.5 Modelo de casos de uso del sistema**

En esta sección se identifican los actores y casos de uso, lo que resultó el diagrama de casos de uso del sistema y la descripción de cada uno de estos casos de uso.

### **2.5.1 Definición de los actores del sistema**

Los actores del sistema intercambian información con él, aunque no forman parte de este. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. A continuación se muestra el actor y la justificación que tiene en el sistema.

Tabla 2: Actores del sistema. Elaboración propia

Actores del Sistema	Justificación
Usuario	Persona que accede al sistema.
Administrador	Usuario autorizado para gestionar la configuración del sistema.
Desarrollador	Usuario que gestiona los sistemas externos los cuales consumen los servicios que provee la aplicación.

### 2.5.2 Definición de los casos de usos del sistema

El diagrama de casos de uso del sistema es el encargado de la representación del o los actores que van a estar presentes en la aplicación e interactuarán con los casos de uso o las funcionalidades que esta proveerá. Se muestran las relaciones entre casos de uso ya sean de extensión, inclusión, asociación o generalización/especialización.

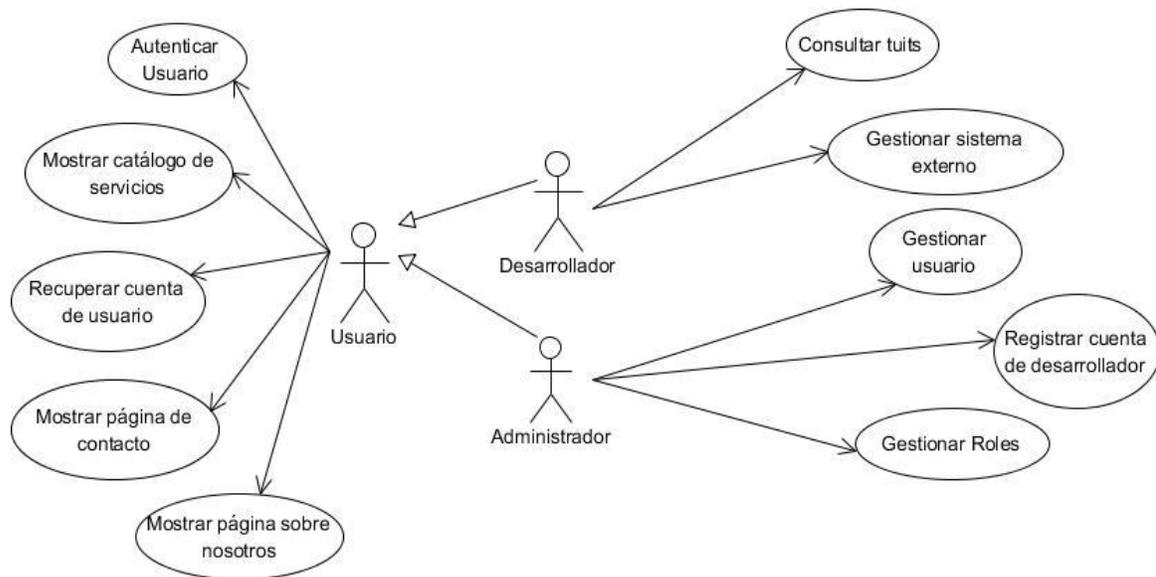


Figura 3: Diagrama de Casos de Uso del sistema

### 2.5.3 Descripción de los casos de usos del sistema

La especificación de los Casos de Uso del sistema, propicia una mayor comprensión de los procesos del negocio, tanto por parte del cliente como de los desarrolladores. Se describe de forma textual como se

llevan a cabo las actividades dentro del sistema y quienes las realizan [63]. A continuación se muestra la descripción de los casos de uso del sistema a implementar.

Tabla 3: Descripción del caso de uso del sistema “Gestionar sistemas externos”.

<b>Objetivo</b>	El objetivo que persigue el actor es poder registrar, modificar o eliminar un sistema externo y mostrar los sistemas externos existentes.	
<b>Actores</b>	Desarrollador (Inicia)	
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea registrar, modificar, eliminar o mostrar los sistemas externos que desean conectarse a la aplicación. Termina cuando se haya realizado alguna de estas operaciones.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El desarrollador debe estar autenticado en el sistema.	
<b>Postcondiciones</b>	Se gestionan los sistemas externos asociados a ese desarrollador.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Gestionar sistemas externos&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón Administración de la página principal.	Muestra la interfaz de la página de administración, con opción de desconectarse, crear sistema externo o ir a la página principal.
2.	Hace clic en el botón Sistema externo	Muestra los sistemas externos creados por el desarrollador en caso de existir.
3.	Selecciona una de las siguientes acciones: <ul style="list-style-type: none"> <li>✓ Crear (Ir a la Sección 1 Crear sistema externo).</li> <li>✓ Editar (Ir a la Sección 2 Editar sistema externo).</li> <li>✓ Eliminar (Ir a la Sección 3 Eliminar sistema externo).</li> </ul>	Finaliza el CU.
<b>Sección 1: “Crear sistema externo”</b>		
<b>Flujo básico &lt; Crear sistema externo &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón “Crear”	Muestra la interfaz de adicionar sistema

		externo.
2.	Llena los campos para registrarse: <ul style="list-style-type: none"> <li>✓ Nombre</li> <li>✓ Palabra clave</li> </ul> Luego guarda los datos haciendo clic en el botón Guardar, o Guardar y continuar para guardar los datos y quedarse en la sección.	Valida los datos, y si son correctos crea la nueva aplicación. Finaliza el CU.
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Valores de campos vacíos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Deja campos obligatorios sin llenar.	Muestra un mensaje que se requiere el llenado de los campos.
<b>Nº 2 &lt; Opción Cancelar &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Cancela la acción presionando el botón "Cancelar"	Cancela la operación realizada, cierra el formulario. Finaliza el CU.
<b>Sección 2: "Editar sistema externo"</b>		
<b>Flujo básico &lt; Editar sistema externo &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón "Editar" de cualquier sistema registrado.	Muestra la interfaz de editar, con los campos que el desarrollador puede modificar.
2.	Realiza los cambios necesarios y hace clic en "Guardar" o "Guardar y Continuar".	Valida los datos entrados y si son correctos hace los cambios solicitados por el desarrollador. Termina el CU.
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Opción Cancelar &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Cancela la acción presionando el botón "Cancelar"	Cancela la operación realizada, cierra el formulario. Termina el CU.
<b>Sección 3: "Eliminar sistema externo"</b>		
<b>Flujo básico &lt; Eliminar sistema externo &gt;</b>		

	Actor	Sistema
1.	Hace clic en el botón "Eliminar" de cualquier sistema registrado.	Muestra una notificación "¿Está seguro que desea borrar esta entrada?"
	Hace clic en "Aceptar"	Elimina este sistema. Termina el CU
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Opción Cancelar &gt;</b>		
	Actor	Sistema
1.	Cancela la acción presionando el botón "Cancelar"	Cancela la operación realizada. Termina el CU.
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Opción Eliminar todas las entradas &gt;</b>		
	Actor	Sistema
1.	Selecciona los sistemas externos que desea eliminar haciendo clic en el botón "Seleccionar entrada" de cada uno de los sistemas, o si desea borrarlos todos hace clic en el botón de "Seleccionar todas las entradas", luego despliega el botón "Con la selección" y hace clic en Eliminar.	Muestra una notificación "¿Está seguro que desea eliminar los registros seleccionados?".
2.	Da clic en el botón Aceptar	Elimina los sistemas externos seleccionados. Termina CU
<b>Relaciones</b>	<b>CU incluidos</b>	ninguno
	<b>CU extendidos</b>	ninguno

Tabla 4: Descripción del caso de uso del sistema " Consultar tuits"

<b>Objetivo</b>	A partir de una solicitud de búsqueda de tuits devuelve los tuits correspondiente a esa solicitud.
<b>Actores</b>	Desarrollador (Inicia)
<b>Resumen</b>	Se inicia cuando el sistema externo realiza una solicitud, el sistema devuelve en formato JSON los tuits que coincidan con las palabras claves proporcionadas.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	El desarrollador debe estar autenticado en el sistema.

<b>Postcondiciones</b>	Se muestra un archivo en formato JSON con los tuits.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Devolver tuits dada una consulta&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Realiza una petición al sistema, proporcionando una o varias palabras claves y la cantidad de tuits que desea obtener.	Muestra en formato JSON la cantidad de tuits solicitada por el actor que tengan en su cuerpo la(s) frase(s) entradas. Finaliza el CU.
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Devolver tuits &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Solicita a la aplicación sin especificar palabra claves, ni cantidad de tuits.	Muestra por defecto los primeros 20 tuits almacenados. Finaliza CU.
<b>Relaciones</b>	<b>CU incluidos</b>	ninguno
	<b>CU extendidos</b>	ninguno

Tabla 5: Descripción del caso de uso del sistema "Gestionar roles"

<b>Objetivo</b>	El objetivo que persigue el actor es poder insertar, modificar, eliminar y mostrar los roles del sistema.	
<b>Actores</b>	Administrador (Inicia)	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario desea insertar, modificar, eliminar o mostrar los diferentes roles del sistema. Termina cuando se haya realizado alguna de estas operaciones.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Media	
<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema.	
<b>Postcondiciones</b>	Se gestionan los roles del sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Gestionar roles&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón Administración de la página principal.	Muestra la interfaz de la página de

		administración, con opción de desconectarse, crear sistema externo, crear rol, gestionar usuario o ir a la página principal.
2.	Hace clic en el botón Roles	Muestra los roles existentes.
3.	Selecciona una de las siguientes acciones: <ul style="list-style-type: none"> <li>✓ Insertar (Ir a la Sección 1 Crear rol).</li> <li>✓ Modificar (Ir a la Sección 2 Modificar rol).</li> <li>✓ Eliminar (Ir a la Sección 3 Eliminar rol).</li> </ul>	Finaliza el CU.
<b>Sección 1: “Crear rol ”</b>		
<b>Flujo básico &lt; Crear rol &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón “Crear”	Muestra la interfaz de crear rol.
2.	Llena los campos para insertar: <ul style="list-style-type: none"> <li>✓ Nombre</li> <li>✓ Nombre a mostrar</li> <li>✓ Descripción</li> </ul>	Valida los datos, y si son correctos crea el nuevo rol. Finaliza el CU.
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Valores de campos vacíos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Deja campos obligatorios sin llenar.	Muestra un mensaje que se requiere el llenado de los campos.
<b>Nº 2 &lt; Opción Cancelar &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Cancela la acción presionando el botón “Cancelar”	Cancela la operación realizada, cierra el formulario. Termina el CU.
<b>Sección 2: “Modificar rol”</b>		
<b>Flujo básico &lt; Modificar rol &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace clic en el botón “Editar” de cualquier rol registrado.	Muestra la interfaz de editar roles, con los campos que el actor puede modificar.
2.	Realiza los cambios necesarios y hace clic en	Valida los datos entrados y si son correctos hace

	“Guardar” o “Guardar y continuar”.	los cambios solicitados por el actor. Finaliza el CU
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Opción Cancelar &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Cancela la acción presionando el botón “Cancelar”	Cancela la operación realizada, cierra el formulario. Termina el CU.
<b>Sección 3: “Eliminar rol”</b>		
<b>Flujo básico &lt; Eliminar rol &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona los roles que desea eliminar haciendo clic en el botón “Seleccionar entrada” de cada uno de los roles, o si desea borrarlos todos hace clic en el botón de “Seleccionar todas las entradas”, luego despliega el botón “Con la selección” y hace clic en Eliminar.	Muestra una notificación “¿Está seguro que desea eliminar los registros seleccionados?”.
2.	Da clic en el botón Aceptar	Elimina los roles seleccionados. Termina CU
<b>Flujos alternos</b>		
<b>Nº 1 &lt; Opción Cancelar &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Cancela la acción presionando el botón “Cancelar”	Cancela la operación realizada. Finaliza el CU.
<b>Relaciones</b>	<b>CU incluidos</b>	ninguno
	<b>CU extendidos</b>	ninguno

Los restantes Casos de Uso asociados a la solución propuesta se describen en el Anexo 3.

## 2.6 Arquitectura del sistema

La arquitectura de *software* demuestra la organización, funcionamiento y conexión entre las partes de un sistema. Su objetivo principal es garantizar un mejor desempeño en el desarrollo de las aplicaciones. Proporciona robustez, portabilidad y flexibilidad a la aplicación. Es considerado el elemento de enlace entre los requerimientos y la implementación del sistema [64].

Para el desarrollo de la solución propuesta se decide utilizar el estilo arquitectónico N-Capas. En este estilo los componentes son las capas o niveles, que pueden estar implementadas internamente tanto por objetos como por procedimientos. Cada nivel tiene asociada una funcionalidad: los niveles bajos implementan funciones simples, ligadas al hardware o al entorno, mientras que los niveles altos implementan funciones más abstractas. El mecanismo de interacción entre componentes es el de llamadas a procedimientos (métodos), con la restricción de que son las capas superiores las que invocan funciones o métodos de las inferiores [65].

El uso de este estilo permite aprovechar las siguientes ventajas:

- ✓ Abstracción: no es necesario conocer el funcionamiento interno de otras capas, solo la forma de interacción con estas.
- ✓ Aislamiento: los cambios internos de una capa no afectan el sistema completo.

La aplicación a implementar cuenta con tres capas, descritas a continuación:

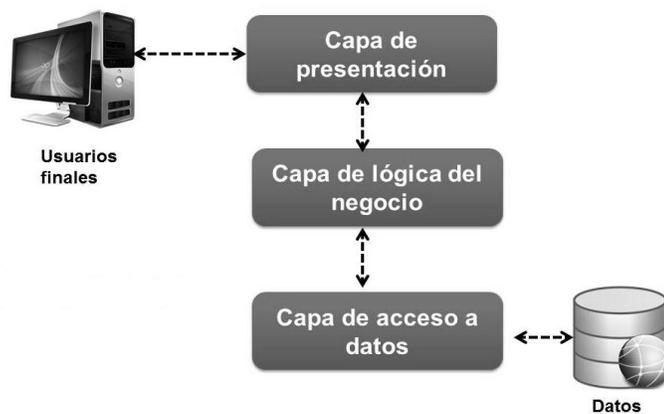


Figura 4: Modelo N-Capas del sistema

La **capa de presentación** o también conocida como “capa de usuario”, es la responsable de comunicar y capturar información al usuario e interpretar sus acciones. Se comunica únicamente con la **capa de lógica del negocio**. Mientras esta se encarga de recibir las peticiones hechas por los usuarios, así como de emitir una respuesta a estas. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de bases

de datos almacenar o recuperar datos de él. Por otro lado la **capa de acceso a datos** es la encargada de persistir las entidades que se manejan en el negocio, el acceso a los datos almacenados y reúne todos los aspectos del *software* que tienen que ver con el manejo de los datos persistentes. Es donde se gestiona el almacenamiento físico y recuperación de datos. Ayuda a separar el acceso a los datos de la lógica del negocio, así como tampoco contiene elementos de la interfaz de usuario. Además debe cumplir los requerimientos de la aplicación a nivel de rendimiento, seguridad, mantenibilidad y soportar cambios de requerimientos de negocio.

## 2.7 Patrones de diseño

Los patrones de diseño permiten la reutilización de diseños exitosos, basando nuevos diseños en experiencias previas. Estos patrones resuelven problemas específicos de diseño, y vuelven el diseño orientado a objetos más flexible, elegante y extremadamente reutilizable [63]. Estos patrones pueden ser clasificados en dos categorías: *GRASP* es un acrónimo de *General Responsibility Assignment Software Patterns* (en español, patrones generales de *software* para asignar responsabilidades) y *GoF* es un acrónimo de *Gang-Of-Four* (en español, La pandilla de los cuatro).

Una buena práctica en el desarrollo de un software es auxiliarse de los patrones de diseño; es por ello que el funcionamiento del marco de trabajo Flask y la implementación de las librerías utilizadas hacen uso de patrones de diseños característicos de sistemas Orientados a Objetos.

Los **patrones GRASP** constituyen un apoyo para ayudar a los autores a entender el diseño de objetos esenciales, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos [32].

Los siguientes patrones GRASP son los utilizados en el sistema:

- ✓ **Alta cohesión:** define que la información almacenada en una clase debe ser coherente y estar relacionada con esta. Propone que no se debe saturar una clase de métodos, sino asignar las responsabilidades a cada clase correspondiendo a la información que almacena [32]. Este patrón

se evidencia en todas las clases del subsistema, incrementa la claridad y facilita la comprensión del diseño. Simplifica el mantenimiento y las mejoras.

- ✓ **Bajo acoplamiento:** el uso de este patrón garantiza que las clases estén lo menos ligadas posible entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases [32]. Este patrón se utiliza en todas las clases del subsistema, el uso de este asegura que los cambios en un componente no afecten a otros.

Con la utilización de una arquitectura N Capas se contribuye a mantener una Alta Cohesión y un Bajo Acoplamiento ya que las clases no se encuentran saturadas de responsabilidades, presentan poca dependencia entre sí, de forma que en caso de realizar cambios en una capa se tenga la mínima repercusión posible en las otras capas.

- ✓ **Experto en información:** se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Expresa la intuición común de que los objetos hacen las cosas relacionadas con la información que tienen. Se mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Normalmente, esto conlleva un bajo acoplamiento, lo que da lugar a sistemas más robustos y más fáciles de mantener. Se distribuye el comportamiento entre las clases que contienen la información requerida, por tanto, se estimula las definiciones de clases más cohesivas y ligeras que son más fáciles de entender y mantener [32]. Este se evidencia en las clases dentro del componente “models” de la aplicación, en los paquetes “client” y “security”, las cuales contienen la información necesaria para las operaciones que implementan. El uso de este patrón contribuye a un adecuado encapsulamiento, lo que favorece la robustez y facilidad de mantenimiento del sistema. A continuación el uso de este patrón se muestra en las figuras correspondientes al Anexo 4.

```

class Client(db.Document):
    client_id = db.StringField(max_length=40, unique=True)
    client_secret = db.StringField(max_length=55)
    _redirect_uris = db.StringField()
    _default_scopes = db.StringField()
    user_id = db.StringField()
    name = db.StringField(max_length=40, required=True, verbose_name='Nombre')
    phrase = db.ListField(db.StringField(max_length=150, required=True), required=True,
                          verbose_name='Palabra clave')

```

Figura 5: Ejemplo de utilización del patrón Experto en información

- ✓ **Creador:** se asigna la responsabilidad a una clase de crear cuando contiene, agrega, compone, almacena o usa otra clase, esto brinda una alta posibilidad de reutilizar la clase creadora [32]. La utilización de este patrón se evidencia en el componente “kernel.py” donde se inicializa todos los módulos del sistema. Ver ejemplo en las figuras correspondientes al Anexo 4.

Los **patrones GoF** son en el campo del diseño orientado a objetos los más conocidos y usados en la actualidad. Estos describen las formas en las que pueden ser organizados los objetos para trabajar unos con otros [32]. Para la implementación de la solución propuesta se utilizó el patrón active record que se explica a continuación:

- ✓ **Active Record:** es un patrón en el cual, el objeto contiene los datos que representan una colección, además de encapsular la lógica necesaria para acceder a la base de datos. De esta forma el acceso a datos se presenta de manera uniforme a través de la aplicación. Una clase Active Record consiste en el conjunto de propiedades que representa las columnas de la tabla, más los típicos métodos de acceso como las operaciones CRUD (en inglés: *Create, Read, Update and Delete*), búsqueda, validaciones, y métodos de negocio [66]. El uso de este patrón se demuestra en la clase *Client* que hereda de la clase *Document* de mongoengine la cual tiene una función *save* que al crear un nuevo modelo cliente se guarda con esta información en la base de datos. Ver ejemplo en las figuras correspondientes al Anexo 4.

## 2.8 Modelo de diseño

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación. Está compuesto por clase del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos. El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en resumen: como cumple el sistema sus objetivos [67]. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

### 2.8.1 Diagrama de clase del diseño

Un diagrama de clase del diseño es un tipo de diagrama estático que describe la estructura de un sistema; mostrando sus clases, atributos, métodos y las relaciones entre ellos. Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. Teniendo en cuenta la complejidad que el marco de trabajo utilizado incorpora en los diagramas, se procede a la elaboración de los diagramas de clase del diseño solo con las clases que son desarrolladas o modificadas por los autores de la investigación. A continuación se muestran para cada uno de los casos de uso:

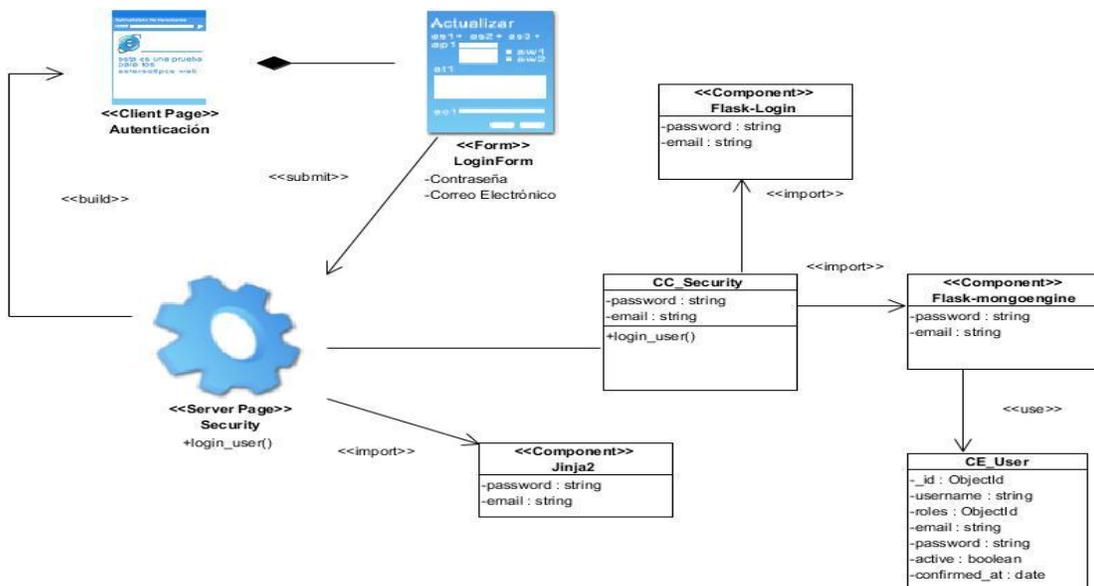


Figura 6: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Autenticar usuario

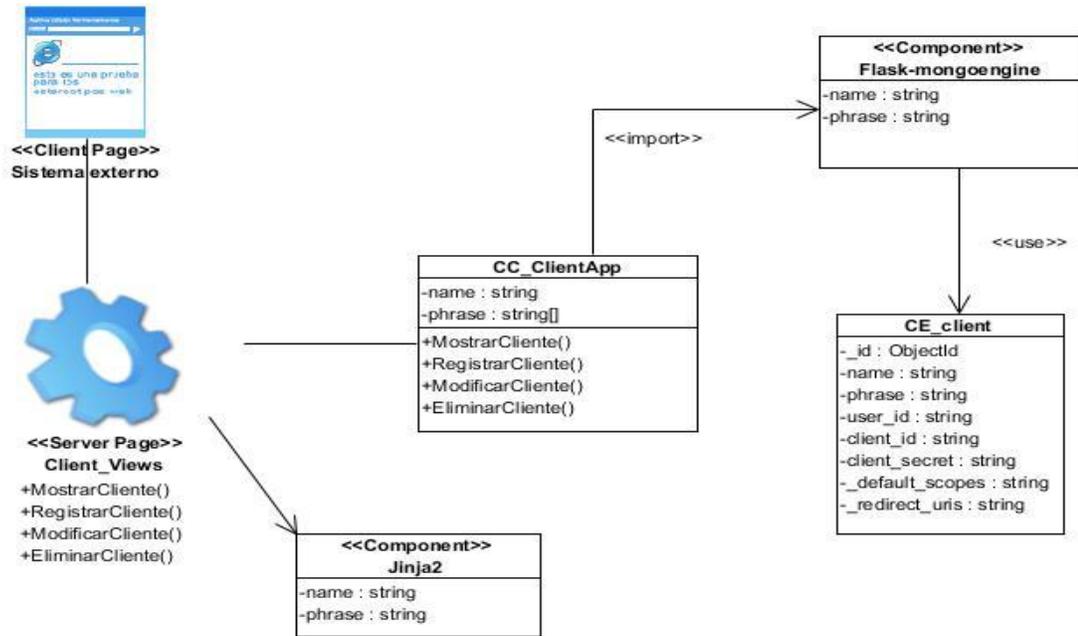


Figura 7: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Gestionar sistema externo

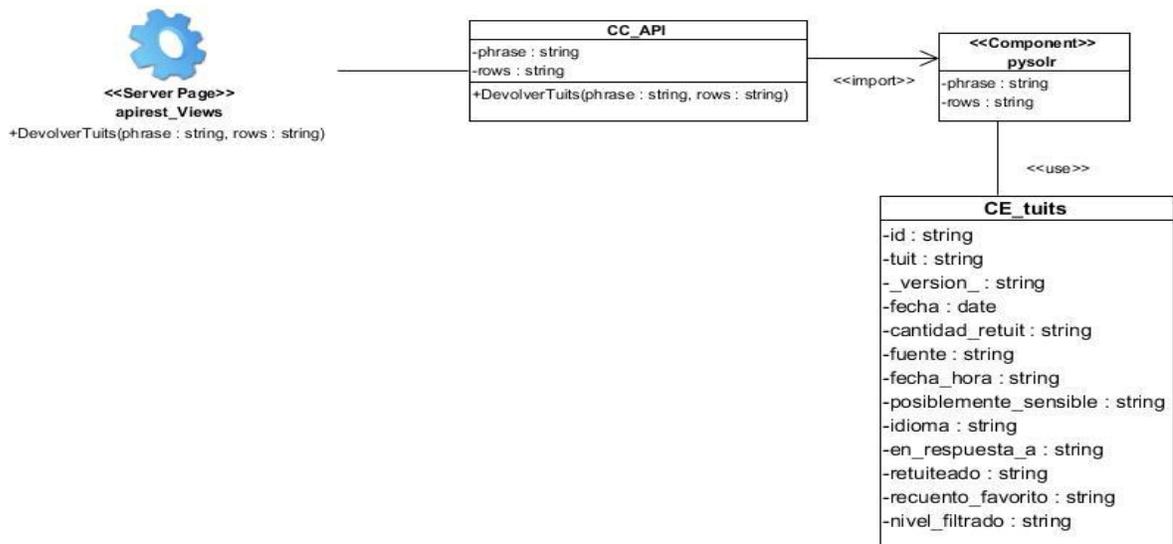


Figura 8: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Consultar tuits

Los restantes Diagramas de clases de diseño asociados a la solución propuesta se describen en el Anexo 5.

## 2.9 Modelo de base de datos

Uno de los pasos cruciales en la construcción de una aplicación que requiera manejar la persistencia de la información, es sin duda, el diseño de la base de datos, la cual, correctamente diseñada, permite obtener acceso a información exacta y actualizada. El sistema que se propone hace uso de una base de datos ya que es necesario persistir una cantidad considerable de información relacionada con los tuits almacenados.

Como se definió anteriormente, el gestor encargado de almacenar la información de la gestión de usuarios y sistemas externos por la aplicación es MongoDB, una base de datos documental escalable y de alto rendimiento. Mientras para almacenar la información referente a los tuits será el servidor de indexación Solr. El subsistema de almacenamiento y recuperación de tuits estará compuesta por dos bases de datos. La base de datos para la gestión de usuarios y sistemas externos por MongoDB contiene tres colecciones de datos:

- ✓ La colección “role” almacena la información referente a los permisos que se le asignan a los usuarios que acceden a la aplicación teniendo en cuenta el rol que tengan.
- ✓ La colección “user” almacena la información de los usuarios creados, que tienen permiso de acceder a la aplicación.
- ✓ La colección “client” guarda toda la información relacionada con los sistemas externos creados por un usuario determinado.

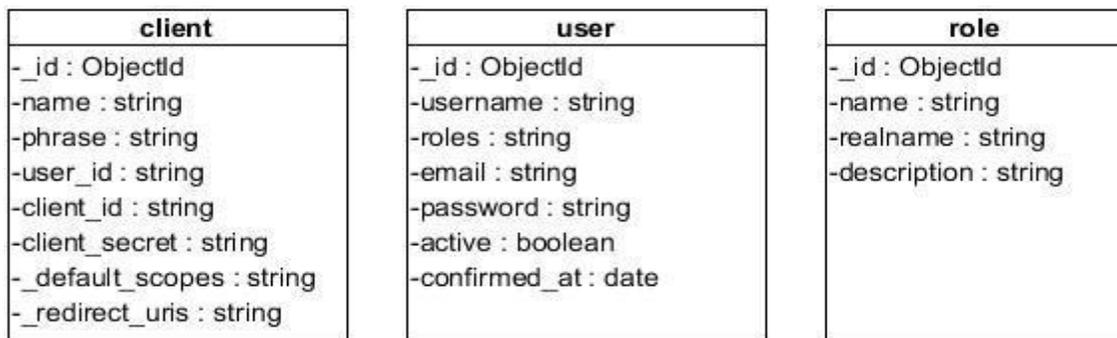


Figura 9: Modelo físico de datos para MongoDB

La base de datos para el almacenamiento de tuits por Solr contiene una colección de datos:

- ✓ La colección “tuits” almacena la información de los tuits descargados.

tuits
-id : string
-tuit : string
-_version_ : string
-fecha : date
-cantidad_retuit : string
-fuente : string
-fecha_hora : string
-posiblemente_sensible : string
-idioma : string
-en_respuesta_a : string
-retuiteado : string
-recuento_favorito : string
-nivel_filtrado : string

Figura 10: Modelo físico de datos para Solr

## 2.10 Conclusiones parciales

En este capítulo se presenta la descripción de la solución propuesta la cual ofrece una mejor comprensión del subsistema que se va a implementar. La elaboración del modelo de dominio brinda una visión más clara de los componentes y los conceptos asociados al subsistema de almacenamiento y recuperación de tuits, así como las relaciones entre estos. La creación del diagrama de casos de uso y la descripción de los mismos sirve de base para las próximas etapas de desarrollo. Se especifican las funcionalidades que debe cumplir el subsistema para satisfacer las necesidades del cliente. La arquitectura N-Capas seleccionada permitió definir las capas en las que está estructurado el sistema. Como patrones de diseño se usaron los GRASP y los GOF, entre ellos alta cohesión, bajo acoplamiento, experto en información, controlador, decorador e instancia única. Se representa la estructura estática y dinámica del sistema, a través del diagrama de clase del diseño. También se describen las clases más relevantes con el propósito de lograr un mejor entendimiento de las mismas.

## Capítulo 3: Implementación y validación de la solución propuesta

### 1.1 Introducción

En este capítulo se realiza el diagrama de componentes para estructurar el modelo de implementación y se muestra la ubicación física de los nodos de procesamiento a través del diagrama de despliegue. Se describen las pruebas a seguir, las cuales tienen el objetivo de validar la propuesta de solución de forma tal que demuestren su correcto funcionamiento según los requerimientos establecidos. Se explican los estilos de códigos utilizados con el propósito de lograr un mayor entendimiento del código. Se muestran y describen los resultados obtenidos en el proceso de ejecución de las pruebas aplicadas.

### 1.2 Modelo de implementación del sistema

Es la realización de las especificaciones técnicas o algoritmos como un programa, componente de *software*, u otro sistema de cómputo. Está conformado por los diagramas de componentes y de despliegue, describen cómo los elementos del modelo de diseño se desarrollan en términos de componentes, ficheros de código fuente o ejecutables.

#### 1.2.1 Diagrama de componentes

Es un diagrama desarrollado en el lenguaje unificado de modelado (UML). Este diagrama representa cómo un sistema de *software* es dividido en componentes y muestra las dependencias entre estos. Los diagramas de componentes ilustran las piezas del *software* y controladores embebidos que conformarán un sistema [68]. Permite modelar la vista estática del sistema mostrando dependencias lógicas entre un conjunto de componentes de *software*.

La siguiente figura muestra el diagrama de la aplicación desarrollada. Se puede apreciar la estructura de los paquetes y la distribución de los componentes dentro de estos. De forma general cada componente de los paquetes "templates" y "lib" van a ser utilizados por un componente dentro del paquete "mods", este se relacionará con su respectivo componente dentro del paquete "vars", todos ellos a su vez utilizarán los componentes del paquete "library" así como del componente "config.py" en dependencia de sus necesidades. En el componente "kernel.py" se registrarán todos los paquetes de "mods" haciendo

funcionar el sistema como un todo, este a su vez utilizará para esto las configuraciones necesarias del componente "config.py".

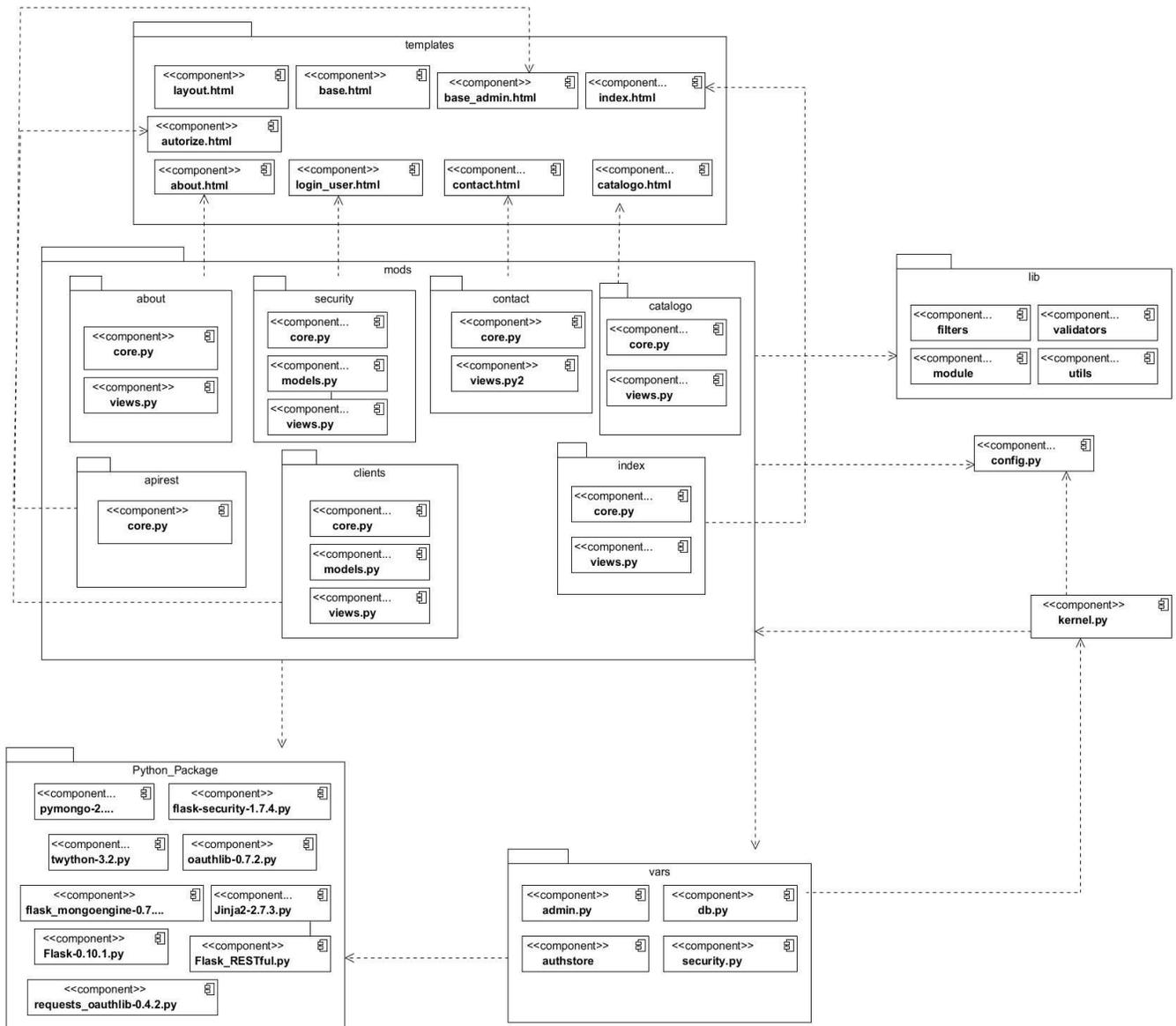


Figura 11: Diagrama de componentes del subsistema de almacenamiento y recuperación de tuits

### 1.2.2 Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez [69].

Para facilitar el entendimiento del código y fijar un modelo a seguir, se establecieron estándares de codificación. Los aspectos para los que generalmente se establecen estándares son los siguientes:

#### **Indentación:**

El lenguaje de programación Python usa indentación para indicar estructuras de control, por tanto se requiere obligatoriamente una buena indentación. Haciendo esto, la necesidad de marcar con llaves ({ y }) es eliminada, y la legibilidad es mejorada sin interferir con los estilos de codificación comunes. Se pueden utilizar tanto espacios como tabuladores para indentar el código, pero se recomienda no mezclarlos.

#### **Comentarios:**

Los comentarios se pueden poner de dos formas. La primera y más apropiada para comentarios largos es utilizar la notación `''' comentario '''`, tres apóstrofes de apertura y tres de cierre. La segunda notación utiliza el símbolo `#`, y se extienden hasta el final de la línea.

#### **Variables:**

Las variables se definen de forma dinámica, lo que significa que no se tiene que especificar cuál es su tipo de antemano y puede tomar distintos valores en otro momento, incluso de un tipo diferente al que tenía previamente. Se usa el símbolo `=` para asignar valores.

#### **Funciones:**

Las funciones se definen con la palabra clave `def`, seguida del nombre de la función y sus parámetros.

#### **Importar:**

Los `imports` deberán colocarse en distintas líneas, por ejemplo:

```
import os
import sys
```

Los *imports* se colocan siempre en la parte superior del archivo, justo después de cualquier comentario o cadena de documentación del módulo, antes de las variables globales y las constantes del módulo.

### Clases:

Casi sin excepciones, los nombres de clases usan la convención CapWords. Las clases de uso interno tienen un guión bajo al principio del nombre. Las clases se definen con la palabra clave `class`, seguida del nombre de la clase y, si hereda de otra clase, el nombre de esta.

"`__init__`" es un método especial que se ejecuta al instanciar la clase, se usa generalmente para inicializar propiedades y ejecutar métodos necesarios.

### 1.2.3 Modelo de despliegue

Es la representación física de los nodos, es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar [61]. El modelo de despliegue provee un modelado detallado de la forma en la que los componentes se desplegarán. Detalla las capacidades de red, las especificaciones del servidor y otra información relacionada al despliegue del sistema propuesto.

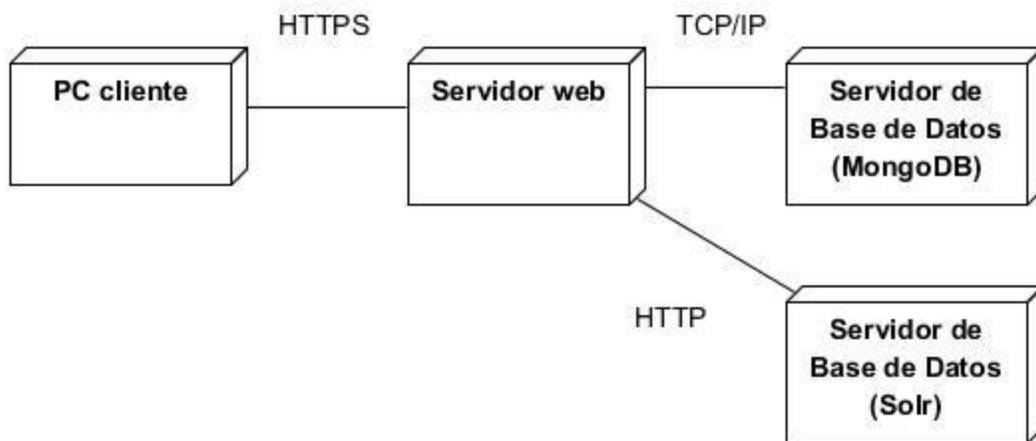


Figura 12: Diagrama de despliegue

A continuación se explica cada nodo del diagrama de despliegue:

**PC cliente:** representa el conjunto de computadoras a través de las cuales los usuarios pueden consultar la información que se encuentra en el Servidor web. La comunicación entre las PC cliente y el Servidor web se establece utilizando el protocolo de comunicación HTTPS.

**Servidor de Aplicación:** es el nodo que realiza las funciones de intermediario entre las PC cliente que realizan sus peticiones y el servidor donde se encuentra la información. Este servidor web toma los datos, realiza sus funciones y presenta la información a las PC cliente mediante el protocolo TCP/IP.

**Servidor de Bases de Datos (MongoDB):** nodo que contiene todos los datos de la gestión de usuarios, sistemas externos y roles.

**Servidor de Bases de Datos (Solr):** nodo que contiene todos los tuits descargados por la aplicación.

### 1.3 Validación del sistema

Una vez terminada la implementación del producto que se requiere es necesario realizarle pruebas, con el objetivo de detectar errores en la aplicación y la documentación; este proceso resulta de gran importancia porque da una medida de la calidad del mismo, siempre que se lleve a cabo de la forma correcta.

Las pruebas son los procesos que permiten verificar y revelar la calidad de un producto de *software*. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un *software*. En este proceso se ejecutan pruebas dirigidas al sistema de *software* en su totalidad, con el objetivo de medir el grado en que el *software* cumple con los requerimientos.

#### 1.3.1 Pruebas funcionales

Las pruebas funcionales, también denominadas pruebas de comportamiento o de caja negra se centran en los requisitos funcionales del *software*. O sea, la prueba de caja negra garantiza al ingeniero del *software* obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Permiten descubrir otra clase de errores tales como: funciones incorrectas o

ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación [70].

La partición equivalente según Roger S. Pressman es una técnica de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Esta se dirige a la definición de casos de prueba que descubran clases de errores, para reducir el número total de casos de prueba que hay que desarrollar. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada [70]. Las pruebas se realizan a partir de una matriz de datos donde 'V' indica válido, 'I' inválido y 'NA' indica que no es necesario proporcionar un valor del dato, ya que es irrelevante. A continuación, se muestra la descripción de las variables que representan valores de entrada de datos para los casos de prueba aplicados a los casos de usos Autenticar usuario y Gestionar usuario.

Tabla 6: Descripción de las variables correspondiente al caso de prueba para el caso de uso Autenticar usuario.

Variables	Nombre del campo	Clasificación	Valor nulo	Descripción
V1	Usuario	Campo de texto	No	Campo que admite caracteres de cualquier tipo. Inicialmente contiene el usuario, puede ser modificado.
V2	Contraseña	Campo de texto	No	Campo que admite caracteres de cualquier tipo. Inicialmente contiene la contraseña del usuario, esta puede ser modificada.

Tabla 7: Descripción del caso de prueba para el caso de uso Autenticar usuario

Sección 1: "Autenticar usuario"						
Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central	
<b>EC 1.1</b> Insertar datos de autenticación correctamente.	Se inserta el usuario y contraseña correctamente.	<b>V</b>	<b>V</b>	Autentica el usuario y le otorga permisos en correspondencia con su rol.	1.	Se introduce usuario y contraseña.
					2.	Se selecciona la opción "Entrar".

<b>EC 1.2</b> Dejar campos vacíos.	Se intenta acceder al sistema dejando campos vacíos.	I	I	No autentica el usuario	1. Se selecciona la opción "Entrar".
<b>EC 1.3</b> El usuario y contraseña no existen.	Se inserta usuario y contraseña.	V	V	No autentica el usuario	1. Se introduce usuario y contraseña. 2. Se selecciona la opción "Entrar".

Tabla 8: Descripción de las variables correspondiente al caso de prueba para el caso de uso "Gestionar usuario"

No	Nombre del campo	Clasificación	Valor nulo	Descripción
V1	Nombre de usuario	Campo de texto	No	Campo que admite caracteres alfanuméricos. Puede ser modificado.
V2	Correo electrónico	Campo de texto	No	Campo que contiene el correo electrónico de un usuario, admite caracteres válidos que cumplen con la estructura de una dirección de correo. Puede ser modificado.
V3	Contraseña	Campo de texto	No	Campo que admite caracteres de cualquier tipo. Inicialmente contiene la contraseña del usuario. Puede ser modificado.
V4	Rol	Campo de selección	No	Campo que contiene una lista de objetos de tipo Rol, puede modificarse.
V5	Activo	Campo de selección	Sí	Campo para seleccionar si desea que el usuario esté activo o no.
V6	Confirmado	Campo de selección	Sí	Campo de fecha que permite insertar la fecha de creación del usuario.

Tabla 9: Caso de prueba para el caso de uso Gestionar usuario: Sección 1: "Crear usuario"

Sección 1: "Crear usuario"									
Escenario	Descripción	V	V	V	V	V	V	Respuesta del sistema	Flujo central
		1	2	3	4	5	6		
<b>EC 1.1</b> Insertar	Se insertan todos los	V	V	V	V	N	N	Inserta un nuevo usuario.	1. Se selecciona la pestaña "Usuario" en

datos del usuario correctamente.	datos del usuario correctamente							Muestra la lista de los usuarios creados y un mensaje indicando que el usuario fue creado satisfactoriamente.	la página de administración. 2. Selecciona la opción "Crear". 3. Se introducen los datos. 4. Se selecciona la opción "Guardar" o "Guardar y Continuar".
<b>EC 1.2</b> Dejar campos vacíos.	Se intenta insertar un usuario dejando campos vacíos.	I	I	I	I	<b>N A</b>	<b>N A</b>	Señala los campos que aún estén vacíos, para que se llenen. Muestra un mensaje indicando que los campos no pueden estar vacíos.	1. Se selecciona la pestaña "Usuario" en la página principal. 2. Selecciona la opción "Crear". 3. Se introducen los datos. 4. Se selecciona la opción "Guardar" o "Guardar y Continuar". 5. Se muestran mensajes de error.
<b>EC 1.3</b> Insertar datos inválidos.	Se intenta insertar un usuario con datos incorrectos.	I	I	I	I	<b>N A</b>	<b>N A</b>	Señala los campos que tengan datos inválidos. Muestra un mensaje indicando que entre los datos correctamente.	1. Se selecciona la pestaña "Desarrollador" en la página principal. 2. Selecciona la opción "Crear". 3. Se introducen los datos. 4. Se selecciona la opción "Guardar" o "Guardar y Continuar". 5. Se muestran mensajes de error.

Los restantes secciones del caso de prueba "Gestionar usuario" y demás casos de pruebas se pueden consultar en las tablas del Anexos 6.

La implementación de las pruebas funcionales de los casos de uso muestra una visión de la calidad del *software*. En una primera iteración, se obtuvo como resultado del análisis de los escenarios de pruebas, un conjunto de 5 no conformidades correspondientes a los servicios implementados. Las no conformidades fueron corregidas posteriormente, dándole paso a la segunda iteración de pruebas, en las cuales se evaluaron nuevamente, obteniendo 4 no conformidades. En una tercera iteración de las pruebas se verificó la corrección de las no conformidades arrojadas en la segunda iteración y se verificó nuevamente los requisitos correspondientes a todos los casos de uso del sistema. Las no conformidades detectadas estuvieron entorno a errores de idioma, ortografía, funcionalidad y validación. En sentido general se validó el correcto funcionamiento de los servicios y el cumplimiento de los requisitos funcionales del *software*.

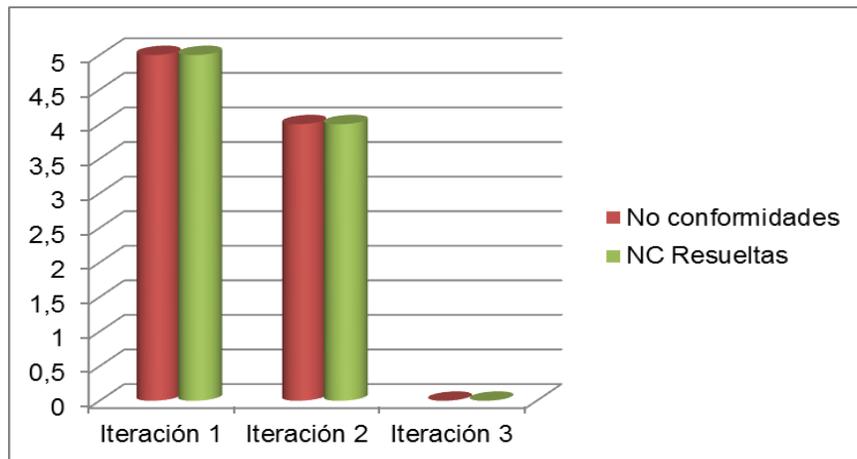


Figura 13: Resultados obtenidos en las pruebas funcionales al subsistema de almacenamiento y recuperación de tuits

### 1.3.2 Pruebas de rendimiento

Se enfoca en la capacidad de recibir peticiones mediante la utilización de una herramienta, verifica cuantas peticiones puede sostener el sistema sin que este se vea afectado, así como la velocidad de respuesta de este. Las pruebas serán realizadas utilizando Jmeter que es una aplicación de escritorio y de código abierto, diseñada para realizar pruebas de carga y medir el rendimiento del sistema. Es una herramienta Java desarrollada dentro del proyecto Jakarta.

Para probar la escalabilidad del sistema es necesario conocer acerca de la carga que puede soportar. Para ello se implementaron las pruebas de rendimiento de *software*, las cuales se apoyaron en el uso de la herramienta Jmeter, que ofrece datos importantes acerca del estrés que puede soportar el sistema. Primeramente se probó el sistema con 50 usuarios accediendo al mismo tiempo a diferentes páginas de la herramienta con un total de 1050 peticiones, arrojando 183,0 peticiones por segundo. En una segunda prueba resultaron 3150 peticiones con 150 usuarios accediendo, arrojando como resultado 22,3 peticiones por segundo.

Label	# Muestras	Rendimiento	Kb/sec
/login	150	26,7/sec	75,31
/	50	36,4/sec	119,51
/static/bootstrap...	450	91,1/sec	20,74
/about	150	44,1/sec	152,68
/catalogo	100	49,4/sec	173,21
/contact	100	64,1/sec	225,05
/logout	50	46,6/sec	150,99
TOTAL	1050	183,0/sec	361,23

Figura 14: Resultados de pruebas con Jmeter con 50 usuarios concurrentes

URL	# Muestras	Rendimiento	Kb/sec
/login	450	3,2/sec	9,0
/	150	1,1/sec	3,7
/static/bootstrap...	1350	9,6/sec	2,2
/about	450	3,2/sec	11,2
/catalogo	300	2,2/sec	7,6
/contact	300	2,2/sec	7,7
/logout	150	1,1/sec	3,6
TOTAL	3150	22,3/sec	44,0

Figura 15: Resultados de pruebas con Jmeter con 150 usuarios concurrentes

#### 1.4 Prototipo de interfaz de usuario

A continuación se muestra las vistas principales de la herramienta desarrollada:

# Autenticación

Recordar

Si olvidó su contraseña o desea tener una cuenta en el sistema, por favor contacte al administrador haciendo [clic aquí](#)

Figura 16: Prototipo de interfaz. Autenticar usuario

```
(envDownTuit)karlo@karlo-PC:/var/htdocs/sart/descargar tuit$ python descargar tuit.py
Invierten como Doctores Honoris Causa a tres personalidades #santiagueras http://t.co/aIbcTt3A
RT @PiezaDelCuboEcu: Mis seguidores son de España (75%) Ecuador (16%) Mexico (9%).Averigua los tuyos en: http://t.co/Xu8w7LPgeR
@dany1604Aguilar pues ganate el md con tini y damelo tenes hasta las 8 pm (Mexico)
RT @edbuscaglia: Creciente violencia política en #Mexico fué predicha x años y seguirá creciendo hasta que"maxi-procesen"a"autoridades"mafi...
Todos los usuarios de Windows son potenciales víctimas del ciberespionaje #TICHabana2015 #Cuba http://t.co/U64MDu3ehW
@_Michana holiwis! Les mando un saludo y un besote de mexico
Armada Of UFOs Caught In Mexico [Video] http://t.co/SASP8Jmksa
RT @newslinereport: Al reducir precios, la #TVpaga de #México gana un millón de clientes. #TVpaga #NewsLineReport
http://t.co/cqMPS4lQB7 ht...
Dos horas de viaje por el infierno de la Línea 3 del #Metro http://t.co/6oyifyLa0k #CDMX #DF #PRD_DF #MEXICO #mm http://t.co/Kmzdcnx00
@Legokid3000 PRI y PAN compran y acarrean votos, amenazan y manipulan personas, son un mal para mexico
Sesiona en Cuba Festival de la Cultura China - Cuba.cu http://t.co/SUftDCoEU5 #VitalEnergy
#Elecciones2015 -Pasan los días y la Historia se Repite...DERROCHE, REPRESION #Puebla #MEXICO https://t.co/R0LxBuy80D -
@MeriPrieto Es tiempo de que tengas en tu cabeza el valor de vencer al sistema bipartidista. Cambiemos Mexico
En #Puebla #NO les des mas Poder ! https://t.co/b051s2FBZ3 - #MEXICO #Elecciones2015 #CDMX
#PAN Gobiernos sin respuestas Así son ellos, A Poco No? #Puebla #MEXICO #Elecciones2015 https://t.co/hSSte7atrZ
En #Puebla #NO les des mas Poder ! #MEXICO #Elecciones2015 http://t.co/z04HXMXgW3
Por qué tiene #Cuba tan mal índice de conexión a internet
http://t.co/J1hxIpXRIx http://t.co/4xmjbILYSL
#PAN Gobiernos sin respuestas Así son ellos, A Poco No? #Puebla #MEXICO #Elecciones2015 http://t.co/aTsfXTHXai
En #Puebla #NO les des mas Poder ! #MEXICO #Elecciones2015 http://t.co/Z6XiaySwb8
Moreno Valle espía llamadas de políticos opositores, revela dossier http://t.co/uVIbxNJRu2 ... #Puebla #MEXICO http://t.co/j6FEaxogxG
En #Puebla #NO les des mas Poder ! #MEXICO #Elecciones2015 http://t.co/KKjilFIIm75
@iigza Es tiempo de que tengas en tu cabeza el valor de vencer al sistema bipartidista. Cambiemos Mexico
```

Figura 17: Vista de la herramienta descargando tuits

```

[{"Palabra clave": "samsung", "Tuits": [{"_version_": "1501798557976363008", "id": "#Maicol Samsung preinstalar\u00e9 en los Galaxy S6 y S6 Edge el asistente virtual Sherpa http://t.co/e65zIPgXUB #FLoww", "tuit": "#Maicol Samsung preinstalar\u00e9 en los Galaxy S6 y S6 Edge el asistente virtual Sherpa http://t.co/e65zIPgXUB #FLoww"}, {"_version_": "1501703614972821504", "id": "#QuierounGalaxyS6 porque me encanta Samsung mi primer celular fue uno y yo lo quiero", "tuit": "#QuierounGalaxyS6 porque me encanta Samsung mi primer celular fue uno y yo lo quiero"}, {"_version_": "1501701582288322560", "id": "@ZIPCOMPUTER: #OFERTA D\u00e9 de las Madres - Samsung Galaxy Fame Lite 0424-3572610 Movistar Llama Ya https://t.co/ktGFpTYK3H", "tuit": "@ZIPCOMPUTER: #OFERTA D\u00e9 de las Madres - Samsung Galaxy Fame Lite 0424-3572610 Movistar Llama Ya https://t.co/ktGFpTYK3H"}, {"_version_": "150165660442886144", "id": "122", "tuit": "RT @ErnestoVeraRod: #Cuba Cada d\u00eda es una nueva oportunidad para luchar por la libertad de mi gente Enviado de Samsung Mobile"}, {"_version_": "1501656621105807360", "id": "159", "tuit": "RT @ErnestoVeraRod: #Cuba gracias a los amigos que recargaron mi movil 53162276. Ahora tengo para twitear Enviado de Samsung Mobile"}, {"_version_": "1501656884325646336", "id": "732", "tuit": "RT @ErnestoVeraRod: #Cuba Odio la monoton\u00eda de los domingos Enviado de Samsung Mobile"}, {"_version_": "1501656888532533248", "id": "742", "tuit": "RT @ErnestoVeraRod: #Cuba Fui a Internet a ver si hab\u00eda ocurrido algo nuevo pero nada: Lo mismo del domingo pasado. Enviado de Samsung Mobi\u00e9"}, {"_version_": "1501703446847291392", "id": "Cada que veo un comercial de Samsung #QuieroUnGalaxyS6", "tuit": "Cada que veo un comercial de Samsung #QuieroUnGalaxyS6"}, {"_version_": "1501096023008739328", "id": "Celulares bajo pedido #Samsung #Note4 $919.99 Info: 0989322252 #SigueNuestrasPromociones http://t.co/SteMknIX4u", "tuit": "Celulares bajo pedido #Samsung #Note4 $919.99 Info: 0989322252 #SigueNuestrasPromociones http://t.co/SteMknIX4u"}, {"_version_": "1501820910540161024", "id": "Conozca las caracter\u00edsticas del nuevo Samsung Galaxy A8 http://t.co/d0TwG5RBss", "tuit": "Conozca las caracter\u00edsticas del nuevo Samsung Galaxy A8 http://t.co/d0TwG5RBss"}, {"_version_": "1501820904295890944", "id": "En #Tecn\u00f3 tenemos para ti una gran variedad de forros y cargadores para tu #smartphone de las marcas: iPhone, Samsung, Blackberry, etc", "tuit": "En #Tecn\u00f3 tenemos para ti una gran variedad de forros y cargadores para tu #smartphone de las marcas: iPhone, Samsung, Blackberry, etc"}, {"_version_": "1501820905659039744", "id": "La velocidad entre el #Samsung S5 y el iPhone 5S es casi abismal. A favor de #iPhone claro.", "tuit": "La velocidad entre el #Samsung S5 y el iPhone 5S es casi abismal. A favor de #iPhone claro."}, {"_version_": "1501538902053224448", "id": "Mira cu\u00e1nto cuesta el nuevo Samsung Galaxy S6 - http://t.co/RkWzEUrxtq\u00e9 http://t.co/OZRKFNDZRzH", "tuit": "Mira cu\u00e1nto cuesta el nuevo Samsung Galaxy S6 - http://t.co/RkWzEUrxtq\u00e9 http://t.co/OZRKFNDZRzH"}, {"_version_": "1501538910839242752", "id": "Participo en el #Chapuzas4Aniversario para ganar el SSD Samsung 850 EVO que regala @SeismicEs y @elchapuzas http://t.co/PIW4hXwzHu", "tuit": "Participo en el #Chapuzas4Aniversario para ganar el SSD Samsung 850 EVO que regala @SeismicEs y @elchapuzas http://t.co/PIW4hXwzHu"}, {"_version_": "1501095945593421824", "id": "RT @LuisGyG: Arranca la prueba del Samsung Galaxy S6 edge. \u00bfQu\u00e9 les interesa que pruebe de este aparato? http://t.co/gVRDIRfOMc", "tuit": "RT @LuisGyG: Arranca la prueba del Samsung Galaxy S6 edge. \u00bfQu\u00e9 les interesa que pruebe de este aparato? http://t.co/gVRDIRfOMc"}, {"_version_": "1501486092274106368", "id": "RT @trinicperez: @MiltonCentraal Samsung es peor. Sab\u00e9s que hace cuarenta minutos lo prendo y se me apaga. LO PRENDO Y SE ME APAGA ENTEND\u00c9\u00e9", "tuit": "RT @trinicperez: @MiltonCentraal Samsung es peor. Sab\u00e9s que hace cuarenta minutos lo prendo y se me apaga. LO PRENDO Y SE ME APAGA ENTEND\u00c9\u00e9"}, {"_version_": "1501820892212101120", "id": "SAMSUNG GALAXY NOTE 3\n\nPOR SOLO 114\u00d2ac\u00e9\n\nENVIO CONTRAREEMBOLSO\n\nPEDIDOS WHATSAPP 644978349 http://t.co/ZDHSy1HOvd", "tuit": "SAMSUNG GALAXY NOTE 3\n\nPOR SOLO 114\u00d2ac\u00e9\n\nENVIO CONTRAREEMBOLSO\n\nPEDIDOS WHATSAPP 644978349 http://t.co/ZDHSy1HOvd"}, {"_version_": "1501798059038736384", "id": "Samsung preinstalar\u00e9 en los Galaxy S6 y S6 Edge el asistente virtual Sherpa http://t.co/2fnnxIKvWx", "tuit": "Samsung preinstalar\u00e9 en los Galaxy S6 y S6 Edge el asistente virtual Sherpa http://t.co/2fnnxIKvWx"}]}

```

Figura 18: Vista de la herramienta mostrando los tuits solicitados por un cliente

Las restantes vistas de la herramienta est\u00e1n en las figuras correspondientes al Anexo 7.

## 1.5 Conclusiones parciales

La estructuraci\u00f3n del modelo de implementaci\u00f3n en t\u00e9rminos de subsistemas, permite lograr una mayor comprensi\u00f3n sobre c\u00f3mo est\u00e1 integrada la aplicaci\u00f3n y sus dependencias. La definici\u00f3n y utilizaci\u00f3n del est\u00e1ndar de codificaci\u00f3n a emplear durante la implementaci\u00f3n, permiti\u00f3 el desarrollo de los componentes con un alto grado de legibilidad; lo que provee una gu\u00eda para el mantenimiento y actualizaci\u00f3n del sistema, con c\u00f3digo claro y bien documentado. A trav\u00e9s de la validaci\u00f3n del sistema, aplicando pruebas funcionales y de rendimiento; facilit\u00f3 identificar y solucionar las deficiencias detectadas en el sistema. La aplicaci\u00f3n de estas pruebas permiti\u00f3 monitorear el funcionamiento de la aplicaci\u00f3n en diferentes entornos de ejecuci\u00f3n y carga.

## **Conclusiones**

Con la realización de este trabajo se cierra un período de desafíos y gratitudes asociadas a la superación de un número importante de insatisfacciones en el uso de la red social Twitter, las que dificultaban el trabajo del Centro de Ideoinformática y las necesidades de información. El estudio realizado a los sistemas de almacenamiento y recuperación de tuits desarrollados a nivel internacional, permitió caracterizarlos y comprobar la no existencia de uno que cumpliera las expectativas asociadas a las necesidades de uso de Twitter en el Centro de Ideoinformática, tomando referencia a buenas prácticas y el funcionamiento asociado al consumo de tuits para el desarrollo de la propuesta.

Se desarrolló un subsistema de almacenamiento y recuperación de tuits para el trabajo con la red social Twitter lo que permite economizar el uso de ancho de banda del acceso a Internet y aumentar la disponibilidad de los tuits. Este garantiza la descarga de tuits desde Twitter a partir de consultas parametrizadas, almacena la información de forma local y brinda servicios a terceras aplicaciones para el consumo de la información registrada. Se sometió el sistema propuesto a varias pruebas concebidas como parte del proceso de desarrollo, validándose en cada caso el funcionamiento del mismo y la respuesta ante pruebas de carga y estrés.

## **Recomendaciones**

Se recomienda extender el uso del sistema desarrollado a otras instituciones en las que se utilicen diferentes aplicaciones para el análisis de la red social Twitter.

## Referencias bibliográficas

1. PLA-SANTAMARIA, David. *Localización de información específica en la web* [En línea]. Universidad Politécnica de Valencia, 2005. [Consulta: 29 marzo 2015]. ISBN 9788497057400. Disponible en: <http://books.google.es/>.
2. GARCÍA, Leannys de la Caridad Labrada y PÉREZ, Lisandro Calvis. La ética informática en la red social Facebook. [En línea]. 9 marzo 2012. [Consulta: 16 octubre 2014]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/3946](http://repositorio_institucional.uci.cu/jspui/handle/ident/3946)
3. CARBALLIDO, Juan Ramón Sánchez. Perspectivas de la información en Internet: ciberdemocracia, redes sociales y web semántica. *Zer - Revista de Estudios de Comunicación*. 2008. Vol. 13, no. 25, p. 61-81.
4. ORIHUELA-COLLIVA, J. L. (José Luis). Internet: la hora de las redes sociales. *Nueva revista de política, cultura y arte*. 14 octubre 2008. No. 119, p. 57-65.
5. STONE, Biz. There's a List for That. *Twitter Blogs* [En línea]. 2009. [Consulta: 6 mayo 2015]. Disponible en: <https://blog.twitter.com/2009/theres-list>
6. BARBA, Claudia Fabiola Ortega y CAMPERO, Alejandra Banderas. Percepción de los jóvenes universitarios sobre el uso de Twitter en los procesos de enseñanza-aprendizaje. *Revista Apertura* [En línea]. 29 junio 2012. Vol. 3, no. 2. [Consulta: 26 mayo 2015]. Disponible en: <http://www.udgvirtual.udg.mx/apertura/index.php/apertura3/article/view/203>
7. MICHEL LÓPEZ CAMINO y VIDAL, Raydel Muñoz. *Gestor de Recuperaciones Dinámicas para aplicaciones web*. [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Informática. La Habana: Instituto Superior Politécnico José Antonio Echeverría, 2006. [Consulta: 16 octubre 2014]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/TD\\_0174\\_06](http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_0174_06)
8. VALLE, Amaury E. del. Se amplía el acceso público a Internet en Cuba. *Cubadebate* [En línea]. 28 mayo 2013. [Consulta: 10 junio 2015]. Disponible en: <http://www.cubadebate.cu/noticias/2013/05/28/se-amplia-el-acceso-publico-a-internet-en-cuba/>
9. ¡Una buena noticia!: el cable submarino entre Venezuela y Cuba. *Cubadebate* [En línea]. [Consulta: 9 junio 2015]. Disponible en: <http://www.cubadebate.cu/noticias/2011/01/12/%c2%a1una-buena-noticia-el-cable-submarino-entre-venezuela-y-cuba/>
10. Misión | Portal de la Universidad de las Ciencias Informáticas. [En línea]. [Consulta: 27 mayo 2015]. Disponible en: <http://www.uci.cu/?q=mision>

11. SAMPIERI, Roberto Hernández. *Metodología de la investigación* [En línea]. 3. México DF : Mc Graw Hill Interamericana, 1998. ISBN 970-10-5753-7. Disponible en: [http://sapp.uv.mx/univirtual/especialidadesmedicas/mi2/modulo1/docs/Met\\_Invest\\_a.pdf](http://sapp.uv.mx/univirtual/especialidadesmedicas/mi2/modulo1/docs/Met_Invest_a.pdf)
12. ZAMORA, Marcelo. *Redes sociales en Internet. Maestros del web*. 2006.
13. MANETTO, Francesco. La cara oculta de Facebook. [En línea]. 2008. [Consulta: 16 octubre 2014]. Disponible en: [http://elpais.com/diario/2008/01/27/sociedad/1201388405\\_850215.html](http://elpais.com/diario/2008/01/27/sociedad/1201388405_850215.html)
14. SUEIRAS, Edita. Las redes sociales o cómo conectarse con el mundo. [En línea]. 2010. Disponible en: [http://www.lasticenelaula.es/escuela20/index.php?option=com\\_content&view=article&id=42:las-redes-sociales-o-como-conectarse-con-el-mundo&catid=30:redes-sociales&Itemid=40](http://www.lasticenelaula.es/escuela20/index.php?option=com_content&view=article&id=42:las-redes-sociales-o-como-conectarse-con-el-mundo&catid=30:redes-sociales&Itemid=40).
15. SEDANO, Gloria. Beneficios de las redes sociales. *Webspacio* [En línea]. 11 noviembre 2014. [Consulta: 11 noviembre 2014]. Disponible en: <http://www.webspacio.com/beneficios-redes-sociales/>
16. PUON, Liz. Las 6 mejores redes sociales de acuerdo con el número de usuarios activos. *Revista Merca2.0* [En línea]. 23 octubre 2014. [Consulta: 23 octubre 2014]. Disponible en: <http://www.merca20.com/las-6-mejores-redes-sociales-de-acuerdo-con-el-numero-de-usuarios-activos/>
17. About Twitter, Inc. | Investors. *Twitter About* [En línea]. 23 octubre 2014. [Consulta: 23 octubre 2014]. Disponible en: <https://investor.twitterinc.com/>
18. Twitter. [En línea]. Disponible en: [twitter.com](http://twitter.com)
19. EL MORAL, J.A. *Redes Sociales ¿Moda o nuevo paradigma? Madrid: Asociación de usuarios de Internet*. 2005.
20. PÉREZ, Reynier Martínez. *Subsistema de almacenamiento y gestión de archivos para el producto Captura y Catalogación de Medias* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2010. [Consulta: 23 octubre 2014]. Disponible en: [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/TD\\_03676\\_10](http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_03676_10)
21. SHARP, Remy. Snap Bird - search twitter's history. [En línea]. [Consulta: 6 mayo 2015]. Disponible en: <http://snapbird.org/>
22. RODRÍGUEZ, Iván. TweetBackup - Archiva diariamente tus tweets. [En línea]. 13 noviembre 2014. [Consulta: 13 noviembre 2014]. Disponible en: <http://www.twittboy.com/2009/12/tweetbackup-archiva-diariamente-tus.html>
23. TwimeMachine ~ Buscar tus tweets más antiguos. *TwimeMachine ~ Buscar tus tweets más antiguos. ~ Curioseando* [En línea]. 15 mayo 2010. [Consulta: 13 noviembre 2014]. Disponible en: <http://curioseandito.blogspot.com/2010/05/twimemachine-buscar-tus-tweets-mas.html>

24. REST APIs. *Twitter Developers* [En línea]. [Consulta: 6 mayo 2015]. Disponible en: <https://dev.twitter.com/rest/public>
25. SOMMERVILLE, Ian. *Ingeniería del software* [En línea]. 7ma. Pearson Educación, 2005. ISBN 9788478290741. Disponible en: <https://books.google.es/>.
26. PÉREZ, Isaías Carrillo, GONZÁLEZ, Rodrigo Pérez y MARTÍN, Aureliano David Rodríguez. Metodologías de desarrollo. *Scribd* [En línea]. [Consulta: 2 junio 2015]. Disponible en: <https://es.scribd.com/doc/54060274/Metodologias-de-desarrollo>
27. PRESSMAN, Roger S. *Ingeniería del software: un enfoque práctico*. 5ta. McGraw-Hill, 2003. ISBN 9788448132149.
28. IBM - Rational Unified Process. [En línea]. 2011. [Consulta: 11 noviembre 2014]. Disponible en: <http://www-01.ibm.com/software/awdtools/rup/>.
29. CARVAJAL, Juan José Hernández. *Desarrollo del Módulo Administrador de Reportes del Generador Dinámico de Reportes* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2011. [Consulta: 16 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/TD\\_04378\\_11](http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04378_11)
30. Proceso de entrega: OpenUP/Basic Lifecycle. [En línea]. 2 noviembre 2014. [Consulta: 2 noviembre 2014]. Disponible en: [http://epf.eclipse.org/wikis/openupsp/openup\\_basic/deliveryprocesses/openup\\_basic\\_process\\_0uyGoMlgEdmt3adZL5Dmdw\\_desc.html](http://epf.eclipse.org/wikis/openupsp/openup_basic/deliveryprocesses/openup_basic_process_0uyGoMlgEdmt3adZL5Dmdw_desc.html)
31. Open UP - Blog de OpenUP. [En línea]. [Consulta: 16 abril 2015]. Disponible en: <http://openup.blogspot.es/tags/open-up/>
32. LARMAN, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2. Pearson Educación, 2003. ISBN 9788420534381.
33. WILLIAM S. DAVIS. *Herramientas CASE: metodología estructurada para el desarrollo de los sistemas* [En línea]. Thomson-Paraninfo, 1992. [Consulta: 26 mayo 2015]. ISBN 8428319278. Disponible en: <http://dialnet.unirioja.es/servlet/libro?codigo=227309>
34. Visual Paradigm para UML. [En línea]. 11 noviembre 2014. [Consulta: 11 noviembre 2014]. Disponible en: <http://www.software.com.ar/visual-paradigm-para-uml.html>
35. PROENZA, Manuel A. Novoa y GÓMEZ, Yasmani J. Álvarez. *Capa de servicio web para el motor de búsqueda Orión* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2012. [Consulta: 16 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/TD\\_05126\\_12](http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05126_12)

36. LOBOS, María Elena De los. *Aprende a programar* [En línea]. 2005. Disponible en: <http://www.mailxmail.com/curso-aprende-programar/tipos-estructuras-programacion-estructuras-basicas-secuencial>.
37. Lenguajes de programación, programación Java. [En línea]. 11 noviembre 2014. [Consulta: 11 noviembre 2014]. Disponible en: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>
38. Oracle Technology Network for Java Developers | Oracle Technology Network | Oracle. [En línea]. [Consulta: 10 mayo 2015]. Disponible en: <http://www.oracle.com/technetwork/java/index.html>
39. Lenguaje de programación Python. [En línea]. 13 noviembre 2014. [Consulta: 13 noviembre 2014]. Disponible en: <http://programandoideas.com/el-lenguaje-de-programacion-python/>
40. JOSUTTIS, Nicolai M. *The C++ Standard Library: A Tutorial and Reference*. Addison-Wesley, 2012. ISBN 9780132977739.
41. Twython 1.2: Python Package Index. [En línea]. 28 noviembre 2014. [Consulta: 28 noviembre 2014]. Disponible en: <https://pypi.python.org/pypi/twython/1.2>
42. CRESPO, Ernesto. Probando la librería twython para Twitter. *Blog de Ernesto Crespo* [En línea]. 12 noviembre 2014. [Consulta: 12 noviembre 2014]. Disponible en: <http://ernestocrespo13.wordpress.com/2013/09/22/probando-la-libreria-twython-para-twitter/>
43. IDE (Entornos de Desarrollo Integrado). *Vigo: Unión Webmaster Libres* [En línea]. 28 noviembre 2014. [Consulta: 28 noviembre 2014]. Disponible en: <https://sites.google.com/site/vigomiciudad/otras-cosa-inter-santes/ide-entornos-de-desarrollo-integrado>
44. Python IDE & Django IDE for Web developers : JetBrains PyCharm. [En línea]. 28 noviembre 2014. [Consulta: 28 noviembre 2014]. Disponible en: <https://www.jetbrains.com/pycharm/>
45. PyCharm el IDE de Python lanza su versión 3.4. *Hipertextual* [En línea]. [Consulta: 16 abril 2015]. Disponible en: <http://hipertextual.com/archivo/2014/06/pycharm-ide-python/>
46. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). [En línea]. [Consulta: 17 abril 2015]. Disponible en: <http://www.w3.org/TR/soap12-part1/>
47. RIEHLE, Dirk. *Framework design* [En línea]. Diss. Technische Wissenschaften ETH Zürich, Nr. 13509, 2000, 2000. Disponible en: <http://dirkriehle.com/computer-science/research/dissertation/diss-a4.pdf>
48. CHRISTIE, Tom. Django REST Framework. [En línea]. 2014. Disponible en: <http://www.django-rest-framework.org/>.

49. GRINBERG, Miguel. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 2014. ISBN 9781491947616.
50. JSON. [En línea]. 2010. [Consulta: 11 noviembre 2014]. Disponible en: <http://www.json.org/json-es.html>
51. PADRÓN, Ailen Moreira y CABRERA, Yoan Abreu. *Librerías de la programación en el cliente (JavaScript). Incorporación de estas tecnologías al Sistema de Administración de Contenidos Drupal*. [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2007. [Consulta: 16 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/TD\\_0908\\_07](http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_0908_07)
52. RAMAKRISHNAN, Raghu y GEHRKE, Johannes. *Database management systems*. New York: McGraw-Hill : McGraw-Hill New York, 2003.
53. MATO, Rosa M. García. Diseño de Bases de Datos. *Scribd* [En línea]. 1991. [Consulta: 11 noviembre 2014]. Disponible en: <https://es.scribd.com/doc/222220519/Rosa-Maria-Diseno-de-Bases-de-Datos>
54. *MongoDB* [En línea]. 2011. Disponible en: <http://www.mongodb.org/>
55. CHODOROW, Kristina y DIROLF, Michael. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage--New from O'Reilly*. [En línea]. 2010. [Consulta: 2 noviembre 2014]. Disponible en: <http://www.oreilly.com/pub/pr/2646>
56. The MongoDB 2.6 Manual — MongoDB Manual 2.6.4. <https://github.com/mongodb/docs/blob/master/source/index.txt> [En línea]. 28 noviembre 2014. [Consulta: 28 noviembre 2014]. Disponible en: <http://docs.mongodb.org/manual/>
57. SÁNCHEZ, Roberlán Rodríguez y DOMÍNGUEZ, Joe del Toro. *Seminario sobre Ficheros de Índice, Árboles B, B+, B\**. Ciudad Habana, 2009.
58. Apache Solr -. [En línea]. 1 diciembre 2014. [Consulta: 1 diciembre 2014]. Disponible en: <http://lucene.apache.org/solr/>
59. SUÁREZ, José Manuel Sánchez. Introducción a Apache Solr. [En línea]. 28 abril 2011. [Consulta: 21 mayo 2015]. Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=apacheSolrIntro>
60. GARCÍA, Anay Medina y CHONG, Meylin Martínez. *Propuesta arquitectónica de un Sistema de Recuperación de Información Geográfica para el motor de búsqueda Orión* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2011. [Consulta: 16 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/TD\\_04252\\_11](http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_04252_11)

61. JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. *El proceso unificado de desarrollo de software*. Addison Wesley Reading, 2000.
62. ¿Cuál es la definición de Requerimientos (desarrollo de sistemas/software)? [En línea]. [Consulta: 11 marzo 2015]. Disponible en: <http://www.alegsa.com.ar/Dic/requerimientos.php>
63. PRESSMAN, Roger S. *Ingeniería de Software. Un enfoque práctico*. 5ta. Mc Graw Hill, 2001.
64. GAMMA, Erich y HELM, Richard. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994. ISBN 9780321700698.
65. SÁNCHEZ, Rammel Maestre. *Arquitectura del Módulo de Administración y Control de Recursos del Sistema de Gestión de Emergencia y Seguridad Ciudadana 171* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2009. [Consulta: 17 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/TD\\_2674\\_09](http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_2674_09)
66. FOWLER, Martin. *Patterns of Enterprise Application Architecture* [En línea]. Addison-Wesley Professional, 2003. [Consulta: 5 junio 2015]. ISBN 9780321127426. Disponible en: <https://books.google.com.cu/books?id=vqTfNFDzzdIC&printsec=frontcover&dq=inauthor:%22Martin+Fowler%22&hl=es&sa=X&ved=0CE4Q6AEwBmoVChMlm6K2stSExglVyC2MCh0kCwDp#v=onepage&q&f=false>
67. MELGAREJO, Yadira Estrada y MARTÍNEZ, Yasiel Jorge. *Servicios Web para la creación de aplicaciones SEO* [En línea]. Trabajo de Diploma para optar por el título en Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2009. [Consulta: 17 abril 2015]. Disponible en: [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/TD\\_2699\\_09](http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_2699_09)
68. SPARX SYSTEMS PTY LTD. Sparx Systems - UML 2 Tutorial - Component Diagram. *Sparx System* [En línea]. 2015 2000. [Consulta: 10 junio 2015]. Disponible en: [http://www.sparxsystems.com/resources/uml2\\_tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com/resources/uml2_tutorial/uml2_componentdiagram.html)
69. Revisiones de código y estándares de codificación. [En línea]. 19 abril 2015. [Consulta: 19 abril 2015]. Disponible en: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx)
70. PRESSMAN, Roger S. *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005. ISBN 9780073019338.

## Glosario de términos

**API:** interfaz de programación de aplicaciones (por sus siglas en inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

**Banner:** están diseñados con la intención de llamar la atención, resaltar notorios y comunicar el mensaje deseado. Se crean con imágenes (GIF, JPEG o PNG), o con animaciones creadas a partir de tecnologías como Java, Adobe Shockwave y, fundamentalmente, Flash.

**BSD:** es la licencia de *software* otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de *software* libre permisiva.

**Consultas parametrizadas:** cuando se realiza una petición esta lleva algún parámetro el cual especifica los criterios de la búsqueda. Un parámetro de consulta es un elemento que contiene un valor que se puede cambiar para influir en los resultados obtenidos.

**GNU/GPL:** (Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License*) es la licencia más ampliamente usada en el mundo del *software* y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el *software*.

**HTML:** siglas de *HyperText Markup Language* (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web.

**HTTP:** *Hypertext Transfer Protocol* o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la *World Wide Web*.

**OAuth:** por sus siglas en inglés *Open Authorization* es un protocolo abierto, que permite autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, móviles y web.

**PDF:** siglas del inglés *Portable Document Format*, formato de documento portátil es un formato de almacenamiento de documentos digitales independiente de plataformas de *software* o *hardware*.

**TCP/IP:** el nombre TCP / IP Proviene de dos protocolos importantes, el Protocolo de Control de Transmisión (por sus siglas en inglés *Transmission Control Protocol*) y el Protocolo de Internet (por sus siglas en inglés *Internet Protocol*).

**Stakeholders:** persona o entidad que está interesada en la realización de un proyecto o tarea, auspiciando el mismo ya sea mediante su poder de decisión y/o de financiamiento o a través de su propio esfuerzo. Son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa.

**UML:** es un lenguaje de modelado visual de propósito general, que es empleado para especificar, visualizar, construir y documentar los artefactos de un sistema de *software*.

**XML:** por sus siglas en inglés de *eXtensible Markup Language* en español lenguaje de marcas extensible, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.