

**Universidad de las Ciencias Informáticas  
Facultad 1**



**Título: “Sistema de análisis estadístico de  
Twitter”.**

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

**Autores:** Yoandry Lázaro Romay Herrera  
Yisel Verdecia Jorge

**Tutores:** Ing. José Gabriel Espinosa Ramírez  
Ing. Odelkis Rodríguez Irsula

La Habana, junio 2015

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado **Sistema de análisis estadístico de Twitter**, y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los\_\_ días del mes de Junio del año 2015

---

Yoandry Lázaro Romay Herrera  
Autor

---

Yisel Verdecia Jorge  
Autora

---

Ing. José Gabriel Espinosa Ramírez  
Tutor

---

Ing. Odelkis Rodriguez Irsula  
Tutora

## Agradecimientos

*Primeramente a mi abuelita Fátima (mima como solía decirle) por siempre creer que yo podía llegar a donde quisiera, por sentirse orgullosa de mí en cualquier parte que estuviese, porque sé que le hubiese encantado estar en este momento conmigo, pero Dios no lo quiso así, en donde estés ahora mima te doy las gracias por ser de la forma en que lo fuiste conmigo y sé que cuando esté de pie frente al tribunal exponiendo, tú vas a estar ahí en unas de las sillas mirando a tu niño graduarse de Ingeniero.*

*A mi mamá, mi papá, a mi abuela Luisita, a mi abuelo Erasmo, a mi abuelo Omar y a mis tías por todo lo que se han sacrificado para que yo esté aquí, por todo el amor que me han dado siempre, por estar ahí en todo momento para mí, por tanto que me han tenido que soportar, por todo y por lo que aún resta, porque espero poder seguir molestándolos por mucho tiempo más.*

*A Isvany por apoyarme en este último año, por estar siempre aconsejándome y ayudarme a ser un poco mejor, a Dari por tanto que me ha ayudado, por estar conmigo en mis peores momentos, por cuidarme y brindarme un consejo siempre que lo necesitara, a mis amigos Roger, Jorge, Arisney y Suyín por soportarme estos 5 años, por darme su mano cada vez que me hizo falta y por ser excelentes amigos. A mis compañeros de clase y a mis profesores por las veces que me dieron su ayuda. A mis compañeros de apartamento, a los que ya no están en la universidad y a los que aún se encuentran por ser excelentes compañeros de juego y por enseñarme lo que significa ver un buen filme.*

*A mi compañera de tesis porque sin ella creo que este momento hubiese sido casi imposible, por ser tan excelente persona, por los estresantes momentos que pasamos y por aguantar que la molestara tanto.*

*A mis tutores por su dedicación, por su ayuda y por sus consejos.*

***De Yoandry***

*A mis padres por estar siempre a mi lado a pesar de la distancia, por su apoyo incondicional, por todo el amor y confianza que me han brindado siempre. A toda mi familia en especial a mis hermanos. A mi novio Luis Manuel por soportarme en estos años, por estar siempre a mi lado, incluso en los momentos más difíciles, por su apoyo incondicional y todo su cariño. A mis amigas las chicas superpoderosas (Yanisleidis y Sandra) por compartir esos momentos alegres y tristes conmigo. A todas mis amistades del apartamento 143 308 (incluyendo los varones) por compartir tantos momentos. A mi compañero de aula gracias por su ayuda y apoyo. A los tutores por estar siempre pendiente de nuestro trabajo y brindarnos su apoyo. A mi compañero de tesis por su ayuda, colaboración incondicional y apoyo en todo momento. A todos lo que de una forma u otra han hecho posible este logro en mi vida, muchas gracias.*

***De Yisel***

## **Dedicatoria**

*A mi abuela Fátima, por ser mi guía, por creer en mí y alentarme siempre a que me superara, a que diera lo mejor de mí persona.*

*A mi mamá por todos los sacrificios que ha tenido que hacer para que yo llegara hasta aquí, por no darse por vencida conmigo.*

*A mi papá por todo lo que ha batallado para que yo estuviese lo más cómodo posible, por los sacrificios que ha hecho y los malos momentos que ha pasado por mí.*

*A mi abuela Luisita, a mi abuelo Omar y a mi abuelo Erasmo por ser los mejores abuelos del mundo y por apoyarme en todo momento sin importar cuan malo o buen fuese.*

***De Yoandry***

*A mis padres, a mi hermanita Yadira va dedicado por apoyarme no solo durante mi carrera universitaria sino durante estos años de mi vida, a los cuales le agradezco eternamente. Gracias mamá y papí los amos. A mi novio por su apoyo incondicional en todo momento.*

***De Yisel***

## Resumen

La gestión de la información es la coordinación eficiente y eficaz de la información procedente de fuentes internas y externas. Un ejemplo de estas fuentes son las redes sociales, las cuales en la actualidad han cobrado un papel muy importante en la creación y distribución de la información entre las personas. Entre las redes sociales se encuentra Twitter, para la cual se ha desarrollado un conjunto de aplicaciones online que permite llevar toda esa información a estadísticas mediante tablas y gráficos. A partir del análisis realizado a las aplicaciones existentes, se precisa proveer una solución para el estudio de la red social Twitter que se realiza en el Departamento Operaciones Web y Análisis de Información (DOWAI) del Centro de Ideoinformática (CIDI). En el presente trabajo se describe el proceso de elaboración del Sistema de análisis estadísticos de Twitter que permite generar estadísticas por medio de tablas y gráficos de la información que brinda esta red social. Para su realización se estudian las metodologías de desarrollo, tecnologías y herramientas de software. Se detalla la aplicación de las pruebas para validar el cumplimiento de las funcionalidades identificadas y la optimización del proceso para el cual fue implementada.

**Palabras claves:** Análisis estadísticos, Red social, Twitter.

# Índice de Contenidos

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
FUNDAMENTOS TEÓRICOS.....	6
Definición de redes sociales.....	6
Red Social Twitter .....	6
Estructura de un tuit .....	7
APLICACIONES DE TWITTER.....	10
METODOLOGÍA A UTILIZAR PARA EL DESARROLLO DE LA APLICACIÓN .....	12
Metodologías ágiles .....	13
TECNOLOGÍAS PARA EL DESARROLLO.....	16
Lenguaje de Modelado Unificado.....	16
Herramientas CASE.....	17
Lenguajes de programación .....	18
Lenguajes de maquetado y estilo .....	20
Bibliotecas a utilizar.....	22
Marco de trabajo .....	23
Entorno de Desarrollo Integrado (IDE) .....	25
Sistema Gestor de Base de Datos NoSQL .....	26
Herramienta para Pruebas de Carga y Estrés .....	28
CONCLUSIONES DEL CAPÍTULO.....	29
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN. ....	31
TUITZER COMO PROPUESTA DE SOLUCIÓN.....	31
MODELO DE DOMINIO .....	31
REQUERIMIENTOS FUNCIONALES.....	32
REQUERIMIENTOS NO FUNCIONALES .....	33
MODELO DE CASOS DE USO DEL SISTEMA .....	34
Definición de los actores del sistema.....	34
Diagrama de CU del sistema.....	35

Patrones de CU.....	35
Descripción de los CU .....	36
PATRÓN ARQUITECTÓNICO Y PATRONES DE DISEÑO.....	40
Modelo Plantilla Vista (MTV).....	41
Patrones de diseño.....	41
DIAGRAMAS DE CLASES .....	42
DIAGRAMAS DE INTERACCIÓN .....	43
DISEÑO DE LA BASE DE DATOS .....	44
MODELO DE DESPLIEGUE.....	45
CONCLUSIONES DEL CAPÍTULO.....	46
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS .....	47
IMPLEMENTACIÓN.....	48
Estándares de código.....	48
PRUEBAS.....	51
CONCLUSIONES DEL CAPÍTULO.....	56
CONCLUSIONES GENERALES.....	57
RECOMENDACIONES.....	58
REFERENCIAS BIBLIOGRÁFICAS.....	59
ANEXOS .....	63
ANEXO 1: Descripción de los CU .....	63
ANEXO 2: Diagramas de clase de diseño .....	65
ANEXO 3: Diagramas de interacción del diseño .....	67
ANEXO 4: Casos de pruebas.....	70



# Índice de Tablas

<b>Tabla 1.</b> Estructura de un tuit (7). .....	7
<b>Tabla 2.</b> Comparación de las aplicaciones de gestión estadística de Twitter. ....	11
<b>Tabla 3.</b> Metodologías Tradicionales vs Metodologías Ágiles (8). ....	12
<b>Tabla 4.</b> Justificación de los actores del sistema. ....	34
<b>Tabla 5.</b> Descripción CU Autenticar usuario. ....	36
<b>Tabla 6.</b> Descripción CU Gestionar usuario. ....	37
<b>Tabla 7.</b> Descripción de prueba para el CU Autenticar usuario. ....	52
<b>Tabla 8.</b> Descripción de prueba para el CU Realizar búsquedas simples. ....	53
<b>Tabla 9.</b> Descripción del CU5 Realizar búsquedas simples. ....	63
<b>Tabla 10.</b> Descripción del CU5 Realizar búsquedas avanzadas. ....	64
<b>Tabla 11.</b> Descripción de prueba para el CU Realizar búsquedas avanzadas. ....	70
<b>Tabla 12.</b> Descripción de prueba para el CU Gestionar usuario (Crear usuario). ....	70
<b>Tabla 13.</b> Descripción de prueba para el CU Gestionar usuario (Eliminar usuario). ....	70
<b>Tabla 14.</b> Descripción de prueba para el CU Gestionar usuario (Cambiar permisos). ....	71

## Índice de Imágenes

<b>Imagen 1.</b> Diagrama de clase del modelo de dominio. ....	32
<b>Imagen 2.</b> Diagrama de casos de uso. ....	35
<b>Imagen 3.</b> Arquitectura del Sistema de Análisis Estadístico de Twitter (44). ....	41
<b>Imagen 4.</b> Diagrama de clases de diseño del CU Búsquedas simples. ....	43
<b>Imagen 5.</b> Diagrama de secuencia del CU Búsquedas simples. ....	44
<b>Imagen 6.</b> Modelo Físico de Datos para Mongo DB. ....	45
<b>Imagen 7.</b> Diagrama de despliegue. ....	45
<b>Imagen 8.</b> Diagrama de componentes. ....	47
<b>Imagen 9.</b> Resultado de las pruebas funcionales. ....	54
<b>Imagen 10.</b> Ejecución de los métodos de la clase controladora <i>controladora.py</i> . ....	56
<b>Imagen 11.</b> Resultado de la prueba de los métodos de la clase controladora <i>controladora.py</i> . ....	56
<b>Imagen 12.</b> Diagrama de clase de diseño del CU Autenticar usuario. ....	65
<b>Imagen 13.</b> Diagrama de clase de diseño del CU Gestionar usuario. ....	65
<b>Imagen 14.</b> Diagrama de clase de diseño del CU Realizar búsquedas avanzadas. ....	66
<b>Imagen 15.</b> Diagrama de interacción del diseño del CU Autenticar usuario. ....	67
<b>Imagen 16.</b> Diagrama de interacción del diseño del CU Gestionar usuario: Cambiar permisos de usuarios. ....	67
<b>Imagen 17.</b> Diagrama de interacción del diseño del CU Gestionar usuario: Crear usuario. ....	68
<b>Imagen 18.</b> Diagrama de interacción del diseño del CU Gestionar usuario: Eliminar usuario. ....	69
<b>Imagen 19.</b> Diagrama de interacción del diseño del CU Realizar búsquedas avanzadas. ....	69

## INTRODUCCIÓN

La evolución y el auge de las Tecnologías de la Información y las Comunicaciones (TIC) se ha convertido en un punto fundamental para el desarrollo de los diferentes sectores de la sociedad. La importancia de las TIC ha venido de la mano con la relevancia que ha obtenido la gestión de la información y el conocimiento. La gestión de la información es la coordinación eficiente y eficaz de la información procedente de fuentes internas y externas (1). Un ejemplo de estas fuentes lo constituyen las redes sociales, que actualmente juegan un papel primordial en la creación y difusión de la información entre personas.

Desde su creación, las redes sociales han revolucionado de manera trascendental la sociedad; convirtiéndose en una vía fundamental de comunicación y transferencia de información. Por otra parte, las redes sociales han incurrido de forma profunda en el sector político, cultural y comercial; llegando a ser, en cierta medida, una métrica para elementos como la popularidad, la fama, la influencia política, entre otros.

Entre las redes sociales se encuentra Twitter; un servicio de *microblogging* con textos de hasta 140 caracteres. Tiene más de 288 millones de usuarios activos mensualmente (2) que comparten información y opiniones cotidianas en tiempo real. Los mensajes publicados por los usuarios de Twitter son denominados tuits<sup>1</sup>, y diariamente se envían más de 500 millones (2). El gran flujo de información instantánea de esta red social posibilita el intercambio y la comunicación entre los usuarios que la componen; siendo la inmediatez una de sus características más distintiva.

Twitter ofrece el acceso a los datos almacenados a través de varias API<sup>2</sup>. Cada API representa una faceta de Twitter y permite a los desarrolladores construir aplicaciones de forma novedosa y creativa (3). Haciendo uso de las API, se han creado un conjunto de aplicaciones online que permiten llevar esta información a estadísticas haciendo uso de tablas y gráficos que responden a necesidades de los usuarios. Algunos ejemplos de estas aplicaciones son: Topsy, Followthehashtag, Keyhole y Hashtags.org. Cada una de estas aplicaciones ofrece un conjunto de funcionalidades que permiten al usuario obtener la información contenida en Twitter organizada en tablas o gráficos.

---

<sup>1</sup> Tuit: término de la castellanización de "tweet", que es como originalmente se escribe, aprobado por la Real Academia Española.

<sup>2</sup> API: Application Programming Interface; Interfaz de Programación de Aplicaciones.

La Universidad de las Ciencias Informáticas (UCI) está conformada por varios centros de desarrollo en la esfera de producción; entre los que se encuentra el Centro de Ideoinformática (CIDI). Uno de los departamentos de este centro es el Departamento de Operaciones Web y Análisis de Información (DOWAI), el cual tiene entre sus funciones monitorear y analizar el flujo de información sobre Cuba o temas de interés que se difunden en Internet. Una de las líneas de trabajo de DOWAI enfoca esta función en las redes sociales de Internet, fundamentalmente en Twitter.

Con el fin de analizar tendencias de los temas de interés seguidos en Twitter, se elaboran informes que contienen datos estadísticos que facilitan el análisis de los temas. Para la realización de estos informes, el Centro utiliza aplicaciones gratuitas disponibles en Internet y desarrolladas por terceros. Estas aplicaciones ofrecen información que es recolectada manualmente para elaborar los informes del Departamento.

Por otra parte, los datos brindados por estas herramientas no satisfacen totalmente las necesidades del Centro en cuanto al análisis de los temas de interés; trayendo consigo dificultades para la combinación de datos para crear nuevos datos estadísticos que limitan la toma de decisiones y la realización de estudios basados en la información de Twitter. Además, se carece de una herramienta que sea capaz de elaborar informes de forma autónoma limitándose la precisión de los resultados obtenidos pues se utilizan colecciones de datos externos que no siempre coinciden con la información real.

La situación descrita anteriormente permite identificar el siguiente **problema a resolver**: ¿Cómo elaborar reportes estadísticos automáticos sobre los datos recolectados utilizando la red social Twitter?

Por tanto, el **objeto de estudio** se centra en: la gestión de reportes estadísticos, definiendo como **campo de acción** la gestión de reportes estadísticos sobre los datos recolectados haciendo uso de la red social Twitter.

Para dar solución a la problemática planteada anteriormente, se propone como **objetivo general**: Desarrollar una aplicación para la gestión de reportes estadísticos automáticos sobre la red social Twitter según las necesidades del Departamento DOWAI.

Para garantizar el cumplimiento del objetivo general, se definen los siguientes **objetivos específicos**:

1. Valorar el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales para la elaboración de reportes estadísticos sobre los datos recolectados de la red social Twitter.

2. Diseñar una aplicación que permita la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
3. Implementar una aplicación que permita la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
4. Validar el correcto funcionamiento de la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.

Para guiar la investigación se plantea como **idea a defender**: el desarrollo de una aplicación especializada en la elaboración de reportes estadísticos garantizará el estudio de la información contenida en la red social Twitter.

Para cumplir con los objetivos se proponen las siguientes **tareas de investigación**:

1. Valoración del estado del arte de las herramientas o aplicaciones que realicen reportes estadísticos sobre los datos recolectados de la red social Twitter.
2. Definición de la metodología a utilizar en el desarrollo de la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
3. Identificación de las variables para la realización de los reportes estadísticos sobre los datos recolectados de la red social Twitter, según los intereses de DOWAI.
4. Diseño de la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
5. Implementación de la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
6. Elaboración de la documentación de los artefactos generados durante los procesos de análisis, diseño e implementación de la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.
7. Aplicación de las pruebas de funcionalidad a la aplicación para la creación de reportes estadísticos sobre los datos recolectados de la red social Twitter.

A continuación se muestran los **métodos científicos** utilizados para el desarrollo de esta investigación:

**Métodos teóricos:** Permiten estudiar las características del problema que no son observables directamente.

- **Método Histórico-Lógico:** Se utiliza para analizar y estudiar las características actuales, conceptos y evolución de los elementos relacionados con el desarrollo e implementación de una aplicación para la elaboración de reportes estadísticos sobre los datos contenidos en la red social Twitter. Se investigaron las características actuales, conceptos, términos y vocabularios propios de las redes sociales y la gestión del conocimiento.
- **Método Analítico-Sintético:** El empleo de este método permitió realizar un estudio y análisis de toda la bibliografía consultada referente a las herramientas que utilizan a Twitter para generar estadísticas; así como el análisis de las tecnologías y metodología a utilizar, para luego sintetizar los elementos relevantes para el desarrollo de la solución.

**Métodos Empíricos:** Permiten revelar las relaciones esenciales y las características del objeto de estudio.

- **Entrevista:** Este método fue de gran utilidad para recopilar información cualitativa. Con dicha información se pudo precisar el problema a resolver y las necesidades del cliente. Para el efecto se elaboraron cuestionarios no estructurados, es decir, conformados por preguntas abiertas que fueron destinadas al cliente.

A partir de lo anteriormente planteado los **resultados esperados** serán: una aplicación que contribuya a la elaboración de estadísticas sobre los datos recolectados de la red social Twitter; así como, la documentación de los artefactos generados durante el desarrollo de la misma.

La estructura de este trabajo está basada en tres capítulos, enumerados a continuación:

### **Capítulo 1: Fundamentación teórica**

En este capítulo se realiza un estudio profundo sobre aspectos referentes a la red social Twitter, definiéndose conceptos fundamentales relacionados con Twitter, así como sus características. Se hace referencia a la metodología de desarrollo de software y las herramientas seleccionadas para el desarrollo del trabajo.

### **Capítulo 2: Propuesta de solución**

Se muestran todas las características que tendrá el sistema, partiendo de la descripción general de la propuesta del mismo. Se identifican y se describen los requerimientos funcionales y no funcionales. Se realiza la modelación y descripción de los artefactos generados por la metodología seleccionada.

### **Capítulo 3: Implementación y pruebas**

En este capítulo se presentan las pruebas realizadas a la aplicación para comprobar el correcto funcionamiento de las principales funcionalidades del sistema; así como, un análisis de los resultados obtenidos a partir de las pruebas aplicadas. Se definen los estándares de codificación que se emplearan en la implementación del sistema.

# **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se ofrece una breve introducción a los conceptos abordados a lo largo de la investigación. Se incluyen los resultados del análisis del estado del arte de las diferentes herramientas. Se exponen las principales características de las tecnologías utilizadas para la implementación de la solución propuesta; así como, las herramientas y metodología a utilizar.

## **FUNDAMENTOS TEÓRICOS**

### **Definición de redes sociales**

Las redes son formas de interacción social, definidas como un intercambio dinámico entre personas, grupos e instituciones. Una red social es un sistema abierto y en construcción permanente involucrando a conjuntos de personas que se identifican con las mismas necesidades y problemáticas y que se organizan para potenciar sus recursos (4).

Las redes sociales se han convertido en un fenómeno social, cuyo origen proviene de la filosofía Web 2.0. Son plataformas de comunidades virtuales que proporcionan información e interconectan a personas con afinidades comunes. Las redes sociales son uno de los mejores paradigmas de la Web 2.0 y la construcción de opinión en el entorno digital (5).

En el ámbito de la informática, la red social hace alusión al sitio web que estas personas utilizan para generar su perfil, compartir información, colaborar en la generación de contenidos y participar en movimientos sociales. Las relaciones de los usuarios pueden ser de muy diversa índole y van desde los negocios hasta la amistad (6).

Una vez analizadas las definiciones anteriores, se decide tener en cuenta para esta investigación la siguiente definición de red social: conjunto de personas dentro de una comunidad con afinidades comunes, que intercambian informaciones de un tema determinado.

### **Red Social Twitter**

Twitter es una red social en línea; una plataforma de microblogging que permite a sus usuarios enviar y recibir mensajes cortos de texto de hasta 140 caracteres, llamados "tuits". Estos mensajes se muestran en



la página inicial de los usuarios. Los usuarios pueden suscribirse a los tuits de otros usuarios; a esta acción le llama "seguir" y a los usuarios abonados, "seguidores". Por defecto, los mensajes son públicos, aunque se pueden difundir privadamente mostrándose únicamente a los seguidores de la cuenta que así lo desea. A través de los mensajes se pueden publicar palabras, números, signos, imágenes y enlaces. Un tuit puede ser retuiteado<sup>3</sup>; o sea, repetido o replicado por otros usuarios. Además, se puede marcar como favorito, lo cual significa el grado de agrado en el resto de los usuarios de Twitter.

Los usuarios pueden categorizar mensajes sobre un mismo tema mediante el uso de etiquetas (*hashtag*) o almohadillas – palabras iniciadas mediante el uso de "#". De forma similar, la "@" (arroba) seguida de un nombre de usuario se usa para mencionar o contestar a otros usuarios.

### Estructura de un tuit

A continuación se muestra una tabla con los datos de la estructura interna de un tuit, donde se explica por cada campo el tipo de dato y una breve descripción del mismo.

**Tabla 1.** Estructura de un tuit (7).

Campo	Tipo	Descripción
annotations	Object	Futuro hogar para las anotaciones de estado.
contributors	Collection of Contributors	Una colección breve de objetos de usuario (por lo general solo uno) que indica a los usuarios que contribuyeron a la autoría del tuit, en nombre del autor oficial del tuit.
coordinates	Coordinates	Representa la ubicación geográfica del tuit según lo informado por la aplicación del usuario o cliente.
created_at	String	Fecha de creación del tuit

<sup>3</sup> retuiteado: término de la castellanización de "Retweet", que es como originalmente se escribe, aprobado por la Real Academia Española.

current_user_retweet	Object	Sobre los métodos de apoyo del parámetro include_my_retweet, cuando se establece en true. Detalles del ID del tuit del propio retuit del usuario (si existe) de este tuit.
entities	Entities	Entidades que han sido analizados por el texto del tuit.
favorite_count	Integer	Indica aproximadamente cuántas veces este tuit ha sido "favorito" de los usuarios de Twitter.
favorited	Boolean	Indica si este tuit ha sido favorito por el usuario que se autentica.
filter_level	String	Indica el valor máximo del parámetro filter_level que puede ser utilizado y que flujo sigue este tuit.
id	Int64	Es la representación entera del identificador único para este tuit.
id_str	String	Es la representación de cadena del identificador único para este tuit.
in_reply_to_screen_name	String	Si el tuit representado es una respuesta, este campo contiene el nombre de pantalla del autor del tuit original.
in_reply_to_status_id	Int64	Si el tuit representado es una respuesta, este campo contiene la representación entera del ID original del tuit.
in_reply_to_status_id_str	String	Si el tuit representado es una respuesta, este campo contiene la representación de cadena de ID del tuit original.
in_reply_to_user_id	Int64	Si el tuit representado es una respuesta, este campo contiene la representación entera del original autor del tuit.

in_reply_to_user_id_str	String	Si el tuit representado es una respuesta, este campo contiene la representación de cadena del original del autor del tuit.
lang	String	Lenguaje de la cuenta.
place	Places	Cuando está presente, indica que el tuit se asocia (pero no necesariamente procedente de) a un lugar.
possibly_sensitive	Boolean	El significado del campo no se refiere a los contenidos de Twitter en sí, sino que es un indicador de que la URL contenida en el tuit puede incluir contenido o los medios de comunicación identificados como contenido sensible.
scopes	Object	Un conjunto de pares clave-valor que indica la entrega contextual prevista de la que contiene el tuit.
retweet_count	Int	Número de veces que un tuit ha sido retuiteado.
retweeted	Boolean	Indica si este tuit ha sido retuiteado por el usuario que se autentica.
retweeted_status	Tweet	Los usuarios pueden ampliar la difusión de tuits escritos por otros usuarios retuiteado.
source	String	Se utiliza para publicar el tuit, como una cadena con formato HTML.
text	String	Texto del tuit.
truncated	Boolean	Truncado a 140 caracteres. Sólo posible desde SMS.
user	Users	El usuario que publicó el tuit.

### APLICACIONES DE TWITTER

Para el desarrollo de la solución propuesta se realizó un estudio de las funcionalidades de aplicaciones que permiten llevar la información que brinda Twitter a estadísticas y gráficos; y de esta forma poder ser monitoreada y utilizada con fines específicos. De cada una de estas aplicaciones se exponen sus características, lo cual permite determinar cuáles satisfacen las necesidades del Departamento. Estas herramientas estudiadas son utilizadas por el Departamento DOWAI para la realización de los informes estadísticos. Todas las herramientas estudiadas son gratuitas.

#### **Topsy**

Topsy es un motor de búsqueda en tiempo real que indexa y clasifica los resultados de la búsqueda sobre la base de las conversaciones más influyentes de los usuarios de Twitter. Además, es una herramienta de monitorización de tendencias en Twitter. Es una aplicación que permite clasificar la búsqueda por enlaces, imágenes, tuits, videos, e *influencers* (personas con una opinión relevante en Twitter ya sea por su profesión, el número de seguidores, etc.). En los términos de búsqueda se pueden monitorizar etiquetas que se grafican en de acuerdo a su comportamiento durante los últimos 30 días (versión gratuita).

#### **Followthehashtag**

Followthehashtag es una herramienta de medición, obtención de datos y análisis de contenidos en Twitter. Su principal uso es la investigación y obtención de datos sociales. Sus capacidades son: Búsqueda simple y avanzada incluyendo datos geolocalizados, *Keyword tracking* (seguimiento de palabras clave). Utiliza los datos históricos en Twitter desde 2006. Permite exportación a formato PDF, análisis de contenido, de influencia, de género y etiquetas. Es una herramienta gratis que analiza hasta 1500 tuits de un *hashtag* o términos de búsqueda, en un período determinado. Indica tuits emitidos, media de tuits por hora y usuarios, de un modo muy visual.

#### **Keyhole**

Esta herramienta se centra en el análisis de etiquetas. Empezando por las veces que se ha repetido la etiqueta de búsqueda, el número de cuentas que lo han emitido, el alcance y las impresiones. Ofrece los

mejores tuits que la contienen, los porcentajes de retuits o respuestas que han utilizado la etiqueta, la localización, el género de los tuiteros y desde qué plataformas se han publicado los tuits.

**Hashtags**

La versión gratuita de esta herramienta ofrece análisis de una etiqueta en las últimas 24 horas, los usuarios más influyentes que la han utilizado y los últimos tuits. También se puede visualizar las etiquetas más relacionadas a esta, o sea, las que más se han utilizado en conjunto con la etiqueta de búsqueda. No es la mejor de las opciones, sus datos son bastantes limitados pero es una buena opción para recurrir rápidamente y gratis, a datos de las últimas 24 horas. Para acceder a todas las herramientas con que cuenta el servicio se debe pagar de \$49.00 USD a \$249.00 USD por mes.

**Monitor Tuitter**

Esta es una herramienta que permite el análisis de tuits, etiquetas, usuarios, geolocalización, imágenes, enlaces. Ofrece el análisis de lo anteriormente mencionado por un período solo de 24 horas. Es una herramienta desarrollada en Venezuela, a al cual tiene acceso usuarios determinados.

A continuación se muestra una tabla con la comparación entre las aplicaciones estudiadas con respecto a las necesidades del Departamento DOWAI.

**Tabla 2.** Comparación de las aplicaciones de gestión estadística de Twitter.

<b>Aplicaciones/ Necesidades</b>	Topsy	Followthehashtag	Keyhole	Hashtags.org	Monitor Tuitter
Análisis de tuits	X	X	X		X
Análisis de etiquetas/términos			X		X
Análisis de los usuarios que publican		X	X		X
Análisis de geolocalización		X	X		X
Análisis de imágenes	X				X
Análisis de enlaces	X	X	X		X

Comportamiento de tuits/tiempo	X	X	X	X	X
Análisis por período de tiempo	X	X	X		

Dadas las necesidades del Departamento DOWAI y las funcionalidades de las aplicaciones estudiadas se llega a la conclusión de que las mismas por sí solas no satisfacen dichas necesidades y se decide utilizar como guía para el desarrollo de la solución propuesta la herramienta Monitor Tuitter.

### **METODOLOGÍA A UTILIZAR PARA EL DESARROLLO DE LA APLICACIÓN**

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Para el desarrollo de la solución propuesta se hace necesaria la selección de una metodología de desarrollo acorde a las características propias del proyecto y que a su vez garantice la eficiencia y calidad del proceso de desarrollo de software. Con este objetivo fueron analizadas las metodologías ágiles y tradicionales.

**Tabla 3.** Metodologías Tradicionales vs Metodologías Ágiles (8).

<b>Metodologías Tradicionales</b>	<b>Metodologías Ágiles</b>
Rigidez ante los cambios, de manera lenta o moderada.	Flexibilidad ante los cambios del proyecto de forma moderada a rápida.
Los clientes interactúan con el equipo de desarrollo mediante reuniones.	Los clientes hacen parte del equipo de desarrollo.
Grupos de gran tamaño y varias veces distribuidos en diferentes sitios.	Grupos pequeños (promedio 10 participantes in situ) en el mismo lugar.
Dependencia de la arquitectura de software mediante modelos.	Menor dependencia de la arquitectura de software
Poca retroalimentación, lo que extiende el tiempo de entrega.	Continua retroalimentación acortando el tiempo de entrega.
Mínimos roles.	Diversidad de roles.
Basadas en normas de estándares de desarrollo.	Basadas en heurísticas a partir de prácticas de producción de código.
Procesos muy controlados por políticas y normas.	Procesos menos controlados, pocas políticas y normas.

Seguimiento estricto del plan inicial de desarrollo.	Capacidad de respuesta ante los cambios.
--	--

A partir de la tabla anterior, se arriba a la conclusión de que la propuesta de metodología a utilizar sea la metodología ágil; pues se ajusta más a las características del proyecto por ser de poca duración, con un equipo reducido que cumple con las necesidades y objetivos trazados para el desarrollo de software.

### Metodologías ágiles

A continuación se muestran las características de las metodologías ágiles estudiadas.

#### Programación Extrema (XP)

Programación Extrema, por sus siglas en inglés XP<sup>4</sup>, es un enfoque de la ingeniería del software formulado por Kent Beck. Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (9).

Entre las ventajas de XP se encuentran las siguientes:

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- La presión está a lo largo de todo el proyecto y no en una entrega final.

Una de las desventajas de XP es:

---

<sup>4</sup> XP: eXtreme Programming

- Delimitar el alcance del proyecto con los clientes (9).

### **SCRUM + Programación Extrema (SXP)**

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica. Se parte de la introducción de procedimientos ágiles que permiten actualizar los procesos de software para el mejoramiento de la actividad productiva. Fomenta el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma tal que se sepa cómo va el desarrollo del proceso de gestión de software. Por otra parte, XP es una metodología encaminada al desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

SXP consta de cuatro fases principales:

- **Planificación-Definición:** Donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** Es la fase donde se realiza la implementación y construcción del sistema hasta que esté listo para ser entregado, luego de un conjunto de iteraciones de desarrollo.
- **Entrega:** Fase donde se pone en marcha el producto desarrollado y se genera la documentación necesaria para hacer la entrega al cliente.
- **Mantenimiento:** En esta se realiza el soporte para los problemas que pueda presentar el software durante su despliegue.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo (10).



### OpenUP

OpenUP<sup>5</sup> es un proceso mínimo y suficiente, lo que significa que solo 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 (11).

Como principios de OpenUp se exponen los siguientes: (11)

- Colaborar para sincronizar intereses y compartir conocimiento. Este principio promueve prácticas que impulsan un ambiente de equipo saludable, facilitan la colaboración y desarrollan un conocimiento compartido del proyecto.
- Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto. Este principio promueve prácticas que permiten a los participantes de los proyectos desarrollar una solución que maximice los beneficios obtenidos por los participantes y que cumple con los requisitos y restricciones del proyecto.
- Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo.
- Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo. Este principio promueve prácticas que permiten a los equipos de desarrollo obtener retroalimentación temprana y continua de los participantes del proyecto, permitiendo demostrarles incrementos progresivos en la funcionalidad.

El OpenUP está organizado en dos dimensiones diferentes pero interrelacionadas: el método y el proceso. El contenido del método es donde los elementos del método (roles, tareas, artefactos y lineamientos) son definidos, sin tener en cuenta cómo son utilizados en el ciclo de vida del proyecto. El proceso es donde los elementos del método son aplicados de forma ordenada en el tiempo. Muchos ciclos de vida para diferentes proyectos pueden ser creados a partir del mismo conjunto de elementos del método.

El OpenUP estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección

---

<sup>5</sup> OpenUP: Open Unified Process.

para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos.

### **Selección de la metodología a utilizar**

Luego de analizar las características de las metodologías de desarrollo de software antes planteadas, se pudo apreciar que están diseñadas para elevar la calidad de la producción de software y la labor de los desarrolladores. Dadas las particularidades de la presente investigación y el hecho de ser considerado un proyecto de poca duración, con un equipo reducido y está contenida en la base tecnológica definida por el departamento DOWAI del centro CIDI; se selecciona OpenUP como la metodología para el desarrollo de la solución propuesta.

## **TECNOLOGÍAS PARA EL DESARROLLO**

### **Lenguaje de Modelado Unificado**

El Lenguaje de Modelado Unificado, por sus siglas en inglés UML<sup>6</sup> es un grupo de especificaciones de notación orientadas a objeto, las cuales están compuestas por distintos diagramas que representan las diferentes etapas del desarrollo de un software. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para moldear un sistema de software previo al proceso intensivo de escribir código. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Los diagramas más comunes son: diagramas de clases, objetos, casos de uso, estados, secuencias, actividades, colaboraciones y componentes; además los mismos permiten examinar un sistema desde distintos puntos de vista.

Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. Es importante recalcar que en un modelo UML no es necesario que aparezcan todos los diagramas. De hecho, la mayoría de los modelos UML contienen un subconjunto de los diagramas que se han mencionado (12).

---

<sup>6</sup> UML: Unified Modeling Language

Se utiliza como lenguaje de modelado el UML, ya que es capaz de definir, detallar, documentar y construir un sistema de software. Permite modelar los procesos de negocio siendo un lenguaje más abarcador con relación al sistema que se desea implementar. Es un lenguaje fácil de aprender, adaptable a cambios y ofrece gran variedad de diagramas para visualizar el software desde varias vías, facilitando así la relación entre clientes y desarrolladores.

### **Herramientas CASE**

Se puede definir a las herramientas CASE<sup>7</sup> como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (13).

### **Visual Paradigm for UML**

Visual Paradigm for UML es una herramienta profesional que facilita el modelado del ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML soporta las últimas versiones del mismo y la Notación y Modelado de Procesos de Negocios. Ayuda a una más rápida construcción de aplicaciones de calidad y a un menor coste ya que permite generar código desde diagramas y generar documentación.

Visual Paradigm ofrece distintas funcionalidades como:

- Diseño centrado en casos de uso y enfocado al negocio generando un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa en su versión profesional e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones para cada necesidad (14).

### **Rational Rose Enterprise**

Rational Rose Enterprise es una potente herramienta de modelado visual para ayudar en el análisis y diseño de software orientado a objetos. Es la mejor elección para el ambiente de modelado que soporte la

---

<sup>7</sup> CASE: Computer Aided Software Engineering; Ingeniería de Software Asistida por Ordenador.

generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Rational Rose Enterprise posee características adicionales incluidas:

- Es capaz de analizar la calidad del código y de generar código gracias a las prestaciones de sincronización configurable entre el modelo y el código.
- Permite la gestión granular y el uso de modelos con una característica de componentes de modelo que se puede controlar por separado.
- Proporciona el modelado UML para diseños de bases de datos.
- Permite integrar requisitos de datos y aplicación a través de diseños lógicos y físicos.
- Incluye un complemento de modelado web, que proporciona visualización, modelado y herramientas para desarrollar aplicaciones web (15).

### **Selección de la herramienta CASE a utilizar**

Después de realizar un análisis entre las herramientas CASE, se escogió Visual Paradigm for UML 8.0 por su robustez, usabilidad y portabilidad. Cuenta con la licencia para su utilización y soporta el ciclo completo del proceso de desarrollo de software propuesto por la metodología OpenUP. Utiliza UML como lenguaje de modelado, que tiene como propósito visualizar, especificar, construir y documentar proyectos de software. En cambio, Rational Rose es una herramienta muy utilizada pero es software propietario y no es multiplataforma, al contrario de Visual Paradigm for UML que es una herramienta multiplataforma y algunas de sus versiones son gratuitas.

### **Lenguajes de programación**

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos monosémicos (es decir, con sentido único) y una regla principal que resume las demás, todas estas son utilizadas por el programador a través de las cuales crea un programa o subprograma (16).

### **Python**

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90, cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Se trata de un lenguaje de programación multiparadigma, pues soporta orientación a objeto, programación imperativa, programación orientada a aspectos y en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma.

El entorno de ejecución detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos. Posee un rico juego de estructuras de datos que se pueden manipular de modo sencillo. Una ventaja fundamental es la gratuidad de su intérprete, el cual tiene versiones para prácticamente cualquier plataforma en uso: sistemas PC<sup>8</sup> bajo Linux, sistemas PC bajo Microsoft Windows, sistemas Macintosh de Apple, etc. (17).

Algunas de las ventajas del uso de Python son (18):

- **Propósito general:** Se pueden crear todo tipo de programas.
- **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para este.
- **Interpretado:** No se debe compilar el código antes de su ejecución. En realidad sí se realiza una compilación, pero esta se realiza de manera transparente para el programador.
- **Interactivo:** Dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudar a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a Objeto:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables. Además, Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.
- **Funciones y bibliotecas:** Dispone de muchas funciones incorporadas en el propio lenguaje.

---

<sup>8</sup> PC: Personal Computer; Computadora personal.

- **Sintaxis clara:** Tiene una sintaxis muy visual, gracias a una notación *identada* (con márgenes) de obligado cumplimiento. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.
- **Mixto:** Se puede integrar de manera "fácil" con otros lenguajes de programación.
- **Gratuito:** Una ventaja fundamental de Python es la gratuidad de su intérprete.

El uso del lenguaje de programación Python en esta investigación está regido principalmente por las potencialidades de este puesto que cuenta con un amplio número de funciones incorporadas que le facilitan el trabajo al programador. Además, permite la adición de módulos que expanden aún más las facilidades que este brinda. Al contar con una sintaxis de obligatorio cumplimiento permite la obtención de un código claro y ordenado, facilitando su comprensión. Por otra parte Python es capaz de integrarse a otros lenguajes de programación de manera bastante sencilla, lo cual propicia que la aplicación se pueda integrar con otras aplicaciones en caso de ser necesario.

### JavaScript

JavaScript es un lenguaje de programación interpretado, lo cual básicamente se refiere a que no necesita ningún programa intermediario para ejecutarse en un navegador web, es robusto y a la vez ligero (19).

A pesar de no ser un lenguaje orientado a objetos implementa muchas de las características de este paradigma. Se utiliza generalmente, para crear páginas web dinámicas: aquellas que presentan textos, imágenes, animaciones, ventanas que interactúan entre ellos y con el usuario. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en el ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web (20).

### Lenguajes de maquetado y estilo

#### HTML5

HTML<sup>9</sup>, más que el lenguaje de marcado predominante usado para estructurar el contenido web, es realmente una tecnología web que interrelaciona varios estándares. En esta última revisión del lenguaje (HTML5) se incluyen varias características verdaderamente útiles, como son nuevos elementos semánticos,

---

<sup>9</sup> HTML: HyperText Markup Language; Lenguaje de marcas de hipertexto.

mejoras en los formularios, soporte nativo para video y audio, un lienzo (canvas) para el dibujo a través de JavaScript, soporte para almacenamiento local y geolocalización (21).

Algunas de las características que incorpora HTML5 son:

- **Canvas:** es un nuevo componente que permite dibujar en la página, por medio de las funciones de un API, todo tipo de formas, que pueden estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún *plugin*.
- **Etiquetas para contenido específico:** Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- **Nuevas API para interfaz de usuario:** temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, son incorporadas al HTML5 por medio de un API (22).

El uso de lenguaje HTML5 se rige principalmente, por la proporción de mejoras en el desarrollo de los sitios web en cuanto a animaciones, videos, componentes de interfaces complejas, audio entre otros componentes que requieren del uso de *plugins*. Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores.

### CSS3

Cascading Style Sheets (CSS<sup>10</sup>) no es más que un lenguaje de estilo que describe cómo el contenido HTML es presentado al usuario. CSS3 es la última versión de la especificación CSS, incluyendo nuevas características para ayudar a los desarrolladores a resolver numerosos problemas sin la necesidad de un marcado no semántico, complejos scripts o imágenes extra. Estas nuevas características incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones, transparencias, entre otras (23).

---

<sup>10</sup> CSS: Hojas de Estilo en Cascada.

### Bibliotecas a utilizar

#### HighCharts

HighCharts es una biblioteca escrita en JavaScript que permite la creación de gráficas. Ofrece un método fácil e interactivo para insertar gráficas en su sitio web o aplicación web. Es compatible con todos los navegadores modernos incluyendo iPhone/iPad e Internet Explorer desde su versión 6. No es comercial, no se necesita el permiso de los autores para su implementación en sitios web personales o sin fines de lucro. Es abierto, todas las características pueden ser personalizadas permitiendo una gran flexibilidad. Además, HighCharts está escrito solamente con código JavaScript, solo se requiere incluir el archivo highcharts.js (24).

#### gRaphaël

El objetivo de gRaphaël es facilitar la creación de gráficos usando el lenguaje HTML. Se basa en Raphael, una biblioteca de gráficos que contiene gráficos estáticos e interactivos. Permite simplificar el trabajo con gráficos vectoriales en la web. Proporciona un adaptador que hará que el vector sea compatible con los navegadores soportando Firefox 3.0 o superior, Safari 3.0 o superior, Chrome 5.0 o superior, Opera 9.5 o superior e Internet Explorer 6.0 o superior.

Raphael utiliza la recomendación de la W3C<sup>11</sup> y SVG (gráficos vectoriales escalables) como base para la creación de gráficos. Esto significa que para cada objeto gráfico se crea también un objeto DOM<sup>12</sup>, donde se pueden conectar los controladores de eventos de JavaScript. Es desarrollada con una licencia MIT<sup>13</sup> (25).

#### JQuery sparklines

Es un complemento JQuery que genera gráficos directamente en el navegador utilizando datos suministrados, o por medio de JavaScript. Este complemento es compatible con la mayoría de los navegadores modernos y ha sido probado con Firefox 2.0 o superior, Safari 3.0, Opera 9.0, Google Chrome e Internet Explorer 6, 7, 8, 9 y 10, así como iOS 6.0 y Android 3.0.

---

<sup>11</sup> W3C: Consorcio internacional que produce recomendaciones para la World Wide Web

<sup>12</sup> DOM: Modelo de Objetos del Documento

<sup>13</sup> MIT: Licencia de software creada por el Instituto Tecnológico de Massachusetts



Sparklines pretende ser lo suficientemente pequeño como para caber junto a una línea de texto, para dar una impresión rápida de una tendencia o patrón. No tiene la configuración de los gráficos de tamaño completo. Desde la versión 2.0 se puede obtener sobre los minigráficos los datos subyacentes.

El complemento no tiene soporte para añadir texto a un minigráfico, más allá de la proporcionada por la información de herramientas. Puede usar HTML y CSS para agregar texto apropiado al lado del minigráfico. Además se puede utilizar la librería Flot y el complemento Jqplot para generar gráficos más grandes (26).

### **Selección de la biblioteca a utilizar**

Tomando en cuenta lo anteriormente propuesto se decidió utilizar para el desarrollo de la solución la biblioteca HighCharts, debido a que se puede integrar fácilmente y permite obtener los tipos de gráficos que se necesitan para dar solución a los objetivos propuestos. Además, posibilita la administración de las diferentes funciones necesarias para poder adaptar a la solución propuesta los gráficos generados por esta biblioteca.

### **Marco de trabajo**

Un marco de trabajo o *framework* es un esquema, esqueleto o patrón para el desarrollo o la implementación de una aplicación. En otras palabras, un marco de trabajo se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto (27).

Los marcos de trabajo son diseñados con el propósito de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Un *framework* en el desarrollo del software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

### **Django**

Django es un *framework* de alto nivel basado en Python, facilita el desarrollo de aplicaciones web dinámicas, se abstrae de los problemas comunes del desarrollo web y acelera las tareas más frecuentes en la programación (28).

Entre las ventajas de Django se encuentran:

- Permite mostrar y validar formularios de manera muy simple, fácilmente puede manipular el código del formulario adaptándolo a las necesidades de la aplicación y a una respuesta simple para el usuario.
- Convierte los datos enviados por los usuarios (a través de formularios, por ejemplo) en estructuras de datos que pueden ser manipuladas fácilmente.
- A través de plantillas ayuda a separar el contenido de la presentación, evitando tener que manipular la lógica de negocio cuando se tenga que realizar cambios de apariencia en el sitio web.
- Se integra fácilmente con la capa de base de datos, y permite que el trabajo sea más productivo a un nivel de abstracción más alto.
- Aparte de las ventajas de Python, Django permite un desarrollo rápido.
- Django mantiene de una manera estricta un diseño limpio en su código y permite entender más fácilmente las aplicaciones.
- Django promueve el acoplamiento débil, diferentes módulos de la aplicación deberían ser intercambiables y comunicarse con otros módulos a través de API limpias y concisas.
- Django utiliza la arquitectura MTV<sup>14</sup>, es el patrón que dice que el código para definir y acceder a los datos (modelo) está separado de la lógica de negocio (la vista) que está separada de la interfaz de usuario (la plantilla).

### Web2py

Web2py es un *framework* empresarial libre y de código abierto para desarrollo ágil de aplicaciones web rápidas, escalables y seguras basadas en bases de datos. Se enfoca en el desarrollo rápido y utiliza la arquitectura Modelo Vista Controlador. Web2py fue también inspirado por Django y, como Django, tiene la habilidad de generar formularios para las tablas de la base de datos e incluye un extenso conjunto de

---

<sup>14</sup> MTV: Modelo-Plantilla-Vista.

validadores. Web2py se diferencia de Django porque es más compacto, fácil de leer y no tiene ningún archivo de configuración a nivel de proyecto (29).

Entre sus características están:

- Incluye un servidor web de capas SSL<sup>15</sup> y transmisión de datos, una base de datos relacional, una interfaz administrativa y de desarrollo integrada basada en web, una Capa de Abstracción de Base de Datos, soporte para internacionalización, métodos múltiples de autenticación, control de acceso basado en roles y registro de errores.
- Corre en Windows, Mac, Unix/Linux, Google App Engine, Amazon EC2, y prácticamente en cualquier hospedaje web vía Python 2.4/2.5/2.6, o Java con Jython.
- Presenta soporte para: SQLite, PostgreSQL, MySQL, MSSQL, FireBird, Oracle, IBM DB2, Informix, Ingres, y Google App Engine.
- Aplica buenas prácticas de la Ingeniería de Software (diseño Modelo-Vista-Controlador, Validación de formularios en el servidor) lo que hace al código más legible y escalable (29).

### **Selección del marco de trabajo a utilizar**

Por la gran facilidad y el ahorro de tiempo que brinda Django, y por las características anteriormente mencionadas, cuando se utiliza asociado con el lenguaje de programación Python, es que se decide trabajar con este entorno de desarrollo en su versión 1.7.

### **Entorno de Desarrollo Integrado (IDE)**

Los IDE son un conjunto de herramientas para el programador, que suelen integrar un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores, integración con sistemas controladores de versiones o repositorios.

### **PyCharm Professional Edition**

---

<sup>15</sup> SSL: Capa de Conexión Segura

PyCharm es un IDE que se utiliza para la programación en Python. Proporciona análisis de código, un depurador gráfico, un probador de unidad integrada, integración con sistemas de control de versiones (VCSes), y apoya el desarrollo web con Django. PyCharm es desarrollado por la empresa checa JetBrains.

Dentro de todas las ventajas de PyCharm se encuentran:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con frameworks web como: Django, Flask, Pyramid, Web2Py.
- Frameworks javascripts: jQuery, AngularJS.
- Debugger avanzado de Python y JavaScript.
- Integración con lenguajes de plantillas: Mako, Jinja2, Django Template.
- Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.
- Sistemas de control de versiones: Git, CVS, Mercurial (30).

### **Sistema Gestor de Base de Datos NoSQL**

Los sistemas de gestión de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Su propósito es manejar de forma clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización (31).

Existen bases de datos relacionales y documentales (o no relacionales). El auge de la Web está mirando las bases de datos documentales como clave en sus negocios, por su capacidad de escalabilidad, rendimiento y alta disponibilidad que los actuales sistemas de bases de datos relacionales (32).

En las bases de datos documentales los datos están fuertemente orientados a documentos, haciéndolos más acordes y naturales con el modelo de datos clave/valor que con el modelo de datos relacional. El entorno de desarrollo está fuertemente orientado a objetos, y una base de datos clave/valor podría reducir la necesidad de código tubería. El almacenamiento de datos es económico y se integra fácilmente con la plataforma de servicios web del proveedor.

### **MongoDB**

MongoDB es una base de datos orientada a documentos, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON (del inglés Binary JSON), que es una representación binaria de JSON (del inglés, JavaScript Object Notation) (33).

En MongoDB, cada registro o conjunto de datos se denomina documento. Los documentos se pueden agrupar en colecciones, las cuales se podría decir que son el equivalente a las tablas en una base de datos relacional (solo que las colecciones pueden almacenar documentos con muy diferentes formatos, en lugar de estar sometidos a un esquema fijo). Se pueden crear índices para algunos atributos de los documentos, de modo que MongoDB mantendrá una estructura interna eficiente para el acceso a la información por los contenidos de estos atributos.

Las características que más se destaca de MongoDB son su velocidad y su rico pero sencillo sistema de consulta de los contenidos de la base de datos. Se podría decir que alcanza un balance perfecto entre rendimiento y funcionalidad (34).

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección (concepto similar a una tabla de una base de datos relacional), pueden tener esquemas diferentes.

### **Hbase**

Hbase es una base de datos distribuida de código abierto no relacional. Está inspirada en el BigTable de Google, y es parte del proyecto Hadoop del Apache Software Foundation. Está escrito en Java, con lo que es portable a cualquier sistema operativo. Hbase se centra en el análisis de grandes cantidades de datos, y no en procesar datos rápidamente en tiempo real (35).

Soporta una variedad muy grande de protocolos para comunicarse con las distintas aplicaciones. Tiene una interfaz Thrift a los datos, haciéndolos accesibles desde cualquier lenguaje de programación. Hbase no tiene un punto central de fallo, los datos están distribuidos en todos los nodos de la aplicación. Tiene parámetros configurables para regular el nivel de consistencia de los datos.

Características de Hbase:

- Compartición automática. Las tablas Hbase se distribuyen en el clúster a través de las regiones que son divididas automáticamente y se re-distribuyen tanto como los datos crecen.
- Soporta el procesamiento masivamente paralelizado a través de MapReduce (36).

### Selección del Sistema Gestor de Base de Datos NoSQL.

Se utiliza como sistema gestor de base de datos para el desarrollo de la propuesta de solución, MongoDB en su versión estable 2.6.0; la cual incluye la replicación y la auto-fragmentación. Es un SGBS relativamente nuevo, con licencia libre, orientado a documentos por lo que no implementa el concepto de tablas, esquemas SQL o columnas, no tiene transacciones, *joins* (uniones), claves foráneas o cualquier otra característica que dificulte la comodidad y la rapidez con que se desea trabajar. Es un sistema que se destaca por su velocidad y facilidad de uso, considerada una base de datos muy diferente a las que se está acostumbrado pero sin dudas alcanza un balance perfecto entre rendimiento y funcionalidad.

### Herramienta para Pruebas de Carga y Estrés

#### JMeter

Es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Permite realizar pruebas Web clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y WebServices (en Beta). Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Se puede activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario (37).

Para la elección de la herramienta candidata se analizaron los siguientes aspectos:

- **Funcionalidades:** Prestaciones de la herramienta, elementos funcionales, respuesta a los requerimientos del sistema. Modo de adquisición: Si el software se obtiene por medio de una licencia de pago o es gratuito.
- **Materiales de apoyo:** Si existe documentación de la herramienta que ofrezca una guía para su utilización.

- **Entorno en el que fue desarrollada:** Si es un software a código abierto desarrollado en el ambiente de software libre o si es un software del cual no se puede obtener información de su código o no se puede modificar y adecuar a los ambientes de trabajo del proyecto por ser un software propietario (37).

Atendiendo a la clasificación expuesta anteriormente la herramienta JMeter se clasifica como dinámica de medida por su tipo de función, como manipula datos de entrada para la ejecución de las pruebas y necesita la comparación de los resultados para la evaluación, recurre al método de caja negra. Además se clasifica por su especialización en una herramienta de grabación y reproducción.

La utilización del JMeter supone un 95 % de tiempo menos para la realización de estas pruebas. La mayor inversión de tiempo que se necesita para la realización de las pruebas es la fase de estudio de los casos de uso críticos en la aplicación web y la elaboración del plan de pruebas en la herramienta. Permite almacenar los resultados de la prueba y se generan gráficos que representan los aspectos que se han probado.

### **QALoad**

QALoad de la empresa Compuware es una herramienta de automatización de pruebas de carga para web, Java, .NET, aplicaciones ERP (Planificación de recursos empresariales) y CRM (gestión de relaciones con los clientes y ambientes distribuidos). Simula miles de usuarios desempeñando transacciones clave del negocio de una aplicación para asegurar su desempeño y escalabilidad a su puesta en producción. Con esta herramienta, los equipos de prueba pueden rápidamente detectar problemas, optimizar el desempeño de los sistemas y ayudar a asegurar un despliegue de aplicaciones exitoso (38).

### **Selección de la herramienta a utilizar**

Se decide utilizar la herramienta JMeter para las pruebas de carga y estrés por las funcionalidades que presenta y por ser un software libre y gratis. Otro aspecto a favor de esta herramienta es su desarrollo en el marco del software libre, política que apoya la Universidad y que por las características que tiene, existe un por ciento superior de confianza en la utilización de la misma ya que se puede contar con todo el código que genera la herramienta. Se descarta la herramienta QALoad porque no se usa para Python.

## **CONCLUSIONES DEL CAPÍTULO**

Durante la realización de este capítulo se plantearon conceptos fundamentales relacionados con la investigación; así como, el estudio de las diferentes herramientas existentes que permiten llevar la información que brinda la red social Twitter a estadísticas por medio de tablas y gráficos, determinándose que las mismas por sí solas no solucionan las necesidades del departamento DOWAI. Se define la metodología de desarrollo OpenUp como metodología a utilizar, la cual aporta los conocimientos necesarios para lograr el desarrollo de la aplicación. Se expusieron las tecnologías y herramientas necesarias para el desarrollo del software en busca de la solución a la problemática planteada en el presente trabajo investigativo.



## **CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.**

En el presente capítulo se describen las funcionalidades de la aplicación web, destacando sus características principales. Los artefactos que posee la metodología de desarrollo OpenUp, propuesta en el capítulo anterior, son brevemente detallados y explicados. Para lograr entender mejor su contexto se define el modelo de dominio y se describen los requisitos funcionales y no funcionales de la aplicación web, los casos de usos y las tareas de ingeniería asociadas a las mismas.

### **TUITZER COMO PROPUESTA DE SOLUCIÓN**

Luego de haber realizado un estudio y análisis de las necesidades existentes en el Centro CIDI, específicamente en el Departamento DOWAI, en cuanto a la obtención de datos de la red social Twitter y la creación de informes estadísticos basados en dichos datos y teniendo en cuenta de que no se tiene una aplicación propia que dé solución a dichas necesidades, se propone desarrollar la aplicación Tuitzer. Esta aplicación debe aportar soluciones reales y satisfactorias a las necesidades planteadas.

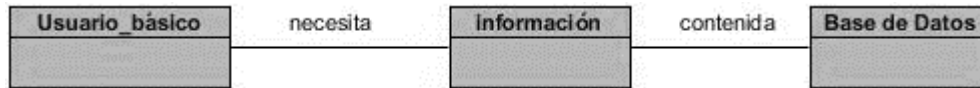
La aplicación gestionará procesos fundamentales como el análisis de tuits generados en un intervalo de tiempo especificado dada una búsqueda de parámetros de entrada para la extracción de los datos de interés. Además, gestionará la conversión de estos datos a gráficos y tablas para una mejor comprensión y manejo.

Se espera que Tuitzer se convierta en una herramienta para el control y monitoreo de las tendencias de los temas de interés seguidos en Twitter. La aplicación quedará desarrollada de forma tal que solo permita el acceso a sus funcionalidades a aquellos usuarios que cuenten con los permisos requeridos para esto.

### **MODELO DE DOMINIO**

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Este representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de software.

Debido a que los procesos del negocio no se encuentran bien definidos se decidió desarrollar un modelo de dominio que represente los conceptos relacionados con estos y las relaciones que se establecen entre ellos (39).



**Imagen 1.** Diagrama de clases del modelo de dominio.

A continuación se muestra la descripción de los conceptos identificados en el modelo de dominio.

- **Usuario\_básico:** Representa a los usuarios autenticados que realizan búsquedas en la aplicación.
- **Base de Datos:** Representa la base de datos MongoDB.
- **Información:** Contiene los datos de la base de datos.

### REQUERIMIENTOS FUNCIONALES

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir (40). A continuación se muestra un listado de requisitos.

RF1. Gestionar usuario.

RF1.1. Crear usuario.

RF1.2. Eliminar usuario.

RF1.3. Modificar permisos de usuario.

RF2. Autenticar usuario.

RF3. Cerrar sesión de usuario.

RF4. Mostrar la cantidad de tuits con una etiqueta determinada en un intervalo de tiempo.

RF5. Graficar el comportamiento de la búsqueda en un espacio de tiempo.

RF6. Mostrar una lista con los 10 primeros usuarios más activos.

RF7. Mostrar una lista con los 10 primeros usuarios más retuiteados.

RF8. Mostrar una lista con los 10 primeros usuarios más influyentes.

RF9. Mostrar una lista con los 10 primeros usuarios más favoritos.

RF10. Mostrar una lista con los 10 primeros mensajes más retuiteados.

- RF11. Mostrar una lista con los 10 primeros mensajes más favoritos.
- RF12. Listar los 10 primeros países desde donde se ha publicado con la búsqueda.
- RF13. Listar los 10 primeros enlaces más posteados.
- RF14. Listar las 10 imágenes que más se han retuiteado.
- RF15. Listar las 10 etiquetas más relacionadas con la búsqueda.
- RF16. Listar las 10 etiquetas más utilizadas en un intervalo de tiempo.
- RF17. Buscar datos específicos de usuarios entrados por parámetros.

### REQUERIMIENTOS NO FUNCIONALES

Los requisitos no funcionales son las propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos, los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente, están vinculados a requerimientos funcionales, es decir, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (40).

Para el sistema propuesto se han definido los siguientes requisitos no funcionales:

- **Seguridad:** Los usuarios deberán estar correctamente autenticados por sus roles correspondientes, para lograr visualizar cada uno de los elementos relacionados con el alcance de su trabajo, y a su vez, tener acceso a los elementos necesarios para su desarrollo laboral. Solo puede existir una sesión abierta por usuario.
- **Portabilidad:** El sistema podrá ser ejecutado sobre los sistemas operativos GNU/Linux o Windows, permitiendo una fácil migración haciendo uso de estándares y tecnologías de código abierto.
- **Apariencia o interfaz externa:** El sistema informático contará con un diseño de interfaz sencillo, agradable, sugerente e intuitivo, de fácil entendimiento.
- **Usabilidad:** La interfaz y los mensajes para interactuar con el usuario, así como los mensajes de error, serán en el lenguaje español. Los mensajes de error deben ser lo suficientemente informativos para dar a conocer la severidad del error.
- **Software:** Para trabajar con la aplicación se debe disponer de un sistema operativo GNU/Linux, Windows 7 o superior. Navegador web con capacidad de interpretación de JavaScript, CSS3 y

HTML5. Preferiblemente Mozilla Firefox 20 o superior.

- **Hardware:** Como características son: en un servidor con 8GB de memoria RAM como requerimiento mínimo, procesador Core i3 o superior y debe contar además con un almacenamiento en disco de 500GB. Como características de las PC\_Cliente son: una PC con 4GB de memoria RAM, procesador Core i3 o superior.
- **Legales:** El sistema y la documentación del mismo, pertenecen a la Universidad de las Ciencias Informáticas. El sistema es desarrollado con herramientas libres lo que permite independencia tecnológica.

### MODELO DE CASOS DE USO DEL SISTEMA

En esta sección se definen los actores y se describen los casos de uso (CU) principales del sistema, quedando conformado el Diagrama de Casos de Uso del Sistema.

#### Definición de los actores del sistema

Los actores del sistema especifican un rol jugado por un usuario o cualquier otro sistema externo que interactúa con la aplicación. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema (41).

A continuación se muestran los actores y la justificación que tienen en el sistema.

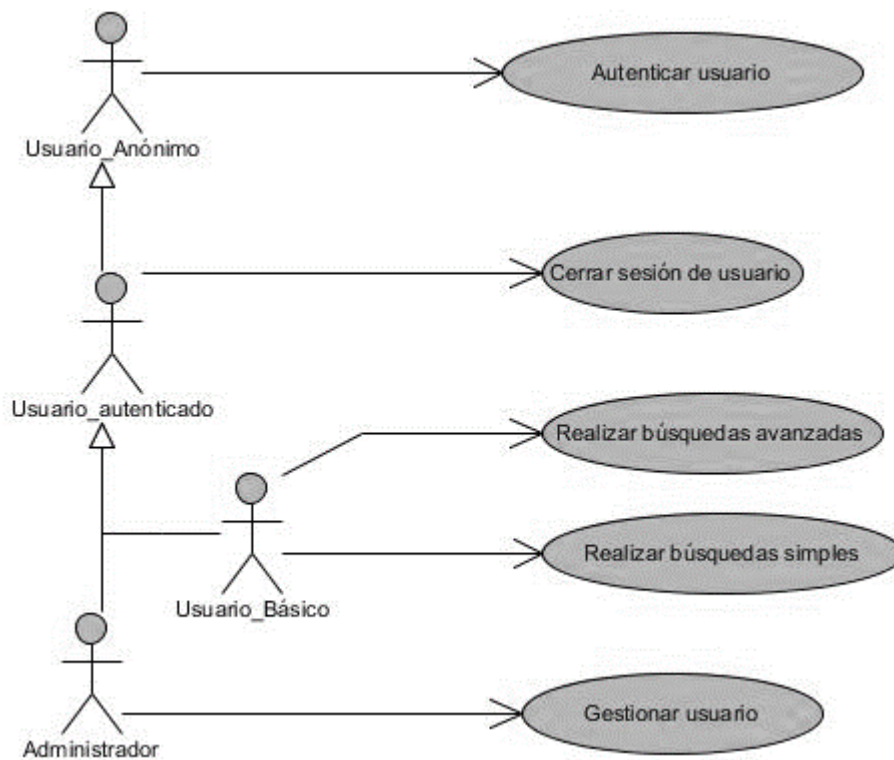
**Tabla 4.** Justificación de los actores del sistema.

Actores del Sistema	Justificación
Usuario básico	El actor puede realizar las acciones relacionadas con las búsquedas.
Administrador	El actor puede realizar todas las acciones relativas a la administración y la gestión de los usuarios.
Usuario autenticado	El actor representa a los usuarios que están autenticados en el sistema.

Usuario anónimo	El actor representa a los usuarios que interactúan con el sistema sin estar autenticados. Tienen solo acceso a la interfaz de autenticación del sistema.
-----------------	--

### Diagrama de CU del sistema

El diagrama de casos de uso documenta el comportamiento de un sistema desde el punto de vista del usuario. Representan las funciones que un sistema puede ejecutar (42).



**Imagen 2.** Diagrama de casos de uso.

### Patrones de CU

La utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Un patrón de caso de uso es un diseño probado en un modelo de casos de uso, junto con

una descripción del contexto en el cual será usado y las consecuencias que tendrá su aplicación en el modelo (43).

**Actores múltiples:** Es un patrón de estructura que plantea que cuando dos actores juegan el mismo papel hacia un caso de uso se representa otro actor, del que heredan los actores que comparten este rol (44).

### Descripción de los CU

Para una mejor comprensión de cómo fueron descritos los casos de usos a continuación se muestran las descripciones de los casos de usos del sistema CU1 Autenticar usuario y CU5 Gestionar usuario, el resto de las descripciones se encuentran en el [Anexo # 1](#).

**Tabla 5.** Descripción CU Autenticar usuario.

<b>Objetivo</b>	Autenticar usuario en el sistema	
<b>Actores</b>	Usuario anónimo	
<b>Resumen</b>	El CU se inicia cuando un usuario intenta autenticarse en el sistema introduciendo sus datos, el CU termina cuando el usuario es autenticado.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe ser anónimo.	
<b>Postcondiciones</b>	El usuario es redireccionado hacia la interfaz correspondiente a su rol.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Autenticar usuario&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede al sistema	
2.		Muestra la interfaz de autenticación.
3.	Introduce los datos.	
4.		Comprueba que los datos cumplan con los requisitos necesarios para que sea válido.
5.		Comprueba los datos del usuario con la base de datos.

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

6.		El usuario es autenticado y redireccionado.
7.		Termina el CU.
<b>Flujos alternos</b>		
4. Los datos no son válidos.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Flujos alternos</b>		
5. Los datos del usuario son incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

**Tabla 6.** Descripción CU Gestionar usuario.

<b>Objetivo</b>	Crear, eliminar usuarios y modificar permisos de usuarios
<b>Actores</b>	Administrador
<b>Resumen</b>	Este CU describe cómo se crean, eliminan usuarios y modifican permisos de usuarios.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Media
<b>Precondiciones</b>	Que el usuario esté autenticado en el sistema.
<b>Postcondiciones</b>	
<b>Flujo de eventos</b>	
<b>Flujo básico &lt;Gestionar usuario&gt;</b>	

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

	Actor	Sistema
1.	Se autentica en el sistema.	
2.		Muestra una interfaz para que escoja la opción de: <ul style="list-style-type: none"> <li>• Crear usuario</li> <li>• Eliminar usuario</li> <li>• Cambiar permisos</li> </ul>
3.	Si se escoge la opción: <ul style="list-style-type: none"> <li>• Crear usuario ir a la interfaz crear usuario</li> <li>• Eliminar usuario ir a la interfaz eliminar usuario.</li> <li>• Cambiar permisos ir a la interfaz cambiar permisos.</li> </ul>	
<b>Flujos alternos</b>		
1. No se puede autentificar usuario		
	Actor	Sistema
2.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<ul style="list-style-type: none"> <li>• <b>Sección 2: “Crear usuario”</b></li> </ul>		
<b>Flujo básico &lt;Gestionar usuario: Crear usuario&gt;</b>		
	Actor	Sistema
1.	Selecciona la opción “Crear usuario”.	
2.		Muestra la interfaz crear usuario.
3.	Introduce los datos correspondientes.	
4.		Comprueba que los datos cumplan con los requisitos necesarios para que sea válido.
5.		Comprueba que no existan los datos en la base de datos.
<b>Flujos alternos</b>		
4. Los datos no son válidos.		
	Actor	Sistema



## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Flujos alternos</b>		
5. Los datos existen.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<ul style="list-style-type: none"> <li>• <b>Sección 3: “Eliminar usuario”</b></li> </ul>		
<b>Flujo básico &lt; Gestionar usuario: Eliminar usuario &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Eliminar usuario”.	
2.		Muestra la interfaz eliminar usuario.
3.	Selecciona usuario a eliminar y hace clic en el botón “Eliminar”.	
4.		Busca usuario seleccionado en la base de datos y lo elimina.
<b>Flujos alternos</b>		
3. Hace clic en el botón “Eliminar” sin haber seleccionado un usuario.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<ul style="list-style-type: none"> <li>• <b>Sección 4: “Cambiar permisos”</b></li> </ul>		
<b>Flujo básico &lt; Gestionar usuario: Cambiar permisos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Cambiar permisos”.	
2.		Muestra la interfaz cambiar permisos.
3.	Introduce los datos correspondientes.	
4.		Comprueba que los datos cumplan con los requisitos necesarios para que sea válido.

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

5.		Comprueba que exista el usuario en la base de datos.
6.		Cambia permisos del usuario.
<b>Flujos alternos</b>		
4. Los datos no son válidos.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Flujos alternos</b>		
5. Los datos no existen.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Flujo básico &lt;Nombre del flujo básico&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
4.	.	Termina el CU.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

### PATRÓN ARQUITECTÓNICO Y PATRONES DE DISEÑO

Los patrones son un amplio repertorio de principios generales basados en la experiencia que guían la creación de un software. Para que una solución sea considerada un patrón debe ser aplicable a diferentes problemas en distintas circunstancias. Los patrones de arquitectura expresan un esquema organizativo estructural fundamental para sistemas de software especificando subsistemas con sus responsabilidades, detallando un grupo de recomendaciones para organizar componentes. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software (45).

### Modelo Plantilla Vista (MTV)

Django, el framework de desarrollo seleccionado a utilizar del lado del servidor, respeta el paradigma conocido como MTV<sup>16</sup>; el cual está fuertemente inspirado en la filosofía de desarrollo MVC<sup>17</sup>, con ciertas diferencias. Por lo tanto, el controlador en un verdadero framework MVC, en Django se le llama vista; y lo que se llamaría vista, se llama plantilla (46). Para un mejor entendimiento se presenta el siguiente diagrama.

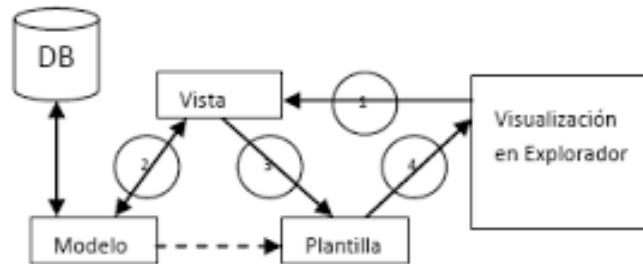


Imagen 3. Arquitectura del Sistema de Análisis Estadístico de Twitter (47).

El patrón MTV al igual que el MVC permite separar los datos, la lógica de la aplicación y la presentación gráfica con la finalidad de conseguir un código limpio y un fácil mantenimiento ya que las modificaciones de una parte de la aplicación no afectan a las demás.

### Patrones de diseño.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. La clave para la reutilización es anticiparse a los nuevos requisitos y cambios, de modo que los sistemas evolucionen de forma adecuada. Facilitan la reusabilidad, extensibilidad y mantenimiento (48).

Estos patrones pueden ser clasificados en:

- **Creacionales:** solucionan problemas de creación de instancias, ayudan a encapsular y abstraer dicha creación.

<sup>16</sup> MTV: Model Template View; Modelo Plantilla Vista

<sup>17</sup> MVC: Modelo Vista Controlador.

- **Estructurales:** solucionan problemas de composición (agregación) de clases y objetos.
- **De Comportamiento:** son soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan (48).

### Patrones de diseño GRASP<sup>18</sup>

Los GRASP son patrones generales de software para la asignación de responsabilidades. En el diseño orientado a objetos, a pesar de utilizar metodologías ágiles como OpenUP, es inevitable elegir cuidadosamente las responsabilidades de cada clase desde la primera codificación; así como, en la refactorización constante del programa. Los patrones que a continuación se mencionan son los utilizados en el Sistema.

- **Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que recibe los datos del usuario y los envía a las distintas clases según el método llamado (49). Este patrón se evidencia en el uso de la clase controladora.
- **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas 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 (49). Se evidencia en la estructura misma del *framework* Django.
- **Alta cohesión:** Se aplica en la mayoría de las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden (49).

### DIAGRAMAS DE CLASES

Los diagramas de clases del diseño, representan las relaciones entre clases que mantienen la información manipulada por el sistema, constituyen además, la interacción de las clases del diseño y sus objetos en la realización de los casos de uso, de forma que de cada uno se desprende un diagrama de clases (50). A

---

<sup>18</sup> GRASP: General Responsibility Assignment Software Patterns.

continuación, se muestra el diagrama de clases del diseño del caso de uso Realizar búsquedas simples, el resto de las descripciones se encuentran en el [Anexo # 2](#).

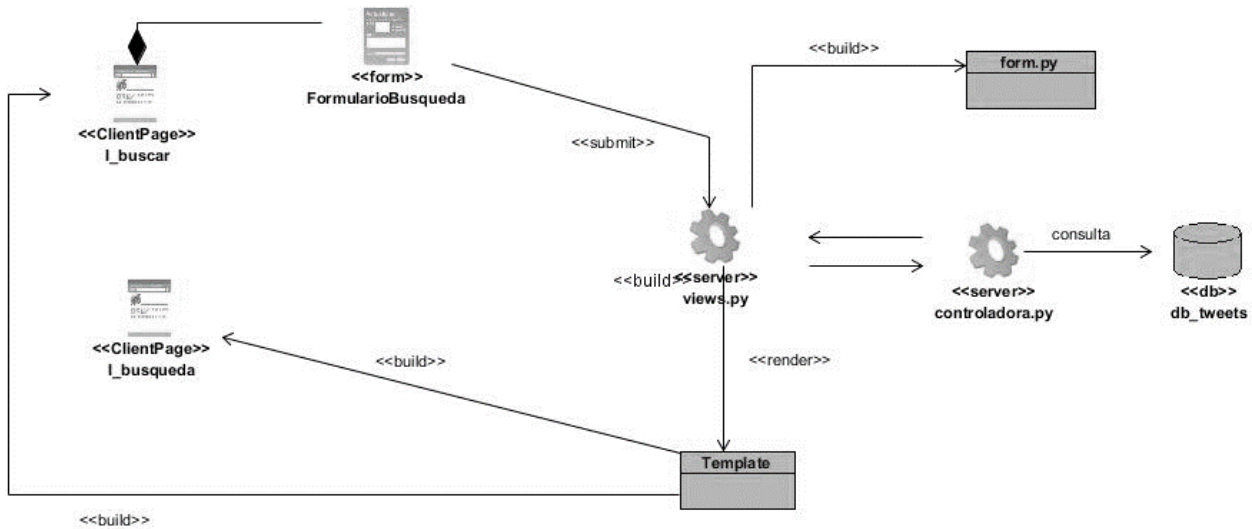
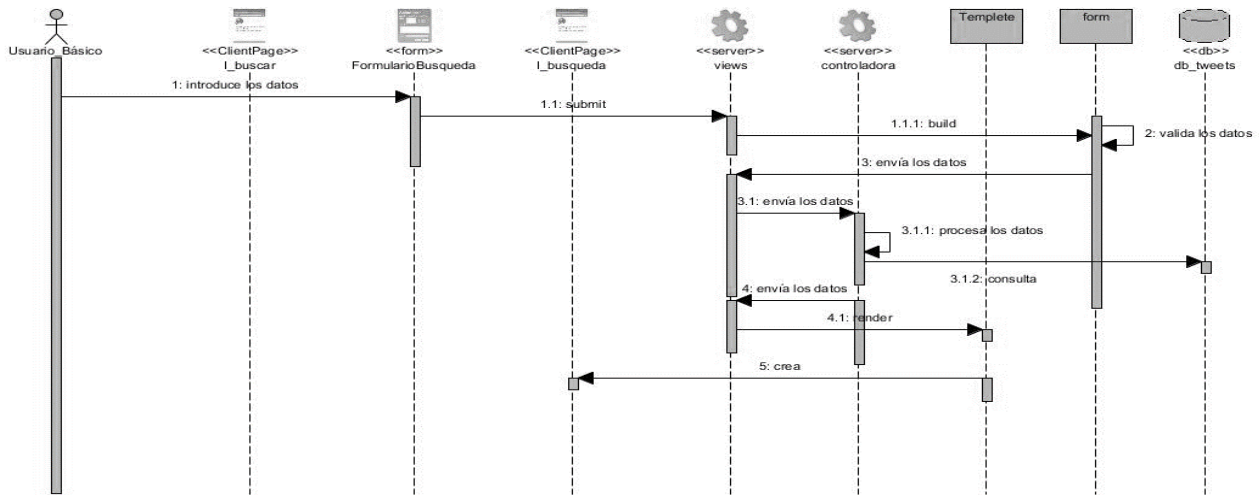


Imagen 4. Diagrama de clases de diseño del CU Realizar búsquedas simples.

### DIAGRAMAS DE INTERACCIÓN

Los diagramas de secuencia, describen las interacciones entre un grupo de objetos mostrando de forma secuencial los envíos de mensajes entre estos (51). A continuación, se presenta el diagrama de secuencia correspondiente al caso de uso Realizar búsqueda simple, el resto de las descripciones se encuentran en el [Anexo # 3](#).



**Imagen 5.** Diagrama de secuencia del CU Realizar búsquedas simples.

### DISEÑO DE LA BASE DE DATOS

Las bases de datos desempeñan un papel crucial en casi todas las áreas de desarrollo de software. Su diseño es una de las tareas más importantes en la construcción de un sistema que requiera manejar la permanencia de la información (52). El sistema que se propone hace uso de una base de datos (BD), ya que es necesario contar con una cantidad considerable de información relacionada los tuits.

MongoDB es un gestor de base de datos no relacional que se basa en colecciones. Cuando se dice que se basa en colecciones, significa que los datos se agrupan en conjuntos llamados "colecciones". Cada colección tiene un nombre único en la base de datos y puede contener un número ilimitado de documentos. Una colección es análoga a una tabla, excepto que no tienen un esquema definido (53). A continuación se muestra el modelo físico de la BD para el sistema propuesto.

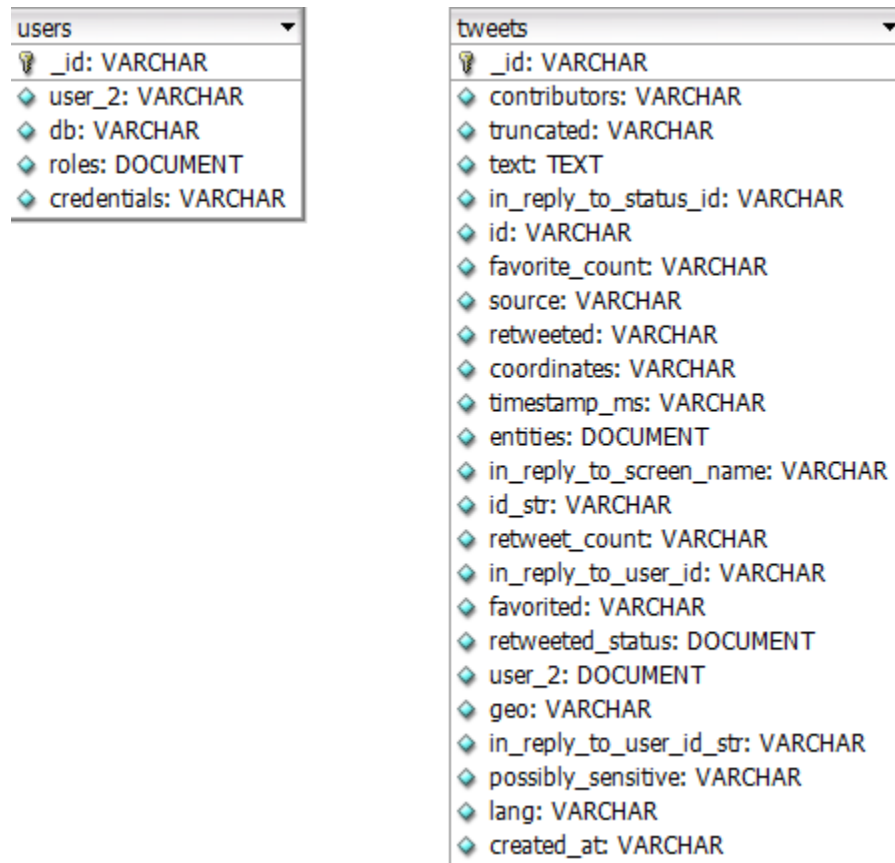


Imagen 6. Modelo Físico de Datos para Mongo DB.

### MODELO DE DESPLIEGUE

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos (51).

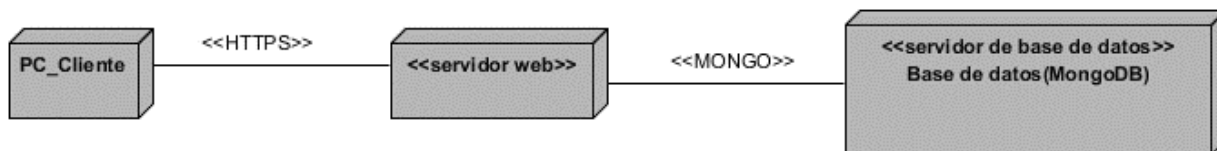


Imagen 7. Diagrama de despliegue.

La aplicación será desplegada de la siguiente manera: la PC\_Cliente a través del navegador utilizará el protocolo HTTPS<sup>19</sup> para la conexión con el servidor web que tendrá alojada la aplicación web Tuitzer. Esta aplicación es la encargada de realizar la recuperación y publicación de la información en la web, que a través del protocolo de comunicación Mongo Wire Protocol (MONGO) garantizará la persistencia de los datos en el servidor de base de datos con el gestor MongoDB ejecutándose.

### CONCLUSIONES DEL CAPÍTULO

En el desarrollo de este capítulo se definieron aspectos correspondientes al desarrollo de la solución propuesta. Se identificaron las características y responsabilidades con las que debe cumplir el sistema, lo que permitió la definición de los requerimientos funcionales y no funcionales del software. Además se identificaron los patrones de diseño; así como, el patrón de arquitectura utilizado. A través de la obtención de los artefactos generados por este flujo de trabajo se obtuvo una definición de los elementos que se tomarán en cuenta en la implementación del sistema.

---

<sup>19</sup> HTTPS: Hypertext Transfer Protocol Secure; Protocolo Seguro de Transferencia de Hipertexto



## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

A partir de los resultados obtenidos durante el diseño de la aplicación, se definen los estándares de codificación empleados. Se realizan pruebas para comprobar el correcto funcionamiento del software, se verifica que la entrada de datos se acepte de forma adecuada y que se produzca un resultado correcto.

### DIAGRAMA DE COMPONENTES

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes (54).

A continuación se muestra el diagrama de componentes.

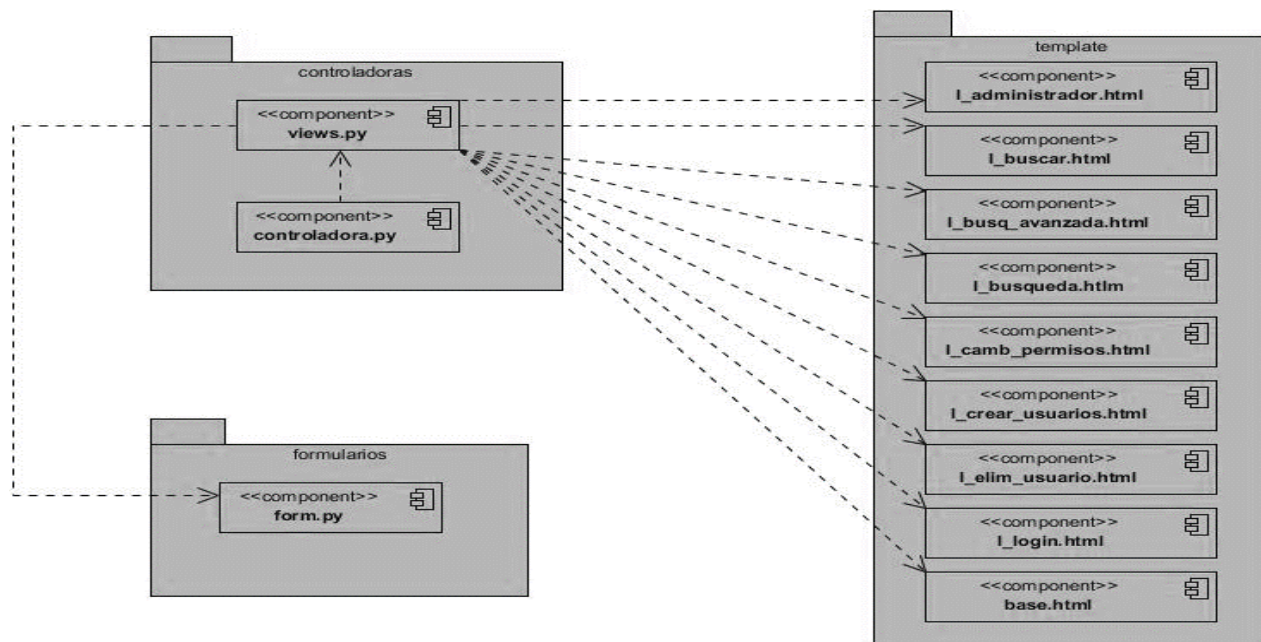


Imagen 8. Diagrama de componentes.

## IMPLEMENTACIÓN

La implementación es como desarrollar y organizar los componentes basándose en las especificaciones del diseño. Es la realización de las especificaciones técnicas o algoritmos como un programa, componente software, u otro sistema de cómputo. Dentro de los principales objetivos que se desea alcanzar en una etapa de implementación se encuentran (55):

- Definir la organización del código, en términos de los subsistemas de implementación organizados en capas.
- Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros).
- Probar y desarrollar componentes como unidades.
- Integrar los resultados producidos por los implementadores individuales, o equipos en un sistema ejecutable.

### Estándares de código

Los estilos de programación, también llamado estándares de código, es un término que describe convenciones para escribir código fuente en un cierto lenguaje de programación. El estilo de programación es dependiente del lenguaje de programación que se haya elegido (56).

A continuación se muestran los estándares utilizados.

**Tamaño máximo de línea:** limita todas las líneas a un máximo de 30 tabulaciones.

**Líneas en blanco:** se separan las funciones no anidadas y las definiciones de clases con dos líneas en blanco. Las definiciones de métodos dentro de una misma clase se separan con una línea en blanco.

### Imports

- Normalmente los *imports* deberían colocarse en distintas líneas, por ejemplo:

*Sí:*

*import os*

*import sys*

No:

```
import sys, os
```

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

- Cuando se importe una clase de un módulo, normalmente se hace.

```
from myclass import MyClass
```

```
from foo.bar.yourclass import YourClass
```

## Espacios en blanco en expresiones y sentencias

Se evita espacios en blanco extras en las siguientes situaciones:

- Inmediatamente después de entrar en un paréntesis o antes de salir de un paréntesis, corchete o llave.

Sí: *spam(ham[1], {eggs: 2})*

No: *spam( ham[ 1 ], { eggs: 2 } )*

- Inmediatamente antes de una coma, punto y coma, o dos puntos:

Sí: *if x == 4: print x, y; x, y = y, x*

No: *if x == 4 : print x , y ; x , y = y , x*

- Inmediatamente antes de abrir un paréntesis para una lista de argumentos de una llamada a una función:

Sí: *spam(1)*

No: *spam (1)*

- Inmediatamente antes de abrir un paréntesis usado como índice o para particionar (*slicing*):

Sí: *dict['key'] = list[index]*

No: *dict ['key'] = list [index]*

- Más de un espacio alrededor de un operador de asignación (u otro operador) para alinearlos con otro.

Sí:  $x = 1, y = 2$

No:  $x = 1, y = 2$

### Usa espacios alrededor de los operadores aritméticos:

Sí:

```
i = i + 1
```

```
submitted += 1
```

```
x = x * 2 - 1
```

```
hypot2 = x * x + y * y
```

```
c = (a + b) * (a - b)
```

No:

```
i=i+1
```

```
submitted +=1
```

```
x = x*2 - 1
```

```
hypot2 = x*x + y*y
```

```
c = (a+b) * (a-b)
```

**No usar espacios alrededor del signo '=' cuando se use para indicar el nombre de un argumento o el valor de un parámetro por defecto.**

Sí: `def complex(real, imag=0.0):`

```
return magic(r=real, i=imag)
```

No: `def complex(real, imag = 0.0):`

```
return magic(r = real, i = imag)
```

### Comentarios de bloque

Los comentarios de bloque generalmente se aplican al código. Cada línea de un comentario de bloque comienza con el símbolo de # y un único espacio. Los párrafos dentro de un comentario de bloque se separan por una línea conteniendo un único símbolo de #.

### Descriptivo: Estilos de Nombrado

- Minúsculas (para las funciones cuando su nombre es una sola palabra)

- minúsculas\_con\_guiones\_bajos (para las funciones: no anidadas o métodos de clases)
- PalabrasEnMayúsculas (para las clases)

### **No comparar valores booleanos con True o False usando ==**

*Sí: if greeting:*

*No: if greeting == True:*

## **PRUEBAS**

A las pruebas se les confiere un valor esencial dentro del desarrollo de aplicaciones ya que un fallo en estas puede representar costos elevados. Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Para la validación de un software de manera general se utilizan principalmente dos métodos de prueba: caja blanca y caja negra. La pruebas de caja blanca basan su funcionamiento en evaluar la información sobre cómo el software ha sido diseñado y codificado. El método caja negra actúa sobre la validación de los requisitos funcionales y se aplica a la interfaz del software para examinar algún aspecto funcional de un sistema, por lo que también se les denominan pruebas funcionales (57).

### **Pruebas de caja negra**

Las pruebas de caja negra intentan encontrar errores de las siguientes categorías (58):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento y errores de inicialización y de terminación.

Esta técnica se basa en recoger todas las combinaciones de escenarios posibles a ejecutar, contienen la descripción de todas las variables o datos de entrada al sistema. Son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente. (57)

Se decide utilizar en la realización de las pruebas funcionales el método de caja negra con la aplicación del criterio partición equivalente. Este tipo de pruebas se aplicaron con el objetivo de localizar fallas funcionales en el sistema, al identificar situaciones en las que las respuestas de este a determinadas acciones del usuario no se apegan a las especificaciones establecidas. Se ejecutaron pruebas a todos los casos de uso con diferentes entradas del usuario para determinar que los resultados obtenidos fueran consistentes bajo cualquier situación, lo que permitió comprobar el cumplimiento de cada uno de ellos y el cumplimiento de los requisitos establecidos.

**Objetivos:** El objetivo de realizar este tipo de prueba al sistema es mostrar el incorrecto o incompleto funcionamiento de este, así como los errores de interfaz, rendimiento y errores de inicialización y terminación.

**Alcance:** El proceso de las pruebas de caja negra se centra principalmente en los requisitos funcionales del software para verificar el comportamiento de la unidad observable externamente y la calidad funcional.

**Descripción:** Se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

### Casos de pruebas

A continuación se presenta una muestra del diseño de casos de pruebas realizados (ver Tabla 7 y 8), el resto de las descripciones se encuentran en el [Anexo 4](#).

**Tabla 7.** Descripción de prueba para el CU Autenticar usuario.

Entrada	Resultado	Condiciones
---------	-----------	-------------

El usuario introduce los datos ( <i>usuario, contraseña</i> ) correctos para autenticarse.	El sistema reenvía al usuario a la interfaz predeterminada para su rol.	Que la BD esté conectada.
El usuario omite algunos de los datos ( <i>usuario, contraseña</i> ) necesarios para autenticarse en el sistema.	El sistema muestra un mensaje de error informándole al usuario los campos requeridos y la especificación para cada campo.	Que la BD esté conectada.
El usuario introduce algunos datos ( <i>usuario, contraseña</i> ) incorrectos para autenticarse.	El sistema muestra un mensaje de error informándole al usuario que alguno de los datos está incorrecto o que la conexión con la BD no se ha establecido.	Que la BD esté conectada.

**Tabla 8.** Descripción de prueba para el CU Realizar búsquedas simples.

<b>Entrada</b>	<b>Resultado</b>	<b>Condiciones</b>
El usuario introduce el criterio de búsqueda ( <i>palabra, etiqueta, usuario</i> ) correctos.	El sistema muestra la interfaz de búsqueda con los datos correspondiente a la búsqueda realizada.	Que la BD esté conectada.
El usuario introduce el criterio de búsqueda ( <i>palabra, etiqueta, usuario</i> ) incorrectos.	El sistema muestra un mensaje de error informándole al usuario que los datos son incorrectos.	Que la BD esté conectada.
El usuario deja el campo de búsqueda vacío.	El sistema le muestra un mensaje informando al usuario que el campo no puede estar vacío.	Que la BD esté conectada.

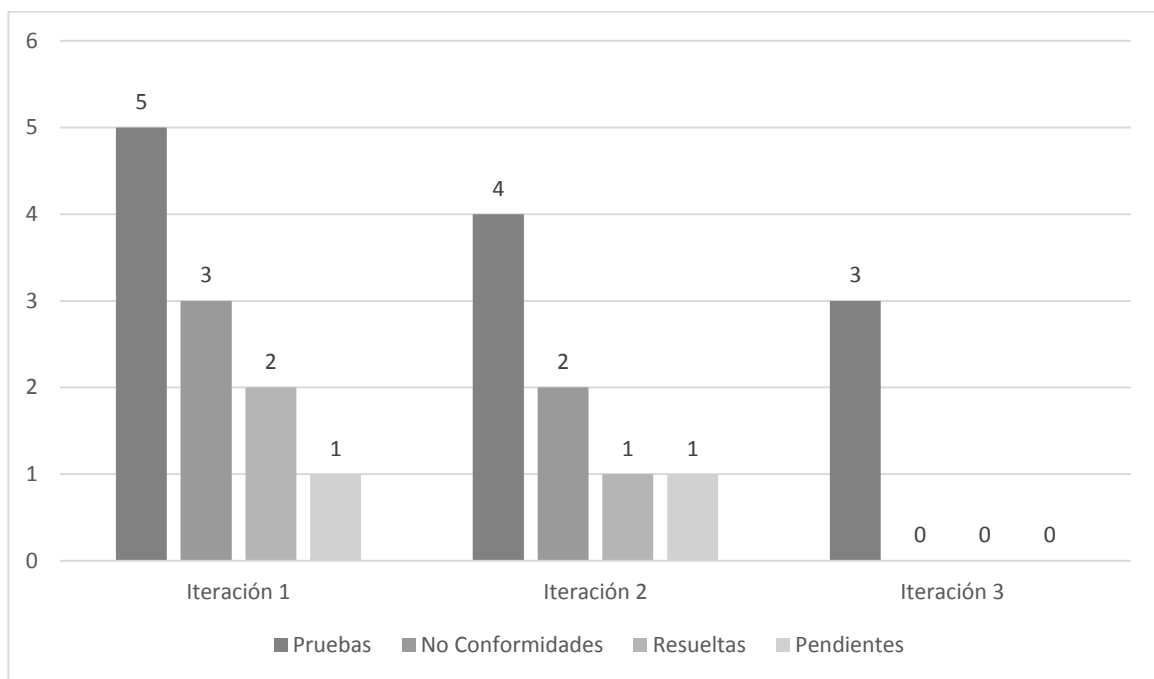
### **Resultado de las pruebas**

Para verificar el correcto funcionamiento de la interfaz del software se realiza la prueba de caja negra, la cual permite identificar las no conformidades encontradas en el sistema. En la primera iteración, se obtuvo como resultado del análisis de los escenarios de pruebas, cinco no conformidades correspondientes a las funcionalidades implementadas hasta el momento de ejecución de las pruebas. Las no conformidades

fueron corregidas posteriormente, dándole paso a la segunda iteración, en las cuales se comprobaron nuevamente los escenarios de pruebas, donde se obtuvieron dos no conformidades.

En una tercera iteración fueron solucionadas las no conformidades pendientes de la iteración anterior y se verificaron nuevamente los requisitos correspondientes a todos los CU del sistema. Se validó en sentido general el correcto funcionamiento de los requisitos funcionales del software.

A continuación se muestra una gráfica con los resultados por cada iteración.



**Imagen 9.** Resultado de las pruebas funcionales.

### **Pruebas de caja blanca**

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.



- Se ejecutan todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas (59).

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo (54). Uno de estos niveles de pruebas son las pruebas unitarias.

**Pruebas unitarias:** Están enfocadas a los elementos *testables* más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca (60).

### Pruebas de caja blanca realizadas al sistema

A continuación se muestran las pruebas que se realizaron a los métodos de la clase controladora *controladora.py*, el resto de las pruebas se encuentran en la clase *test\_controladora.py*.

```

def test_buscar_en_tweets(self):
    self.assertRaises(TypeError, controladora.buscar_en_tweets, "Cuba", "", None, "", "")
    self.assertRaises(TypeError, controladora.buscar_en_tweets, None, "")
    self.assertRaises(TypeError, controladora.buscar_en_tweets)

# función de prueba para el método: conexion
def test_conexion(self):
    self.assertRaises(errors.InvalidName, controladora.conexion, "")
    self.assertRaises(TypeError, controladora.conexion)
    self.assertRaises(TypeError, controladora.conexion, None)

# función de prueba para el método: distribuidor de busqueda
def test_manipulador_de_busqueda(self):
    self.assertRaises(TypeError, controladora.manipulador_de_busqueda, "venezuela", None)

# función de prueba para el método: crear modificar usuario
def test_crear_modificar_usuario(self):
    self.assertRaises(TypeError, controladora.crear_modificar_usuario, "buscador", "estepassword", "modificar")

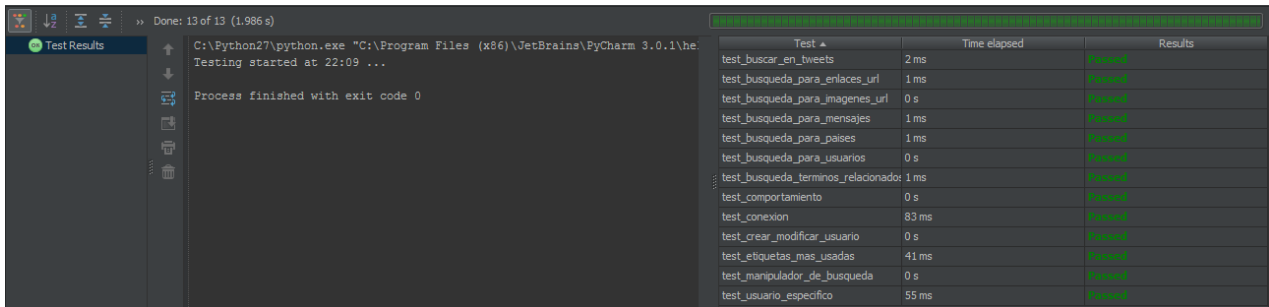
# función de prueba para el método: busqueda para usuarios
def test_busqueda_para_usuarios(self):
    self.assertRaises(TypeError, controladora.busqueda_para_usuarios, "", "", None)

# función de prueba para el método: busqueda para mensajes
def test_busqueda_para_mensajes(self):
    self.assertRaises(TypeError, controladora.busqueda_para_mensajes, "", "", None)

# función de prueba para el método: busqueda para enlaces url
def test_busqueda_para_enlaces_url(self):
    self.assertRaises(TypeError, controladora.busqueda_para_enlaces_url, "", "", None)

```

**Imagen 10.** Ejecución de los métodos de la clase controladora *controladora.py*.



Test	Time elapsed	Results
test_buscar_en_tweets	2 ms	Passed
test_búsqueda_para_enlaces_url	1 ms	Passed
test_búsqueda_para_imágenes_url	0 s	Passed
test_búsqueda_para_mensajes	1 ms	Passed
test_búsqueda_para_países	1 ms	Passed
test_búsqueda_para_usuarios	0 s	Passed
test_búsqueda_terminos_relacionados	1 ms	Passed
test_comportamiento	0 s	Passed
test_conexion	83 ms	Passed
test_crear_modificar_usuario	0 s	Passed
test_etiquetas_mas_usadas	41 ms	Passed
test_manipulador_de_búsqueda	0 s	Passed
test_usuario_especifico	55 ms	Passed

**Imagen 11.** Resultado de la prueba de los métodos de la clase controladora *controladora.py*.

## Pruebas de carga y estrés

Este tipo de prueba se enfoca en evaluar el comportamiento del sistema bajo condiciones anormales. Estrés del sistema se refiere a extrema carga, memoria insuficiente, no disponibilidad de servicios y hardware o recursos compartidos limitados. Este tipo de prueba permite comprender mejor cómo y qué áreas del sistema colapsarán, de este modo es posible planificar contingencias y actualizar el mantenimiento y planear y asignar recursos de antemano. Valida y evalúa la aceptabilidad de un elemento de un sistema sobre diferentes cargas de trabajo mientras el sistema permanece constante. Generalmente se incluye simulación de cargas de trabajo promedio y pico que puedan ocurrir dentro de la tolerancia operacional normal (61).

Para probar la escalabilidad del sistema es necesario conocer acerca de la carga que puede soportar. Para ello se realizaron las pruebas de rendimiento de software, las cuales se apoyan en el uso de la herramienta JMeter, que ofrece datos importantes acerca del estrés que puede soportar el sistema. Se probó el sistema con un total de 2750 peticiones enviadas en 50 hilos concurrentes, arrojando un 0,0% de error al atender dichas peticiones, con un rendimiento de 30,6/sec.

## CONCLUSIONES DEL CAPÍTULO

Se detalló el estándar de codificación utilizado, logrando una mayor legibilidad y limpieza del código. Además, se construyó el sistema de acuerdo a lo definido durante el diseño de cada componente. Se logró desarrollar las pruebas para comprobar el curso fundamental de las funcionalidades y la carga que puede soportar la aplicación.

## CONCLUSIONES GENERALES

La realización del presente trabajo ha evidenciado el cumplimiento de los objetivos propuestos. En general, se ha arribado a las siguientes conclusiones:

- A partir de la investigación realizada a los sistemas homólogos, se concluye que aunque estos no constituyeron una solución al problema identificado, representaron un punto de partida y sentaron las bases para el desarrollo del presente trabajo.
- Se logró desarrollar una aplicación web, encaminada a brindar una solución viable al problema identificado; la cual permite realizar reportes estadísticos automatizados sobre los datos recolectados de la red social Twitter, basándose en las necesidades del Departamento DOWAI.
- Las pruebas aplicadas permitieron validar que el sistema desarrollado cumple con los requerimientos de software que fueron definidos.

## RECOMENDACIONES

Con el objetivo de mejorar y seguir perfeccionando el funcionamiento de la aplicación desarrollada como parte de esta investigación, se recomienda:

- Continuar con el proceso de desarrollo del sistema con el fin de incorporarle nuevas funcionalidades, de acuerdo a posibles cambios de las necesidades en el Departamento DOWAI.
- Integrar la aplicación con otras aplicaciones que realicen reportes sobre datos recolectados de la red social Twitter; por ejemplo, una aplicación de creación de perfiles de usuario.

## REFERENCIAS BIBLIOGRÁFICAS

1. Cronin, Blaise. *Information management: from strategies to action*. London : Aslib, 1985.
2. About Twitter, Inc. About. [Online] abril 30, 2015. <https://about.twitter.com/company>.
3. Developers, Twitter. Documentation. [Online] Noviembre 20, 2015. <https://dev.twitter.com/overview/documentation>.
4. Anidjar, Silvina, et al. *Las Redes Sociales*. Argentina : Universidad Nacional de San Luis, 2009.
5. Cabrera, María Ángeles. *Evolución tecnológica y cibermedios*. s.l. : Comunicación Social, 2010. pp. 112-128.
6. Gallego, J. C. *Tecnologías de la Información y de la Comunicación. Técnicas básicas*. Madrid : Editex, 2010.
7. Sutorius, Brian. *Twitter Developers*. [Online] [Cited: abril 29, 2015.] <https://dev.twitter.com/overview/api/tweets>.
8. FELIX, CARLOS. Metodologías ágiles. [Online] [Cited: febrero 6, 2015.] <http://metodologiasagiles.wikispaces.com/metodos+agiles+vs+metodos+tradicionales>.
9. Figueroa, Roberth G., Solís, Camilo J. and Cabrera, Armando A. *Metodologías Tradicionales vs. Metodologías Ágiles*. 2010.
10. Peñalver, G, Meneses, A and García, S. *Metodología Ágil para el desarrollo de Software*. 2010.
11. OpenUP como alternativa metodológica para proyectos pequeños de software . [Online] [Cited: noviembre 4, 2014.] <http://kasyles.blogspot.com/2008/09/openup-como-alternativa-metodolgica.html>.
12. Schmuller, Joseph. *Aprendiendo UML en 24 horas*. México : s.n., 2010.
13. Menéndez, Evelyn. Plusformación. [Online] [Cited: noviembre 15, 2014.] [http://www.plusformacion.com/Recursos/r/Herramientas-CASE-ara-procesodesarrollo-Software?quicktabs\\_ofertas\\_relacionadas\\_quicktab=1#resumena](http://www.plusformacion.com/Recursos/r/Herramientas-CASE-ara-procesodesarrollo-Software?quicktabs_ofertas_relacionadas_quicktab=1#resumena).
14. Visual Paradigm . [Online] [Cited: noviembre 27, 2014.] <http://www.visual-paradigm.com/>.
15. IBM Rational Rose Enterprise. [Online] [Cited: noviembre 27, 2014.] <http://www-03.ibm.com/software/products/es/enterprise>.
16. Enciclopedia Libre Universal en Español. [Online] [Cited: noviembre 2014, 4.] [http://enciclopedia.us.es/index.php/Lenguaje\\_de\\_programaci%C3%B3n](http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n).
17. González, Raúl. *Python para todos*. España : s.n., 2010.

18. Dayley, Brad. *Python Phrasebook*. 2007 : s.n.
19. FLANAGAN, David. JavaScript: the definitive guide. [Online] 2006.
20. Pérez, Javier Aguiluz. *Introducción a JavaScript*. . 2009.
21. MACDONALD, Matthew. *HTML5: The Missing Manual, 2nd Edition*. 2013.
22. Álvarez, Miguel Ángel. Manual de Canvas del HTML5. Desarrolloweb. [Online] [Cited: octubre 27, 2014.] <http://www.desarrolloweb.com>.
23. Lazaro, Juliana Monteiro. Qué es CSS. [Online] [Cited: diciembre 4, 2014.] <http://www.desarrolloweb.com/articulos/26.php>.
24. Highcharts librería para creación de gráficos. [Online] [Cited: mayo 4, 2015.] <https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/>.
25. Baranovskiy, Dmitry. Raphaël—JavaScript Library. [Online] [Cited: diciembre 1 , 2014.] <http://raphaeljs.com/>.
26. Watts, Gareth. jQuery Sparklines. [Online] [Cited: noviembre 26 , 2014.] <http://omnipotent.net/jquery.sparkline>.
27. Gutiérrez, Javier J. *¿Qué es un framework web?*
28. Tutorial de Django. [Online] [Cited: noviembre 20 , 2014.] <http://davidasorey.net/static/django-tutorial/>.
29. web2py Web Framework . [Online] [Cited: marzo 17, 2015.] [www.web2py.com](http://www.web2py.com).
30. PyCharm: El mejor IDE para tus proyectos en Python. [Online] [Cited: marzo 13, 2015.] <http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para-tus-proyectos-en-python-c114084/>.
31. Date, C. J. Introducción a los Sistemas de Bases de Datos. [Online] [Cited: abril 15, 2015.] <https://unefazuliasistemas.files.wordpress.com/2011/04/introduccion-a-los-sistemas-de-bases-de-datos-cj-date.pdf>.
32. Yunta, Luis. BASES DE DATOS DOCUMENTALES: ESTRUCTURA Y PRINCIPIOS DE USO. [Online] [Cited: abril 19, 2015.] <http://www.unav.es/dpp/documentacion/proteger/lryunta.pdf>.
33. JSON and BSON. . [Online] [Cited: marzo 14, 2015.] <http://www.mongodb.com/json-and-bson>.
34. Una introducción a MongoDB. [Online] [Cited: marzo 9, 2015.] <http://www.genbetadev.com/bases-de-datos/una-introduccion-a-mongodb>.
35. Apache Hbase. [Online] [Cited: marzo 16, 2015.] <http://hbase.apache.org/>.

36. López, MSc. Dailyn Sosa. SISTEMA PARA EL CONTROL DEL USO DE LOS SOFTWARES EDUCATIVOS. [Online] [Cited: abril 21, 2015.] <http://www.eumed.net/libros-gratis/2009c/585/Descripcion%20del%20modelo%20del%20dominio.htm>.
37. Jakarta. [Online] [Cited: mayo 18, 2015.] <http://jakarta.apache.org/jmeter/>.
38. QALoad - Automated Testing Institute . [Online] [Cited: mayo 16, 2015.] [http://www.automatedtestinginstitute.com/home/index.php?option=com\\_content&view=article&catid=51:tescat&id=1043:qaload](http://www.automatedtestinginstitute.com/home/index.php?option=com_content&view=article&catid=51:tescat&id=1043:qaload).
39. López, MSc. Dailyn Sosa. SISTEMA PARA EL CONTROL DEL USO DE LOS SOFTWARES EDUCATIVOS. [Online] [Cited: febrero 12, 2015.] <http://www.eumed.net/libros-gratis/2009c/585/Descripcion%20del%20modelo%20del%20dominio.htm>.
40. Requerimientos Funcionales y No Funcionales (RF/RNF). [Online] [Cited: noviembre 5, 2014.] <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>.
41. Casos de Uso (Use Case). [Online] [Cited: noviembre 5, 2014.] <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>.
42. Pressman, Roger S. *Ingeniería del software. Un enfoque práctico*. La Habana : Félix Varela, 2005.
43. FOWLER, Martin. *Analysis patterns: reusable object models*. Addison-Wesley Professional. 1997.
44. Gunnar Övergaard, Karin Palmkvist. *Use Cases: Patterns and Blueprints*. 2004.
45. Patrones y Antipatrones: una Introducción - Parte I . [Online] [Cited: enero 20, 2015.] <https://msdn.microsoft.com/es-es/library/bb972242.aspx#M2>.
46. Adrian Holovaty, Jacob Kaplan-Moss. *The Definitive Guide to django*. [Online] 2010.
47. *Python - DjangoFramework de desarrollo web para perfeccionistas* Basado en el Modelo MTV. Ayala, José Luis Condori.
48. Patrones de diseño: qué son y por qué debes usarlos. [Online] [Cited: enero 26, 2015.] <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>.
49. Patrones para asignar responsabilidades. grasp. [Online] [Cited: enero 2015, 19.] <http://es.slideshare.net/jpbthames/patrones-para-asignar-responsabilidades-grasp>.
50. Diseño UML . [Online] [Cited: enero 20, 2015.] <http://egdamar877.blogspot.com/2009/05/expocicion.html>.

51. Elementos de UML . [Online] [Cited: enero 20, 2015.] Elementos de UML <https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-elements.html>.
52. Diseño de datos . [Online] [Cited: enero 21, 2015.] <https://msdn.microsoft.com/es-es/library/aa290752%28v=vs.71%29.aspx> .
53. MongoDB: qué es, cómo funciona y cuándo podemos usarlo (o no). [Online] [Cited: enero 2015, 23.] <http://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>.
54. Vilas, Ana Fernandez. Introducción a UML . [Online] [Cited: mayo 5, 2015.] <http://www.gris.det.uvigo.es/~avilas/UML/UML.html>.
55. ¿Qué es la programación de software? [Online] [Cited: marzo 14, 2015.] <http://tech.batanga.com/2007/03/18/%C2%BFque-es-la-programacion-de-software>.
56. Guía de estilo para el código Python - Recursos Python . [Online] [Cited: marzo 21, 2015.] <http://recursospython.com/pep8es.pdf>.
57. Pruebas de Programas. [Online] [Cited: abril 2 , 2015.] [http://eva.uci.cu/file.php/158/Documentos/Recursos\\_bibliograficos/Libros\\_y\\_articulos\\_UD\\_2/Comun/Pruebas\\_de\\_Programas.pdf](http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_2/Comun/Pruebas_de_Programas.pdf) .
58. Martin, J. y Odell, J. *Object-Oriented Methods: A Foundation* Englewood Cliffs. Prentice P-Hall. 1995.
59. Técnicas de prueba. [Online] [Cited: mayo 4, 2015.] <http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
60. PRUEBAS UNITARIAS . [Online] [Cited: mayo 05, 2015.] [http://www.calidadyssoftware.com/testing/pruebas\\_unitarias3.php](http://www.calidadyssoftware.com/testing/pruebas_unitarias3.php).
61. Pruebas de estrés, carga y rendimiento. [Online] [Cited: marzo 5, 2015.] <http://juanbevilacqua.blogspot.com/2010/03/pruebas-de-estres-carga-y-rendimiento.html>.



# ANEXOS

## ANEXO 1: Descripción de los CU

**Tabla 9.** Descripción del CU5 Realizar búsquedas simples.

<b>Objetivo</b>	Realizar búsquedas simples	
<b>Actores</b>	Usuario básico	
<b>Resumen</b>	El CU se inicia cuando un usuario introduce un parámetro de búsqueda, el CU termina cuando se muestra el resultado de la búsqueda al usuario.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Media	
<b>Precondiciones</b>	Debe ser un usuario básico que esté operando en el sistema.	
<b>Postcondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Realizar búsquedas simples&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Introduce una búsqueda.	
2.		Muestra el resultado de la búsqueda en una nueva interfaz.
3.		Termina el CU.
<b>Flujos alternos</b>		
1. Los datos no son válidos.		
	<b>Actor</b>	<b>Sistema</b>
3.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		

<b>Asuntos pendientes</b>	
---------------------------	--

**Tabla 10.** Descripción del CU5 Realizar búsquedas avanzadas.

<b>Objetivo</b>	Realizar búsquedas avanzadas	
<b>Actores</b>	Usuario básico	
<b>Resumen</b>	El CU se inicia cuando un usuario introduce los parámetros de búsqueda, el CU termina cuando se muestra el resultado de la búsqueda al usuario.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Media	
<b>Precondiciones</b>	Debe ser un usuario básico que esté operando en el sistema.	
<b>Postcondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Realizar búsquedas avanzadas &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Introduce los parámetros de búsqueda.	
2.		Muestra el resultado de la búsqueda en una nueva interfaz.
3.		Termina el CU.
<b>Flujos alternos</b>		
1. Algunos de los datos no son válidos.		
	<b>Actor</b>	<b>Sistema</b>
4.		Muestra un mensaje de error donde se le comunica al usuario las causas del mismo.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

ANEXO 2: Diagramas de clase de diseño

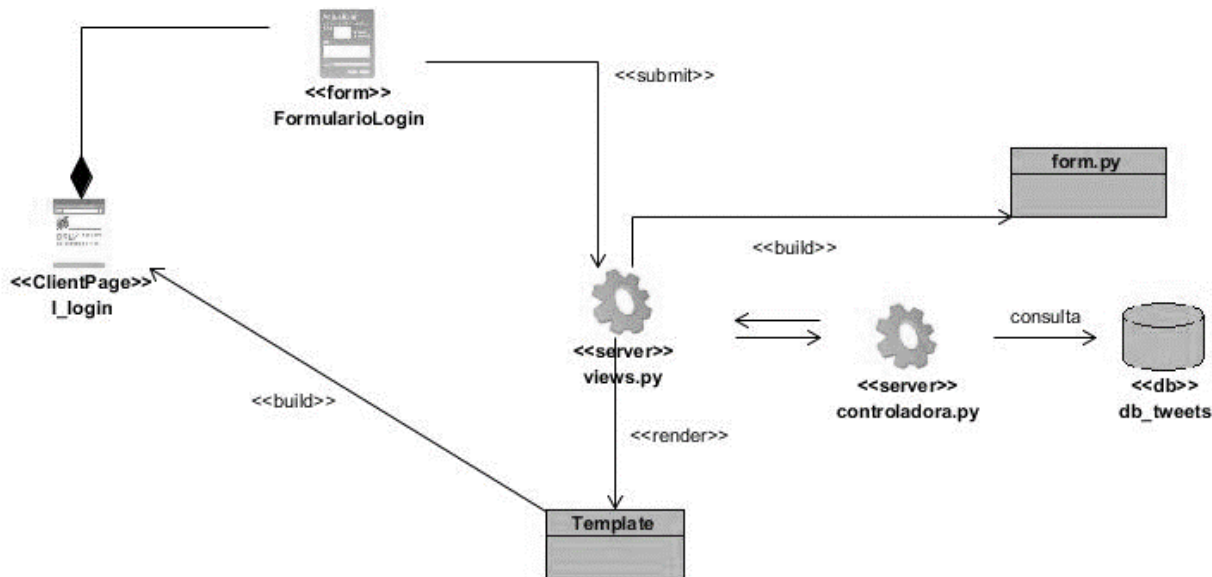


Imagen 12. Diagrama de clase de diseño del CU Autenticar usuario.

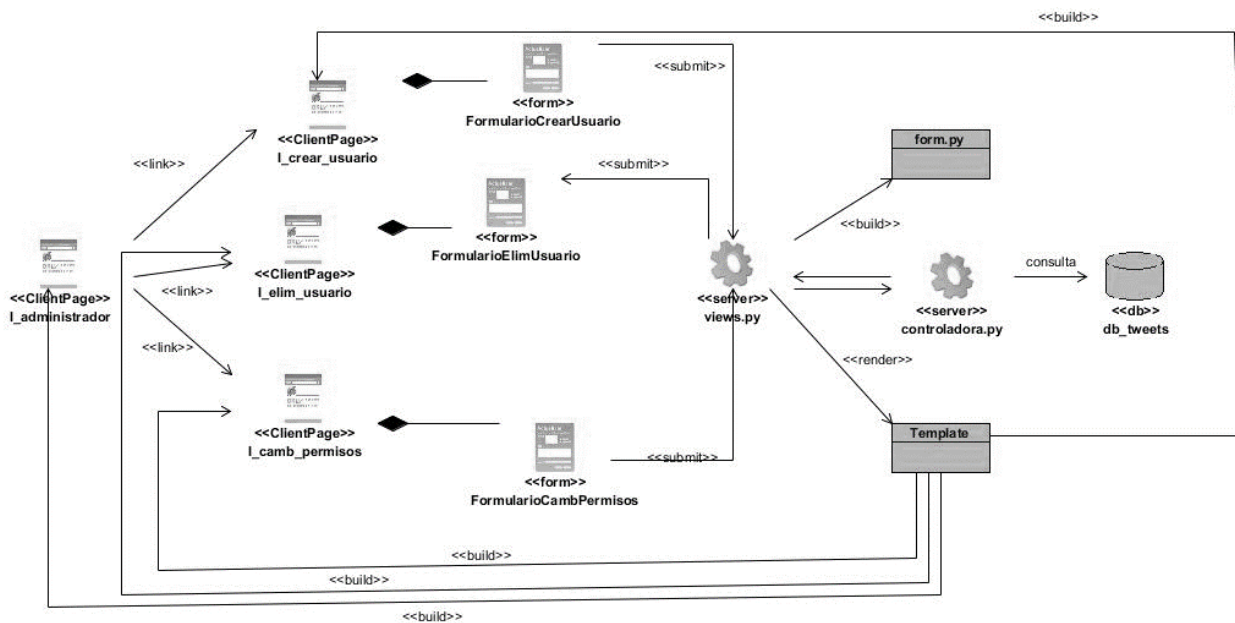


Imagen 13. Diagrama de clase de diseño del CU Gestionar usuario.

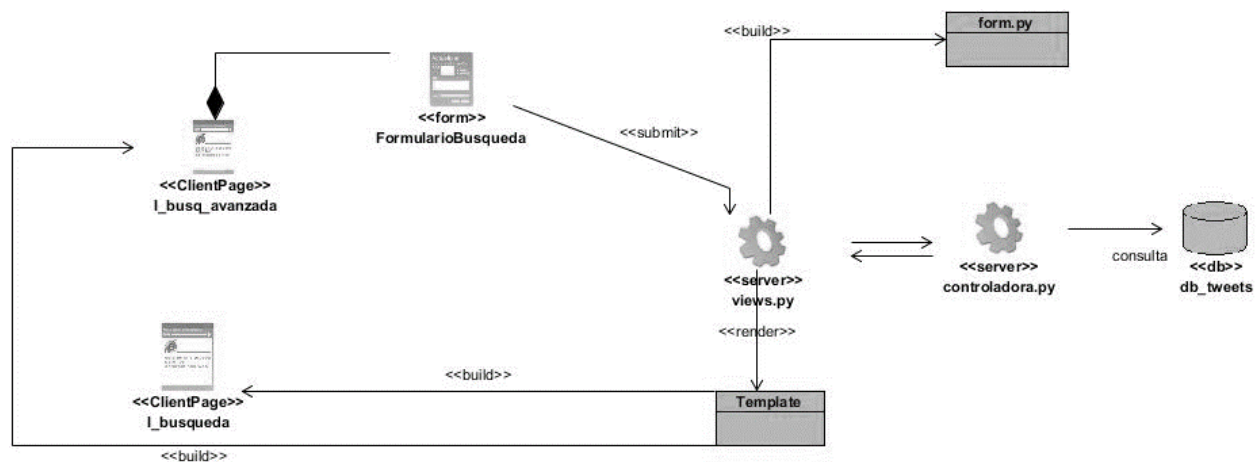


Imagen 14. Diagrama de clase de diseño del CU Realizar búsquedas avanzadas.

### ANEXO 3: Diagramas de interacción del diseño

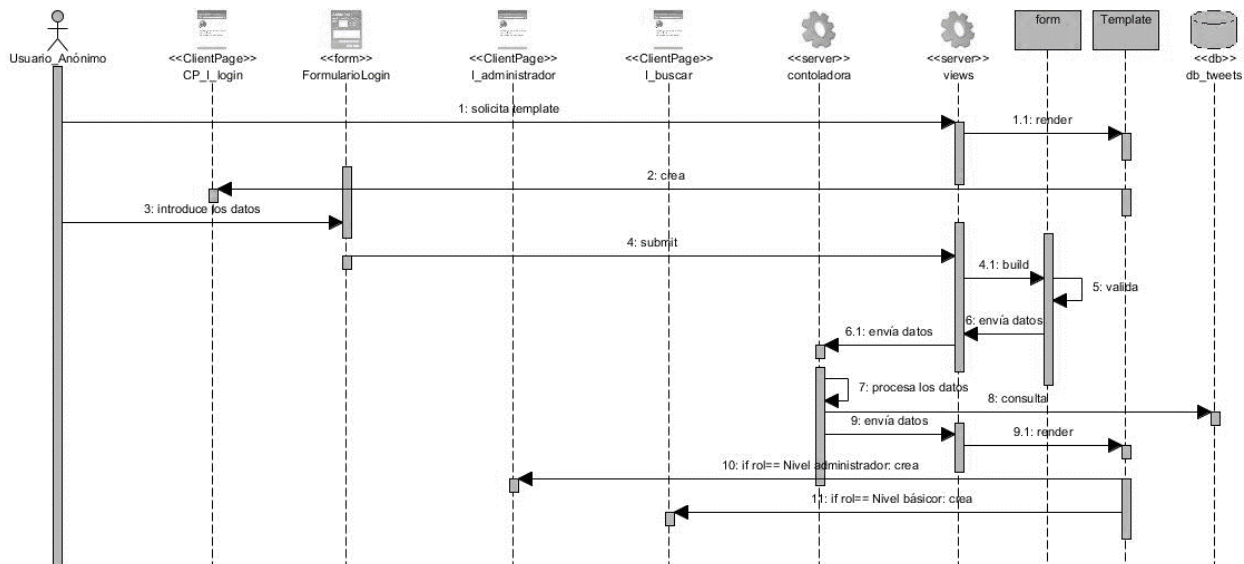


Imagen 15. Diagrama de interacción del diseño del CU Autenticar usuario.

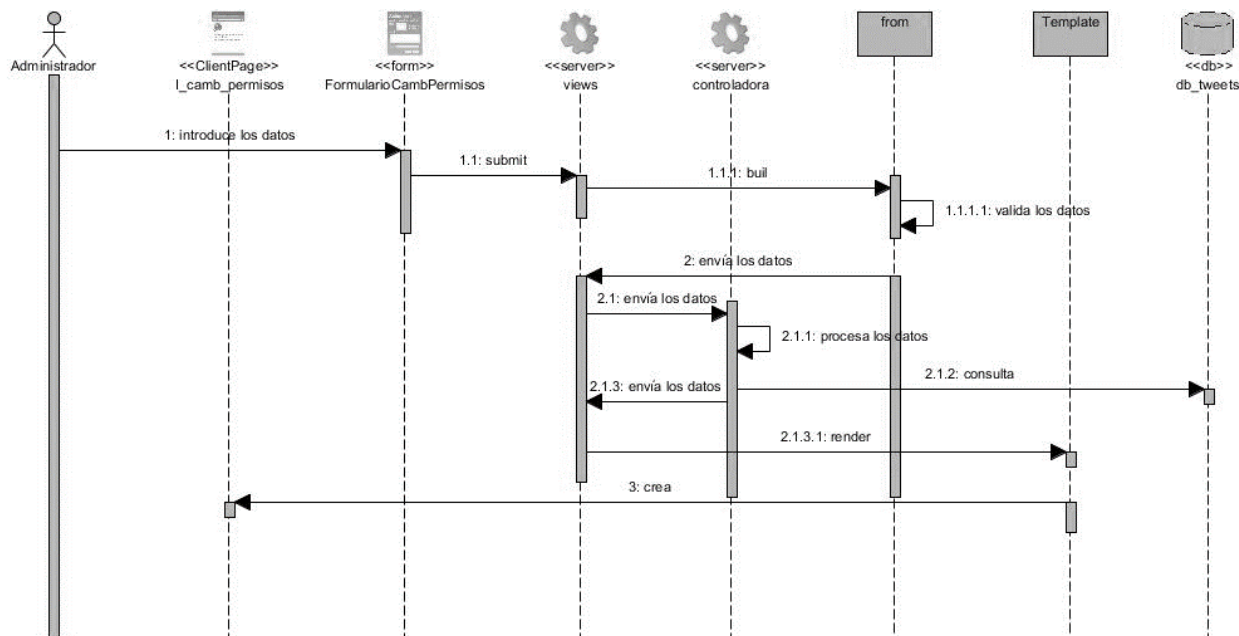


Imagen 16. Diagrama de interacción del diseño del CU Gestionar usuario: Cambiar permisos de usuarios.

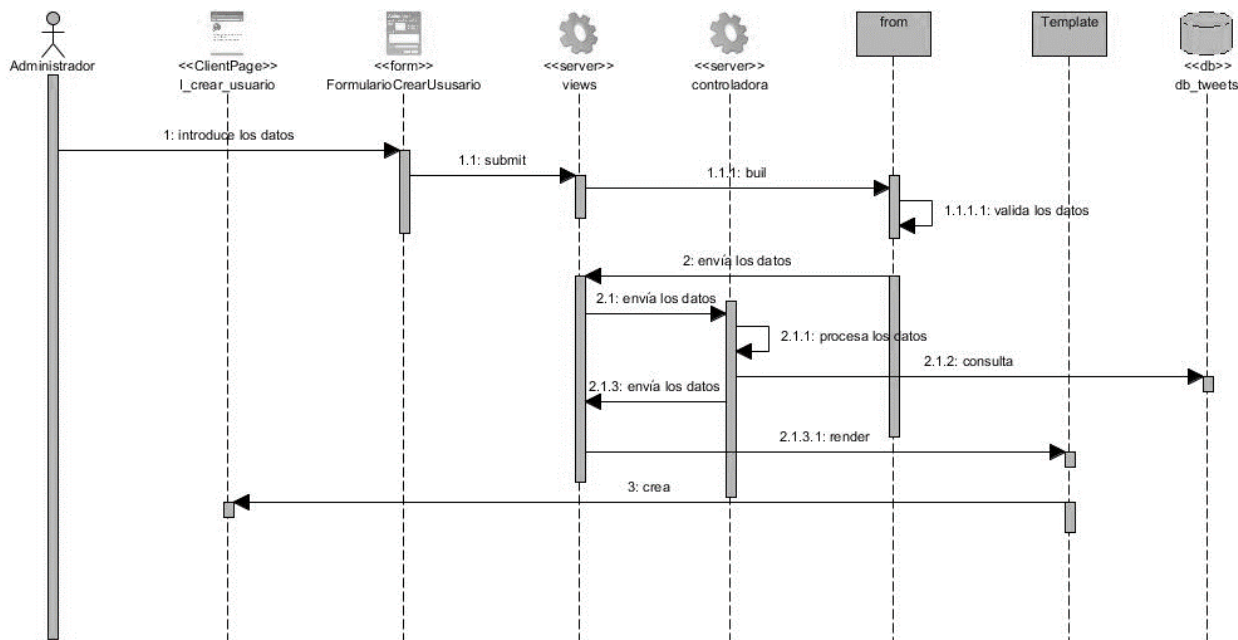


Imagen 17. Diagrama de interacción del diseño del CU Gestionar usuario: Crear usuario.

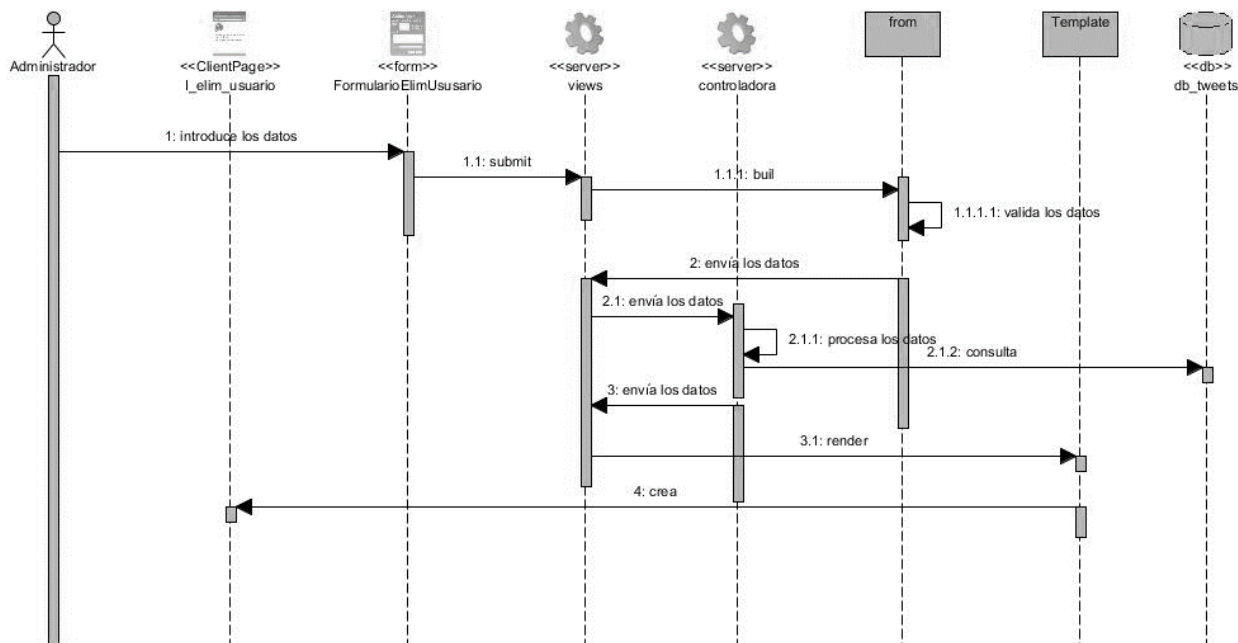


Imagen 18. Diagrama de interacción del diseño del CU Gestionar usuario: Eliminar usuario.

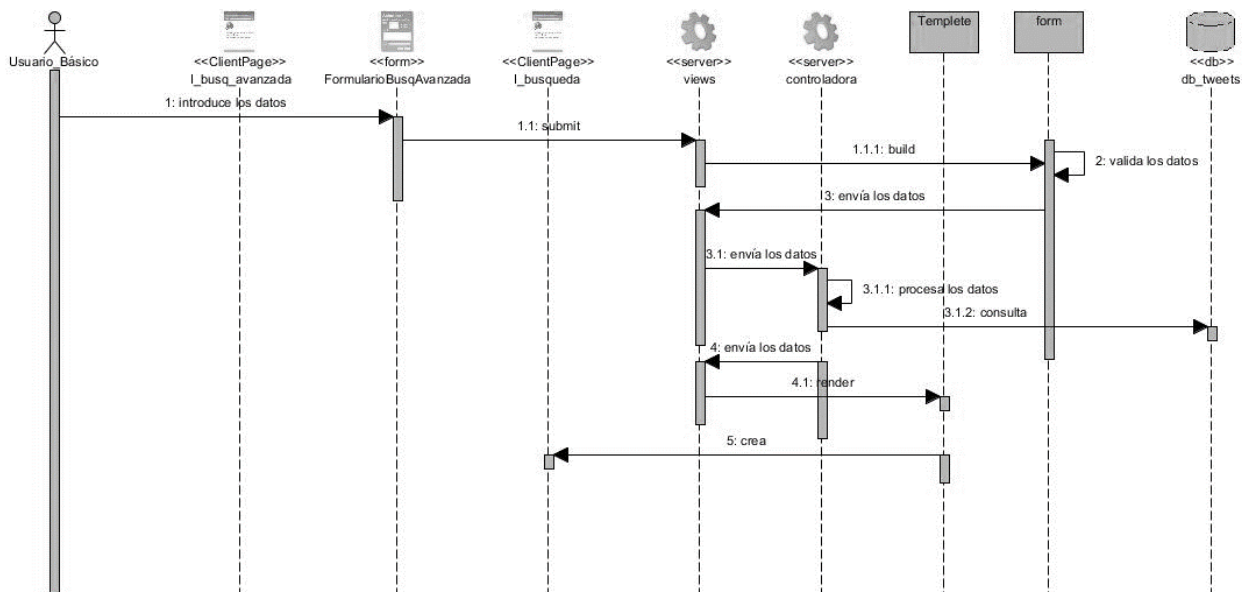


Imagen 19. Diagrama de interacción del diseño del CU Realizar búsquedas avanzadas.

## ANEXO 4: Casos de pruebas

**Tabla 11.** Descripción de prueba para el CU Realizar búsquedas avanzadas.

Entrada	Resultado	Condiciones
El usuario introduce los criterio de búsqueda ( <i>contiene, no contiene y fecha</i> ) correctos.	El sistema muestra la interfaz de búsqueda con los datos correspondiente a la búsqueda realizada.	Que la BD esté conectada.
El usuario introduce los criterio de búsqueda ( <i>contiene, no contiene y fecha</i> ) incorrectos.	El sistema muestra un mensaje de error informándole al usuario que los datos son incorrectos.	Que la BD esté conectada.
El usuario deja alguno de los campos de búsqueda vacío.	El sistema le muestra un mensaje informando al usuario que el campo no puede estar vacío.	Que la BD esté conectada.

**Tabla 12.** Descripción de prueba para el CU Gestionar usuario (Crear usuario).

Entrada	Resultado	Condiciones
El usuario administrador introduce los datos ( <i>usuario, contraseña y rol</i> ) correctos.	El sistema muestra la interfaz de crear usuario.	Que la BD esté conectada.
El usuario administrador introduce los datos ( <i>usuario, contraseña y rol</i> ) incorrectos.	El sistema muestra un mensaje de error informándole al usuario que los datos son incorrectos.	Que la BD esté conectada.
El usuario administrador deja alguno de los campos de búsqueda vacío.	El sistema le muestra un mensaje informando al usuario que el campo no puede estar vacío.	Que la BD esté conectada.

**Tabla 13.** Descripción de prueba para el CU Gestionar usuario (Eliminar usuario).

Entrada	Resultado	Condiciones
El usuario administrador selecciona un usuario	El sistema muestra la interfaz de eliminar usuario.	Que la BD esté conectada.



El usuario administrador no selecciona un usuario.	El sistema le muestra un mensaje informando al usuario que debe seleccionar un usuario.	Que la BD esté conectada.
--	---	---------------------------

**Tabla 14.** Descripción de prueba para el CU Gestionar usuario (Cambiar permisos).

Entrada	Resultado	Condiciones
El usuario administrador selecciona los datos (usuario y rol).	El sistema muestra la interfaz de cambiar permiso de usuario.	Que la BD esté conectada.
El usuario administrador deja alguno de los campos sin seleccionar.	El sistema le muestra un mensaje informando al usuario que no debe dejar los campos en blanco.	Que la BD esté conectada.