

Universidad de las Ciencias Informáticas

Facultad 1



Universidad de las Ciencias
Informáticas

“Sistema para el análisis de perfiles de usuarios en la red social Twitter”

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

Autor: Mayelín Victoria Argudín Ferrán

Tutores: MsC. José Gabriel Espinosa Ramírez

Ing. Martha Luisa Gala Rodríguez

Ing. Samuel Ojeda Pereira

La Habana, 2019

“Año 61 de la Revolución”

Declaración de autoría

Declaro por este medio que yo Mayelín Victoria Argudín Ferrán, con carné de identidad 97061306977, soy la autora del trabajo titulado ***“Sistema para el análisis de perfiles de usuarios en la red social Twitter”*** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de junio de 2019.

Mayelín Victoria Argudín Ferrán

Firma de la Autora

MsC. José Gabriel Espinosa Ramírez

Firma del Tutor

Ing. Martha Luisa Gala Rodríguez

Firma de la Tutora

Ing. Samuel Ojeda Pereira

Firma del Tutor

DEDICATORIA

A mi abuela Gloria por ser mi luz, aunque estés un poco lejos y ya no puedo abrazarte, sé que desde el cielo tú me cuidas y vives orgullosa de mí, siempre guiándome por el camino correcto. Sé que te hubiese encantado estar en este momento conmigo, pero Dios no lo quiso así, en donde estés ahora 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 niña graduarse de Ingeniera.

A mis padres por ser el mayor regalo que la vida me pudo dar, por confiar siempre en mí y apoyarme durante todos estos años, por ser mi inspiración en cada meta que me propongo, por dedicarse a mí toda la vida, ahora me corresponde darle valor al fruto de su sacrificio.

AGRADECIMIENTOS

Hoy es el día más importante de mi vida como estudiante, he conquistado un gran sueño que no hubiese sido posible alcanzar sin el inmenso amor e incondicional apoyo de mi familia, la entrega de profesores y la ayuda de mis amigos. Agradezco desde lo más profundo de mí:

A mi mamá porque sin ella nada de esto hubiese sido posible, porque aunque a veces no lo demuestro estoy orgullosa de que seas mi madre, por ser la persona que Dios escogió para traerme a este mundo, a ella muchas gracias por estar conmigo, por ser un modelo de madre, por darme siempre tu amor que aunque es finito parece no tener límites, por tu sacrificio y entrega por mí; por tu aliento en todos los momentos difíciles de mi vida y por tu comprensión y apoyo en cada una de mis decisiones.

A mi papá, a él por darme de sus fuerzas cuando las mías decaían; por acompañarme en los momentos en que sentía estar sola en esto; por levantar mi cabeza y mostrarme una esperanza cuando todo parecía no tener salida, por ser un ejemplo de fuerza y consagración en la vida. Papí todo lo que puedo decir hoy es poco, estoy muy agradecida de ti.

A mi tío Oscar, gracias por consentirme en todas las cosas que siempre quería, por enseñarme en la vida, mucho de lo que sé lo aprendí de ti, aunque la vida haya querido que nos separemos un poco, quiero que sepas que siempre estarás en mi corazón, que vivo agradecida de todo lo que has hecho por mí, por desde pequeña ser esa hija que no tenías, a ti, muchas gracias.

A mi abuela Nereida por ser un sostén incansable, por sus palabras de apoyo, por estar siempre conmigo y porque mientras he crecido Dios me ha enseñado que es un tesoro para mí.

A mis hermanas Maylín y Maydelín y mi sobrino Justín por ser tan importantes en mi vida, por estar al tanto de mí en el transcurso de esta carrera.

A absolutamente cada miembro de mi familia, a los que tengo cerca, los que están un poco lejos y a los que ya no puedo abrazar porque esta familia es una de las razones por las que he permanecido firme en este lugar.

Agradecimiento especial a Samuel, no tengo palabras para agradecerte, por tan joven enfrentar con gran responsabilidad, desinterés y consagración esta tarea, muchas gracias por todo el tiempo que me dedicaste porque tus horas de descanso se convirtieron en Mayelín, gracias por estar cada vez que te necesité, por esas llamadas a cualquier hora diciéndote que borraré algo, o algo ya no me funciona, por mandarme a estudiar y a buscar una información cuando sabías la respuesta, por proveerme de los conocimientos necesarios para realizar este trabajo y por nunca decirme que No. A usted muchas gracias.

A mi novio Juan Carlos porque sin él creo que este momento hubiese sido casi imposible, por impulsar la mayor parte de mi vida. Por enseñarme, consolarme y corregirme en todo este tiempo, gracias por haber marcado mi vida, por tu paciencia, y amor en todo momento, por soportarme en estos años y estar siempre a mi lado, incluso en los momentos más difíciles, por tu apoyo incondicional y todo tu cariño. Te amo mucho mi vida.

A Heídar por su amor invaluable e incondicional, sus consejos, su apoyo en cada minuto. Aunque estés lejos este logro te lo debo a ti también.

A mi mejor amiga Liana, por ser tan excelente persona, por los estresantes momentos que pasamos, por ser esa persona que siempre ha estado ahí en las buenas y en las malas, por soportarme por todos estos años, por aconsejarme y por corregirme cuando ha sido necesario, A ti, por ser la mejor de todas, por convertir horas de estrés en momentos de alegría y diversión.

A mi segunda familia los Álvarez Medina, por siempre darme la bendición para mi vida y la de mi familia, esa segunda familia que el Señor me ha regalado llegue mis más profundos agradecimientos por su incansable ayuda, por siempre preocuparse, ocuparse y estar al tanto de mí, a ustedes muchas gracias.

A mi madrina Ileana por su ayuda en todo este tiempo, por su aliento y su fuerza, por siempre decirme no te preocupes que ya te falta poco.

A mis amigos, los buenos y los mejores: Irianna, Ivancito, Guido, Jhony, Alexei, Antonio, Ismael, gracias por esos momentos que hemos compartido y vivido juntos, esa alegría, esos tiempos que han hecho recuerdos inolvidables.

A los amigos de la infancia y adolescencia Jessica y Manuel, esos que, aunque te demores en verlos las cosas siguen siendo iguales entre nosotros.

A mis tutores: Martha y José Gabriel por el apoyo, la ayuda y sobre todo la paciencia que tuvieron conmigo. Muchas gracias por ser parte de este trabajo.

Agradecimiento especial a dos personas que han sido mis tutores también, que cuando los necesité no me dijeron nunca no, gracias por su experiencia y conocimiento brindado, por sus sugerencias para obtener este logro: Ivaniét y Yordanka.

A los profes Yanelis Benítez, Hubert, Julio, Ernesto, Máxora, Juan Manuel, Maykel, Addiel, Chávez, Osmar, Yaniel, que me prepararon durante estos 5 años, por formarme como profesional, por sus muestras de afecto, por sus consejos y su dedicación.

A los nuevos amigos que descubres en este camino, a los que te acompañan todos los días en el aula, y los que en momentos difíciles te extienden la mano y te brindan su corazón para siempre: Lissy, Mariana, Daniela.

A mis compañeros de aula que no acabaría si los menciono a cada uno, a los que aún siguen en la UCI y a los que ya se han ido, a los soportables y a los insoportables, a todos muchas gracias por estar ahí desde el principio.

A la Revolución por la oportunidad de estudiar esta carrera.

A todo el que me preguntó alguna vez: “y la tesis... ¿cómo va?”.

A las personas que de una forma u otra han hecho posible que hoy escriba con la satisfacción del deber cumplido estas líneas de agradecimientos. Cualquier palabra es poca para expresar lo que siento... a todos:

Muchas gracias



"Sólo llegarán a la cima, los que en su empeño tengan fe."

Ernesto Guevara de la Serna

Resumen

Con el desarrollo de Internet, las redes sociales se han convertido en catalizadores para movimientos sociales de diferentes tendencias, en especial Twitter, el cual se distingue por su dinamismo y por jugar un papel central en acontecimientos relevantes a escala global. Esta red social, maneja un conjunto de información relacionada con sus usuarios mediante un perfil. Almacenar la información en perfiles, resulta útil a la hora de realizar un seguimiento sobre la actividad que realizan sus usuarios, así como: identificar sus intereses, preferencias y comportamiento. Existen diferentes herramientas informáticas para realizar estudios de perfiles las cuales consisten en obtener información sobre los usuarios de Twitter. A partir de un análisis realizado sobre estas herramientas, se demostró que la información proveniente es sobre determinado aspecto del perfil de los usuarios, hecho que dificulta el análisis que realiza el Departamento Operaciones Web y Análisis de Información del Centro de Ideoinformática. La presente investigación tiene como objetivo desarrollar un sistema informático que contribuya de forma centralizada a la obtención de información para el análisis de perfiles de usuarios en la red social Twitter. En la elaboración de la propuesta de solución se utilizó la metodología de software Variación de AUP para la UCI y se emplearon herramientas libres para garantizar la soberanía tecnológica del sistema desarrollado. Se logró la informatización de las necesidades del cliente y se comprobó el correcto funcionamiento de las funcionalidades implementadas mediante la aplicación de pruebas de software, garantizando la calidad de este producto y la satisfacción del cliente.

Palabras Clave: perfiles de usuarios, redes sociales, Twitter.

Índice

| | |
|---|----|
| <i>Introducción</i> | 1 |
| <i>Capítulo 1: Fundamentación teórica sobre el análisis de perfiles de usuarios en la red social Twitter</i> | 6 |
| 1.1 Principales conceptos asociados a la investigación | 6 |
| 1.2 Sistemas informáticos existentes asociados al análisis de perfiles de usuarios en la red social Twitter | 9 |
| 1.3 Metodología de desarrollo de software..... | 12 |
| 1.4 Lenguajes y herramientas para el modelado del sistema..... | 13 |
| 1.4.1 Lenguaje unificado de modelado (UML 2.0)..... | 13 |
| 1.4.2 Herramienta de modelado | 14 |
| 1.5 Base tecnológica para la implementación del sistema..... | 14 |
| 1.6 Conclusiones del capítulo | 24 |
| <i>Capítulo 2: Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter</i> | 25 |
| 2.1 Propuesta de solución | 25 |
| 2.2 Requisitos..... | 26 |
| 2.2.1 Requisitos funcionales..... | 26 |
| 2.2.2 Requisitos no funcionales | 27 |
| 2.2.3 Historias de usuario | 28 |
| 2.3 Análisis y Diseño..... | 30 |
| 2.3.1 Diagrama de clases del diseño | 30 |
| 2.3.2 Patrones del diseño de software | 31 |
| 2.3.3 Modelado de datos..... | 32 |
| 2.3.4 Diseño arquitectónico..... | 33 |
| 2.3.5 Diagrama de Secuencia..... | 35 |
| 2.3.6 Diagrama de Despliegue | 36 |
| 2.4 Conclusiones del capítulo | 37 |
| <i>Capítulo 3: Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter</i> | 39 |
| 3.1 Implementación..... | 39 |
| 3.1.1 Diagrama de componentes | 39 |

| | |
|---|----|
| 3.1.2 Estándares de codificación | 40 |
| 3.2 Validación de la propuesta de solución..... | 43 |
| 3.2.1 Pruebas funcionales | 44 |
| 3.2.2 Pruebas de rendimiento..... | 45 |
| 3.2.3 Pruebas de seguridad..... | 47 |
| 3.3 Análisis de la aplicación de la técnica de Iadov..... | 48 |
| 3.4 Interfaz gráfica de usuario..... | 52 |
| 3.5 Conclusiones del capítulo | 52 |
| Conclusiones generales | 54 |
| Recomendaciones..... | 55 |
| Referencias Bibliográficas..... | 56 |
| Anexos | 60 |

Índice de tablas

| | |
|--|-----------|
| <i>Tabla 1 Comparación entre las aplicaciones estudiadas.....</i> | <i>12</i> |
| <i>Tabla 2 Listado de requisitos funcionales.....</i> | <i>27</i> |
| <i>Tabla 3 Historia de usuario Mostrar datos más recientes de un usuario dado.....</i> | <i>29</i> |
| <i>Tabla 4 Caso de prueba Mostrar datos más recientes de un usuario dado.....</i> | <i>44</i> |
| <i>Tabla 5 Resultado de la prueba de rendimiento.....</i> | <i>47</i> |
| <i>Tabla 6 Cuadro Lógico de Iadov utilizado.....</i> | <i>49</i> |
| <i>Tabla 7 Escala del Índice de Satisfacción Individual.....</i> | <i>50</i> |
| <i>Tabla 8 Caso de prueba Autenticar usuario</i> | <i>60</i> |
| <i>Tabla 9 Caso de prueba Mostrar datos de un usuario seguido en un intervalo de tiempo.....</i> | <i>60</i> |

Índice de figuras

| | |
|---|-----------|
| <i>Figura 1 Propuesta de solución.....</i> | <i>25</i> |
| <i>Figura 2 Diagrama de clases del diseño Mostrar datos más recientes de un usuario dado.....</i> | <i>31</i> |
| <i>Figura 3 Modelo de datos</i> | <i>33</i> |
| <i>Figura 4 Patrón arquitectónico Modelo Vista Plantilla.....</i> | <i>34</i> |
| <i>Figura 5 Arquitectura de la propuesta de solución.....</i> | <i>35</i> |
| <i>Figura 6 Diagrama de secuencia Mostrar datos más recientes de un usuario dado.....</i> | <i>36</i> |
| <i>Figura 7 Diagrama de despliegue.....</i> | <i>37</i> |
| <i>Figura 8 Diagrama de componentes</i> | <i>40</i> |
| <i>Figura 9 Ejemplificación del uso de los estándares de codificación.....</i> | <i>42</i> |
| <i>Figura 10 Ejemplificación del uso de los estándares de codificación.....</i> | <i>42</i> |
| <i>Figura 11 Resultado de las pruebas funcionales</i> | <i>45</i> |
| <i>Figura 12 Resultado de las pruebas de seguridad.....</i> | <i>48</i> |
| <i>Figura 13 Interfaz gráfica Mostrar datos más recientes de un usuario dado.....</i> | <i>52</i> |

Introducción

Las redes sociales implican una nueva realidad con respecto a la comunicación entre los individuos en el siglo XXI, fenómeno impulsado a partir del surgimiento y expansión de Internet. El nuevo formato de las relaciones sociales comienza a materializarse en el momento que un usuario se crea un perfil dentro de una determinada red social, el cual le permitirá compartir esos datos con el resto de miembros de esa red (Nuñez, 2014). Además, se podrán desarrollar actividades como la publicación de fotos en las que puedes asociar a otros usuarios, la creación de grupos donde participen varias personas con un interés común o el seguimiento de personalidades públicas.

En la actualidad, las redes sociales se utilizan para la comunicación, así como para compartir e intercambiar información en diferentes medios. Las grandes corporaciones, instituciones, organizaciones y empresas por lo general suelen hacer uso de estos espacios digitales para promover sus productos y servicios, y tener un contacto más cercano con los clientes y afiliados. Según (Ramos, 2015), entre las más empleadas se encuentran: Facebook, Instagram, Snapchat, Tumblr, Youtube, Twitter, por solo mencionar algunas.

Twitter, se ha transformado en una importante plataforma para los movimientos sociales y un medidor de los estados de opinión. Además, constituye un medio en el que se comparte información constantemente, considerándose una red social altamente exitosa con un heterogéneo grupo de usuarios. Es también una gran fuente de datos y uno de los sistemas de comunicación más utilizados, debido a su actualización constante de contenido (Cabrera, 2015).

En Twitter, los usuarios crean sus propios perfiles, los cuales poseen gran valor para los negocios, los organismos, las instituciones y las empresas porque permiten conocer las tendencias y preferencias de los usuarios en la red. De esta manera, se puede afirmar, que un perfil de usuario es la identidad que caracteriza un individuo en un momento y/o lugar determinado (Hugo, 2014). Además, juegan un rol fundamental porque no solamente representan una mera auto-presentación, sino también un tipo de requisito previo para identificar un usuario ante los demás y a su vez presenta una forma de comunicarse con otros.

El análisis de un perfil de usuario en el mundo, posee una relevante importancia porque permite establecer relaciones para comparar el comportamiento y rendimiento de los usuarios y determinar la

autoridad social que tiene a partir del estudio de la competencia a través de la técnica de **benchmarking** (Mateo, 2016). Esta técnica se utiliza para crear una marca en el mercado a partir del análisis de los competidores (especialmente a los principales o a los líderes del mercado) para efectuar una retroalimentación y un seguimiento continuo de las estrategias, intereses y tendencias de un usuario determinado (Kerlinger, 2016). Se utiliza para documentar y comunicar la información recolectada por el estudio del mercado y sirve de apoyo al desarrollar las historias de usuarios.

El Departamento de Operaciones Web y Análisis de Información (DOWAI) del Centro de Ideoinformática (CIDI), de la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI) tiene entre sus funciones monitorear y analizar temas de interés que se difunden en Internet. Una de las líneas de trabajo de DOWAI está orientada al análisis de redes sociales, fundamentalmente Twitter, con el objetivo de analizar el comportamiento de los perfiles de los usuarios en esta red social. Este análisis se realiza a partir de solicitudes que los Organismos de la Administración Central del Estado (OACE) le hacen al centro; y consiste en recopilar de cada usuario la comunidad de seguidores, la composición de los *tweets*; así como la actividad que realizan en general.

Este análisis se realiza utilizando diferentes aplicaciones informáticas desarrolladas por terceros que existen en Internet como: *Twitonomy*, *Followerwonk*, *Twitteraudit* y *Tweetstats*. Estas herramientas generan datos específicos sobre determinado aspecto del perfil de los usuarios, por lo que resulta engorroso el proceso de elaboración de informes al tener que buscar y relacionar datos desde diferentes fuentes de manera manual; tributando, en ocasiones, a la pérdida de información. Esto provoca pérdida de tiempo al no poder darle seguimiento a un usuario e impide que la toma de decisiones en función de los informes resulte efectiva.

Además, no permiten mostrar la información por intervalos de tiempo, dificultando el análisis de los perfiles en determinado momento; los resultados que brindan de manera gratuita son básicos, limitando el acceso a mayor nivel de profundidad del comportamiento de los usuarios.

Por lo anteriormente planteado se propone como **problema de investigación**: ¿Cómo contribuir a la obtención de información para el análisis de perfiles de usuarios en la red social Twitter?

Se define como **objeto de estudio**: los sistemas para el análisis de perfiles de usuarios en las redes sociales, el cual enmarca al siguiente **campo de acción**: los sistemas para el análisis de perfiles de

usuarios en la red social Twitter.

Para solucionar el problema de investigación antes mencionado, se plantea como **objetivo general**: Desarrollar un sistema informático que contribuya de forma centralizada a la obtención de información para el análisis de perfiles de usuarios en la red social Twitter.

Para guiar el proceso investigativo se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de análisis de perfiles de usuarios en la red social Twitter?
- ¿Cuáles son las tecnologías adecuadas para la implementación del sistema para el análisis de perfiles de usuarios en la red social Twitter?
- ¿Qué elementos de la metodología de desarrollo de software se deben tener en cuenta en el diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter?
- ¿Cómo materializar en términos de componentes el sistema para el análisis de perfiles de usuarios en la red social Twitter?
- ¿Cómo evaluar el funcionamiento del sistema para el análisis de perfiles de usuarios en la red social Twitter?

Con el propósito de dar cumplimiento a las preguntas científicas planteadas se definen las siguientes **tareas de investigación**:

- Análisis de los fundamentos teóricos que sustentan el proceso de análisis de perfiles de usuarios en la red social Twitter.
- Valoración del estado del arte de las herramientas o aplicaciones que realicen análisis de perfiles de usuarios en la red social Twitter.
- Definición de la metodología a utilizar en el desarrollo del sistema para el análisis de perfiles de usuarios en la red social Twitter.
- Implementación del sistema para el análisis de perfiles de usuarios en la red social Twitter.
- Valoración de los resultados de las pruebas al sistema para el análisis de perfiles de usuarios en la red social Twitter.

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

Métodos teóricos: Estos métodos 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 un sistema para el análisis de perfiles de usuarios en la red social Twitter.
- **Método Analítico-Sintético:** Se emplea para realizar un estudio y análisis de toda la bibliografía consultada referente a las herramientas que utilizan a Twitter para generar datos específicos sobre determinado perfil de usuario; así como el análisis de las tecnologías y metodología a utilizar, para luego sintetizar los elementos relevantes para el desarrollo de la solución.
- **Modelación:** Este método contribuye a la realización de los artefactos correspondientes al análisis y diseño e implementación de las funcionalidades del sistema.

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

- **Entrevista:** Este método es de gran utilidad para recopilar información cualitativa. Con dicha información se pudo precisar el problema de investigación y las necesidades del cliente. Para el efecto se elaboraron cuestionarios no estructurados, es decir, conformados por preguntas abiertas que fueron destinadas al cliente.
- **Encuesta:** Este método se aplica para obtener información del nivel de satisfacción de los especialistas de DOWAI sobre el sistema de análisis de perfiles de usuarios en la red social Twitter a través de la técnica de *ladov*.
- **Observación empírica:** Este método se utiliza para el estudio de las aplicaciones informáticas existentes que realizan análisis de perfiles de usuarios en la red social Twitter y se observa cómo estas aplicaciones efectúan el análisis de seguidores, *tweets* y etiquetas.

Capítulo 1: Fundamentación teórica sobre el análisis de perfiles de usuarios en la red social Twitter:

En el capítulo se presentan los conceptos abordados a lo largo de la investigación. Se incluyen los resultados del análisis realizado a diferentes aplicaciones informáticas que estudian los perfiles de

usuarios de Twitter y se exponen las principales características de las tecnologías utilizadas para la implementación de la solución propuesta; así como, la metodología y herramientas a utilizar.

Capítulo 2: Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter:

En este capítulo se documentan los productos de trabajos realizados en las disciplinas Requisitos y Análisis y diseño. Los requisitos se describen de acuerdo a la metodología definida. Contiene los diferentes diagramas elaborados para el diseño de la propuesta de solución.

Capítulo 3: Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter:

En este capítulo se describe la organización del sistema en un diagrama de componentes. Se especifican los estándares de codificación utilizados para la implementación de la propuesta de solución. Se muestran los resultados obtenidos al aplicar una estrategia de pruebas para evaluar la calidad de la propuesta de solución desarrollada.

Capítulo 1: Fundamentación teórica sobre el análisis de perfiles de usuarios en la red social Twitter

Introducción

El análisis de los perfiles de usuarios en la red social Twitter permite seguir el comportamiento y la actividad de un usuario determinado. Con el objetivo de lograr una mayor comprensión del alcance de la investigación y esclarecer su objeto de estudio, se presentan los principales conceptos asociados a esta red social, así como la metodología de software a utilizar para el desarrollo de la propuesta de solución, enmarcado en la definición de las herramientas y lenguajes que se emplean en el desarrollo del sistema informático para el análisis de perfiles de usuarios en la red social Twitter.

1.1 Principales conceptos asociados a la investigación

Red social en Internet

Las redes sociales en Internet, son formas de interacción social, formadas por comunidades de individuos con intereses o actividades en común. Están definidas como un intercambio dinámico entre personas, grupos e instituciones. Estas, son 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 (Martinez, 2015).

Según (Ramos, 2015), entre las más empleadas se encuentran: Facebook, Instagram, Snapchat, Tumblr, Youtube, Twitter, por solo mencionar algunas.

Red social Twitter

Twitter es un servicio gratuito de *microblogging* creado a finales del 2006, que permite a sus usuarios enviar micro entradas basadas en textos cortos llamados *tweets*, de una longitud máxima de 280 caracteres. El envío de estos mensajes se puede realizar de manera tradicional a través del sitio web de Twitter o por el servicio de mensajes cortos (*Short Message Service* dada sus siglas en inglés SMS) desde un teléfono móvil (Salinas, 2017).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Perfil de usuario

Los estudios de usuarios son un conjunto de técnicas de investigación social que se utiliza para determinar el comportamiento de los mismos. El objetivo fundamental de estos estudios consiste en suministrar datos verídicos que caractericen a los usuarios, y de esta forma, complementar el trabajo de las personas encargadas de manipularlos. Se obtiene como resultado un compendio de información personal y rasgos distintivos de un individuo denominada perfil de usuario (Salazar, 2016).

Según (Fernández, 2017), el perfil de usuario es un "...conjunto de medidas diferentes de una persona o grupo, cada una de las cuales se expresa en la misma unidad de medición..." Mientras que, para la Real Academia Española, es un conjunto de rasgos peculiares que caracterizan a alguien o algo.

Perfil de usuario en las redes sociales

Las redes sociales tienen como característica principal la definición de un perfil de usuario, con el cual las personas y entidades se agrupan de diferentes modos: éticos, culturales, de intercambio y de negociaciones. El mismo se construye a partir de las características que distinguen a un usuario de otro y de los factores de influencia que lo circundan (Montero, 2014). Entre los elementos que componen un perfil de usuario se encuentran los siguientes (Montero, 2014):

- Disciplina de interés (necesidades de información).
- Nivel de escolaridad.
- Función o actividad principal: administrador, científico e investigador.
- Recursos de información utilizados o requeridos: fuentes primarias o secundarias.
- Métodos para localizar la información: citas, uso de la biblioteca, documentación personal.
- Comportamiento en la búsqueda de información: búsqueda personal o delegada.
- Manejo del lenguaje.

Análisis de perfiles de usuarios en la red social Twitter

El análisis de los perfiles de usuarios en la red social Twitter se centra en la visualización del perfil del tipo de tuitero, considerando las siguientes características (Urrutia, 2015):

- **Liderazgo:** Entendiendo con ello qué se desea visualizar, qué tweet es el que se está hablando con más recurrencia y a quién pertenece. Para ello es necesario contar con los mensajes emitidos

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

y con información como la cantidad de “retweet” y “Me gusta” que poseen los *tweets* que hacen referencia a los diferentes perfiles.

- **Experiencia:** Este enfoque busca visualizar qué tan participativo o experto es el usuario que realiza el *tweet*, ya sea porque posee gran cantidad de mensajes emitidos, alta cantidad de personas a las que les interesa su perfil, por la cantidad de perfiles de interés o la cantidad de “Me gusta” que realiza a diferentes *tweets*. Para esto, la información que se requiere es la relacionada con el perfil del usuario, donde los datos relevantes para este análisis corresponden a la fecha de creación de dicha cuenta, la cantidad de *tweets*, cantidad de *followers*, cantidad de *following* y cantidad de *likes* que posee el usuario.
- **Origen:** Utiliza la información relacionada con la localidad del usuario, que es ingresada por él mismo al momento de crear su cuenta. Esto restringe a obtener un análisis efectivo, ya que la localidad se presenta en diversos formatos o incluso de manera incorrecta. Además, esta información no es considerada obligatoria y es frecuente encontrar usuarios que no la definen.

API de Twitter

Twitter proporciona herramientas para el trabajo, conocidas como interfaces de programación de aplicaciones (*Application Programming Interface*, dada sus siglas en inglés API) a los desarrolladores para integrar sus soluciones o interactuar con los servicios de la red social desde y fuera de ésta. Además, ofrece las facilidades necesarias para que las aplicaciones de terceros interactúen con ella mediante sus API's. Haciendo uso de estas API's, se han creado un conjunto de aplicaciones en línea que permiten generar datos específicos sobre un determinado perfil de usuario.

Twitter tiene publicada a disposición de sus usuarios su propia API, de forma tal que se puedan crear aplicaciones que se comuniquen con ésta, teniendo en cuenta ciertas restricciones. En el presente apartado, se hace un análisis acerca del API de Twitter y como ésta ayudará en el desarrollo del proyecto (Salinas, 2017).

La API de Twitter tiene tres partes claramente diferenciadas, dos pertenecen al API Rest y la otra corresponde con la API de Streaming. Si bien, podemos dividir la API Rest en dos API's, quedando la API estructurado de la siguiente forma:

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

La Rest API ofrece a los desarrolladores el acceso al *core* de los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde la API. Esta API permite obtener la colección de los *tweets* más recientes de un usuario determinado, así como un *tweet* determinado que se pasa por parámetro incluyendo el autor del mismo. Permite obtener una colección de *tweets* relevantes que coincidan con una consulta determinada, entre otras funcionalidades. Dependiendo de la operación requiere o no autenticación, con el mismo criterio que en el acceso web. Soporta los formatos: *xml*, *json*, *rss*, *atom* (Goebelbecker, 2017).

La Search API suministra los *tweets* con una profundidad en el tiempo de 7 días que se ajustan a la consulta solicitada. Es posible filtrar por cliente, lenguaje y localización. No requiere autenticación y los *tweets* se obtienen en formato *json* o *atom*. Ofrece una información más limitada del *tweet*, en concreto sobre los datos del autor en el que solo indica el Id, el nombre del usuario y la URL de su avatar. Las otras dos API's si ofrecen el perfil completo del autor en el momento de la escritura del *tweet* (Fackensy, 2016). El conjunto de las API's de Streaming que ofrece Twitter proporciona a los desarrolladores acceso de baja latencia de los datos *tweet*. Se establece una conexión permanente por usuarios con los servidores de Twitter y mediante una petición HTTP se recibe un flujo continuo de *tweets* en formato *json*. Se puede obtener una muestra aleatoria, un filtrado por palabras claves o por usuarios (Fintechs, 2017).

1.2 Sistemas informáticos existentes asociados al análisis de perfiles de usuarios en la red social Twitter

Existen aplicaciones en línea que permiten generar datos sobre un determinado perfil de usuario en la red social Twitter. Como parte del estudio se estudiaron algunas de estas herramientas, enfatizando en su funcionamiento de manera general y en los servicios que brindan. Este estudio se efectúa con el objetivo de resumir sus características, ventajas y carencias para el desarrollo de la propuesta de solución. Aplicando el método de observación empírica se realiza un análisis de las herramientas que se presentan a continuación:

Twitonomy

Es una plataforma en línea <https://twitonomy.com/> para conocer las estadísticas de un perfil de Twitter: número de seguidores, promedio de tweets por día, influencia, intereses, ubicación, crecimiento diario de seguidores y palabras clave o *hashtag* (término en inglés, acrónimo utilizado en redes sociales). Su

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

interfaz tiene un diseño muy intuitivo y la navegación por la página es rápida y sencilla. La versión gratuita de Twitonomy solo tiene acceso a los últimos 3.200 *tweets* de un usuario y la de pago es de 20 dólares en un mes o 199 dólares un año. Con la versión gratuita no incluye un informe exportable a Excel ni a PDF, pero los usuarios pueden acceder a diferentes opciones:

- **Dashboard:** sección para saber el número de *tweets*, seguidores, número de listas en las que está registrado una cuenta de Twitter.
- **Profile:** una de las opciones más amplias de la herramienta; en esta sección se presentan estadísticas de los *tweets* (número de *tweets* por día, número de menciones, *retweets*, enlaces, favoritos, meses con más publicaciones); usuarios con más *retweets*, usuarios con más menciones, *hashtag* más utilizados, días y horas de la semana con más *retweets* y seguidores con más influencia.
- **Mentions & RTs:** muestra el número de menciones por día y año, usuarios más activos, lista de publicaciones con más *retweets* y marcado como favorito.
- **Search:** opción para analizar un determinado *hashtag*.
- **Followers:** muestra una lista de seguidores de la cuenta (nombre de usuario, fecha de creación de la cuenta, número de *tweets*, seguidores, seguidos, listas y fecha del último *tweet*).
- **Following:** opción para buscar, filtrar y obtener información de las cuentas que se siguen. Muestra una completa tabla de datos sobre el nombre de usuario, fecha de creación de la cuenta, número de *tweets*, seguidores, seguidos, listas y fecha del último *tweet*.
- **Lists:** muestra las listas creadas por el usuario y su actividad en tiempo real.

Followerwonk

Esta herramienta se encuentra disponible en <https://followerwonk.com/>, junto a una versión de pago, dispone, de una opción gratuita. Esta última, ofrece métricas e informes de valor para cualquiera que desee optimizar el uso de su perfil en Twitter, permite buscar palabras claves dentro de las actividades de los usuarios. De este modo permite identificar aquellos que tengan mayor alcance y autoridad, brindando la posibilidad de comparar cuentas y su rendimiento, al igual que llevar a cabo un análisis de la competencia.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Funcionalidades de Followerwonk

- **Comparativa:** en el caso de que se introduzcan varias cuentas de Twitter será posible realizar una comparativa. En ella se tomarán en cuenta aspectos como el número de seguidores y de publicaciones.
- **Estadísticas:** la solución ofrece gráficos con estadísticas que facilitarán información como: la ubicación de los seguidores, las horas a las que se conectan y el porcentaje de hombres y mujeres o edad.
- **Seguimiento:** *Followerwonk* también indica el número de seguidores que se va ganando y perdiendo por día. Además, ofrece opciones para conocer los seguidores y a quién sigue una cuenta. Al igual que la clasificación por distintos criterios como influencia, número de seguidores, por ejemplo.

Twitteraudit

Se encuentra disponible en <https://www.twitteraudit.com/>. Analiza cualquier cuenta en sólo tres minutos. Para ello, toma como referencia 5.000 seguidores del usuario y le asigna una puntuación según el número de *tweets*, la fecha del último *tweet* y la proporción de seguidores/seguidos. Y a continuación, asigna un porcentaje de “veracidad” de los seguidores de dicha cuenta. Twitteraudit, permite ver el número de seguidores “fake” o falsos de cualquier cuenta de Twitter, sea o no la personal e incluso sin tener que registrarse (y, por supuesto, gratis). También indica gráficamente cómo ve la cuenta (si considera que además de tener más o menos seguidores falsos, el que tuitea es real o no).

Tweetstats

Se encuentra disponible en <https://www.tweetstats.com/>. Muestra un informe sobre el comportamiento de un perfil de Twitter. Sólo es necesario introducir el nombre del usuario, y esperar hasta que se genere el informe que suele tardar algunos minutos para generarse. En el informe se puede ver, de manera gráfica, el número de *tweets* publicados por el usuario, desglosando los datos mensuales desde la creación de la cuenta. Además, también se puede conocer los detalles de las interacciones, consultar gráficas acumuladas de los días y horas con mayor índice de publicación o las aplicaciones que se han empleado para publicar los *tweets*. Por otra parte, muestra una nube de etiquetas y de palabras de la cuenta analizada. Actualmente está experimentando errores a la hora de analizar algunos perfiles.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

A continuación se muestra una tabla comparativa de las aplicaciones estudiadas con respecto a las necesidades del Departamento DOWAI.

Tabla 1 Comparación entre las aplicaciones estudiadas
Fuente: Elaboración propia

| Aplicaciones/ Métricas | Twitonomy | Followerwonk | Twitteraudit | Tweetstats |
|-------------------------------|-----------|--------------|--------------|------------|
| Análisis de los seguidores | Sí | Sí | Sí | No |
| Análisis de los <i>tweets</i> | Sí | No | Sí | Sí |
| Análisis de las etiquetas | No | No | No | Sí |

Conclusiones de las aplicaciones estudiadas

En el análisis anterior se identifica que estas aplicaciones informáticas, por sí solas, no responden completamente a los intereses del departamento DOWAI porque las funcionalidades que brindan de manera independiente son básicas. Además, generan datos específicos sobre determinado aspecto del perfil de los usuarios, por lo que resulta engorroso el proceso de elaboración de informes al tener que relacionar datos de diferentes fuentes de manera manual. El estudio realizado permitió la identificación de estas aplicaciones como guía y referencia para el diseño de la propuesta de solución debido a que realizan análisis de grandes volúmenes de información; Followerwonk y Tweetstats, generan informes permitiendo exportar datos a formato PDF y será utilizado el análisis de seguidores y de *tweets* como se realiza en la herramienta Twitonomy y Twitteraudit.

1.3 Metodología de desarrollo de software

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (Abreu, 2016).

La metodología seleccionada para el desarrollo de la propuesta de solución es **Variación de AUP para la UCI**, elaborada en la propia universidad. Esta metodología logró estandarizar el proceso de desarrollo de software en la universidad, dando cumplimiento a las buenas prácticas y logrando un lenguaje común en

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

cuanto a fases, disciplinas, roles y productos de trabajos. Variación de AUP para la UCI es la metodología de desarrollo de software usada en CIDI, estructura que acoge la investigación en cuestión. La misma cuenta con tres fases (Rodríguez, 2015):

- **Inicio:** durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, para decidir si se ejecuta o no el proyecto.
- **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- **Cierre:** en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

El presente trabajo de diploma transitará por las siguientes disciplinas propuestas en la metodología: Requisitos, Análisis y diseño, Implementación y Pruebas. El producto de trabajo generado durante la elaboración de la solución está basado en el Escenario No. 4 pues se aplica a los proyectos con un negocio bien definido. El cliente está siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Este escenario es conveniente en proyectos no muy extensos (Rodríguez, 2015).

1.4 Lenguajes y herramientas para el modelado del sistema

1.4.1 Lenguaje unificado de modelado (UML 2.0)

UML 2.0 (*Unified Modeling Language*, Lenguaje de Modelado Unificado) provee un conjunto de herramientas con una buena consistencia del lenguaje para especificar, visualizar, construir y documentar los artefactos de un sistema de *software*. Es un lenguaje fácil de aprender, adaptable a cambios y ofrece variedad de diagramas para visualizar el software desde varias vías, facilitando así la relación entre clientes y desarrolladores.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Los objetivos que se establecieron al desarrollar el UML fueron los siguientes (Velázquez, 2015):

1. Proporcionar a los usuarios un Lenguaje de Modelado Visual de tal forma que sea posible intercambiar información de los modelos.
2. Ser independiente de un lenguaje en particular y del proceso de desarrollo.
3. Permitir crear y modificar modelos expresivos mediante la combinación de los 13 tipos de diagramas que suministra en esta versión.

1.4.2 Herramienta de modelado

Visual Paradigm 8.0 para UML

Es una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) profesional que soporta todo el ciclo de vida del *software*: análisis y diseño orientado a objetos, construcción, prueba y despliegue. Dicha herramienta ayuda a una rápida construcción de la aplicación con alta calidad y a un menor costo. Permite modelar todos los tipos de diagrama de clases, código inverso, generar código desde diagramas y generar documentación.

La herramienta también proporciona abundantes tutoriales, demostraciones y proyectos UML. La misma presenta una interfaz de uso intuitiva y con muchas facilidades. Cuenta con la licencia para su utilización y utiliza UML como lenguaje de modelado, que tiene como propósito visualizar, especificar, construir y documentar proyectos de *software*. Es un potente generador de informes en formato PDF/HTML y permite realizar ingeniería inversa y directa a la hora de modelar los diagramas que soportan la ingeniería de requerimientos.

La herramienta está diseñada para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas y analistas de negocio, o para cualquiera que esté interesado en la construcción de forma fiable a gran escala de sistemas de *software*, utilizando un enfoque orientado a objetos. Además, presenta licencia gratuita y comercial (Schmuller, 2015).

1.5 Base tecnológica para la implementación del sistema

A continuación, se describen las principales tecnologías que se utilizarán para el desarrollo de la investigación y la implementación de la solución.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Lenguaje de Programación

Es una estructura con una base sintáctica y semántica que imparte distintas instrucciones a un programa de computadora. Además, es un convenio acerca de cómo se debe interpretar el significado de los programas de un lenguaje porque en ocasiones se confunden los lenguajes con los compiladores, intérpretes o con los entornos de desarrollo de *software* (Kuzko, 2016).

Python

Python es un lenguaje de programación interpretado, orientado a objetos, multiplataforma y de sintaxis sencilla. Permite dividir un programa en módulos reutilizables desde otros programas en Python. El lenguaje incorpora una gran colección de módulos estándar que se pueden utilizar como base de los programas, o como ejemplos para empezar a aprender Python. Permite, además, escribir programas muy compactos y legibles. Estos programas son normalmente mucho más cortos que sus equivalentes en C o C++, es ampliable o conocido como Lenguaje Integrador. Otra característica de Python es que sintácticamente escribir un programa en este lenguaje es muy sencillo, y la razón de esto es que usa tabulación o espaciado para mostrar estructura de bloques (Kuzko, 2016).

Algunas de las ventajas del uso de Python son:

- **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.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

- **Funciones y bibliotecas:** Dispone de muchas funciones incorporadas en el propio lenguaje.
- **Sintaxis clara:** Tiene una sintaxis muy visual, gracias a una notación *indentada* (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.

Java

Es un lenguaje de programación orientado a objetos, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación. Permite abrir, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red. Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los *bytecodes*, semejantes a las instrucciones de ensamblador. Además, es altamente fiable en comparación con C, ya que se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en el tiempo de ejecución. Por otra parte, es interpretado, ya que los *bytecodes* se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (Kuzko, 2016).

Selección del lenguaje de programación a utilizar

El uso del lenguaje de programación Python en su versión 3.6 en esta investigación está regido principalmente por las potencialidades de éste, 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.

Librería de Twitter

Desde el punto de vista del desarrollo, Twitter ofrece una serie de librerías para distintos lenguajes de programación con soporte al API de Twitter, lenguajes como Java, C++, *JavaScript*, PHP, *Python* entre otros (Prist, 2016).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Entre las librerías que facilitan la comunicación con la API de Twitter implementadas por el lenguaje de programación Python se encuentran:

Python-Twitter

Es una librería que proporciona una interfaz de Python puro para trabajar con la API de Twitter de una manera sencilla. Permite realizar acciones utilizando sencillos métodos, pero no se cuenta con mucha bibliografía disponible acerca de la misma (Rodríguez, 2016).

Tweepi

Tweepi es una librería de Python fácil de usar para acceder a la API de Twitter, sirve para conocer y administrar mejor la cuenta de Twitter del usuario. Al utilizar la biblioteca Tweepi es posible realizar un análisis puntual. La herramienta permite seguir o dejar de seguir por grupos de 40 usuarios de forma rápida. Tweepi tiene como ventaja ser gratuita (Magán, 2018).

Python-OAuth

La API utiliza OAuth para el proceso de autorización, permite autorización segura y simple para aplicaciones de escritorio, móviles, y web. OAuth permite a los usuarios compartir su información privada (fotos, videos, listas de contactos, etcétera) almacenada en un sitio con otro sin necesidad de proporcionar sus credenciales de acceso (Python Social Auth's, 2018).

Selección de la librería para la conexión con Twitter

Se decide utilizar Tweepi porque es una librería de código abierto para Python que incluye todo el conjunto de funciones necesarias para comunicar con Twitter mediante las API definidas por éste. Las funciones definidas por esta librería simplifican excesivamente la conexión y la búsqueda. Por ejemplo, toda conexión a Twitter debe estar certificada con OAuth, y mientras que por defecto habría que configurar esta conexión mediante otra librería como Python-OAuth y establecer cada conexión manualmente, Tweepi simplifica esto con funciones que simplemente esperan los parámetros para configurar todo automáticamente, toda la complejidad de las conexiones la trata internamente simplificando el trabajo.

Entorno de desarrollo integrado (IDE)

Un Entorno de Desarrollo Integrado (*Integrated Development Environment*, IDE), es una aplicación de software que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

de software. Normalmente, un IDE lo compone un editor de código fuente, herramientas de construcción automática y un depurador (Gasterol, 2015).

PyCharm 3.2

Es un IDE con un completo juego de herramientas para el desarrollo productivo con el lenguaje de programación Python. Adicionalmente, provee capacidades de alto rango para desarrolladores profesionales de web con el marco de trabajo Django (Gasterol, 2015).

Las principales características de este IDE son (Gasterol, 2015).

- Editor inteligente.
- Depurador de código gráfico.
- Inspección de código.
- Integración de control de versiones.
- Ejecución de pruebas.
- Consola Python integrada.
- Soporta los repositorios Git, Mercurial, Subversion y Github.
- Permite los lenguajes XML (*Extensible Markup Language*, Lenguaje de Marcado Extensible) y HTML (*HyperText Markup Language*, Lenguaje de Marcas de Hipertexto).

Marco de trabajo

Un marco de trabajo (*framework*) es una gran librería o conjunto de librerías donde además de facilitar funciones para su uso, dispone de una sintaxis o metalenguaje específico del marco de trabajo y una forma de organización de su código. Para usar un marco de trabajo no basta con conocer el nombre de la función a utilizar: hay que saber qué sintaxis emplea, qué obligaciones impone (a la hora de organizar el código, archivos) y su lógica o filosofía de trabajo. El uso de marcos de trabajo se ha extendido debido a la gran complejidad de las aplicaciones que se desarrollan ya que facilitan su organización y mantenimiento (Gutiérrez, 2015).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Django 2.1

Es un marco de desarrollo en Python, de código abierto y muy ligero que permite la creación rápida de páginas y aplicaciones web. Pretende ser sencillo y rápido posibilitando concentrarse principalmente en escribir la aplicación (Gutiérrez, 2015).

Las principales características que presenta este marco de trabajo son (Gutiérrez, 2015):

- Basado en la filosofía DRY (*Don't Repeat Yourself*, no te repitas). Muchas aplicaciones web y proyectos digitales comparten numerosas líneas de código unos con otros. Django es el marco de desarrollo de refactorización de código casi por excelencia. Permite reutilizar programación de unas aplicaciones a otras sin la obligación de tener que repetir las mismas líneas de código entre distintos proyectos.
- Es de alto nivel basado en el patrón arquitectónico Modelo-Vista-Plantilla. Apuesta por la sencillez, la rapidez y la reutilización de código.
- Viene con SQLite, una base de datos usada por compañías tan importantes como Facebook o Bloomberg.
- Contiene una API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) propia para el desarrollo de proyectos digitales.

Servidor de aplicaciones web

Nginx 1.14

Este servidor realiza conexiones bidireccionales y/o unidireccionales con el cliente y genera una respuesta a las peticiones realizadas por éste. Nginx es un servidor HTTP, un servidor proxy inverso, un servidor proxy de correo y un servidor proxy genérico para TCP/UDP. En esencia Nginx es un servidor web de alto desempeño que puede manejar una carga alta, es decir, puede funcionar si su sitio web recibe muchos visitantes y peticiones debido a la facilidad de administrar sesiones concurrentes y contenido estático. Además, posee una implementación asíncrona, diseñada para permitir que más solicitudes sean servidas simultáneamente pasando parte del trabajo de procesamiento a los hilos de escucha.

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Algunas de las ventajas que ofrece Nginx:

- Estadísticamente Nginx es un servidor web en crecimiento que atrae cada vez a más *webmasters* a nivel mundial.
- En comparación con Apache y otros servidores web, Nginx supera ampliamente en cuanto al manejo de sesiones concurrentes, tiempo de respuesta y uso de recursos. Esto se debe a la arquitectura y manejo inteligente de conexiones.
- El manejo de archivos estáticos es muy rápido y eficiente.
- Es compatible con aplicaciones web usadas comúnmente, como WordPress, Joomla, Drupal, Ruby o Python.
- El balanceo de cargas es una ventaja muy importante porque permite un escalamiento más rápido de los servidores http. El balanceo a través de Nginx permite la distribución de tráfico entre distintos servidores, permitiendo que los usuarios escalen sus aplicaciones y que obtengan redundancia http simultáneamente.
- Además es de fácil instalación (solo un par de líneas para distribuciones Linux) y configuración a través del archivo principal "nginx.conf" y los archivos de configuración para los hosts virtuales creados (generalmente ubicados en /etc/nginx/conf.d/ejemplox.com.conf) (Naramore, 2016).

Servidor de Base de Datos

PostgreSQL 11

PostgreSQL es un servidor de bases de datos distribuido bajo licencia BSD ¹ y con el código fuente más potente del mercado disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Posee una gran escalabilidad y es capaz de ajustarse al número de CPUs y a la cantidad de memoria que conserva el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de sub consultas y transacciones, haciendo su funcionamiento mucho más eficaz. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de

¹ Berkeley Software Distribution

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

bases de datos de alto nivel (Junco, 2015). Es un sistema estable, robusto y fácil de administrar para un usuario con conocimientos básicos.

Herramientas de maquetado y estilo

Lenguaje de Marcas de Hipertexto HTML 5

HTML está diseñado para estructurar documentos y presentarlos. Define la estructura y el contenido de las páginas permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas. Su nivel de complejidad es bajo y permite incluir dentro de su código scripts escritos en otros lenguajes como PHP y JavaScript. Es un lenguaje sencillo, fácil de entender y de utilizar. La versión 5 de HTML no se limita sólo a crear nuevas etiquetas, atributos y eliminar las marcas que están en desuso o se utilizan inadecuadamente; sino que pretende proporcionar una plataforma para desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio (Álvarez, 2017).

Entre las principales ventajas que presenta HTML5, se tienen las siguientes (Noguera, 2016):

- Nueva estructura de etiquetas mejorada. Permite definir por separado el encabezado, la barra de navegación, las secciones de la página web, los textos del sitio, los diálogos y el pie de página de los sitios web.
- Incluye una nueva etiqueta de dibujo sobre la página web, llamada Canvas. Convierte el proceso de crear dibujos en el sitio web tan fácil como dibujar con aplicaciones al estilo de Paint.
- Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores.

Hoja de Estilo en Cascada CSS 3

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es utilizado para crear páginas visualmente atractivas e interfaces de usuario para aplicaciones web. Está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características de la presentación y reducir la complejidad y la repetición de código en la estructura del documento (Romero, 2017).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Bootstrap 3

Bootstrap es un marco de trabajo de software libre, desarrollado por Twitter, intuitivo y de gran alcance para el diseño de sitios y aplicaciones web. En la actualidad se ha convertido en uno de los proyectos de código abierto de gran popularidad en el mundo. Simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que se puede crear interfaces que se adapten a los distintos navegadores con la propiedad denominada diseño adaptativo. Los diseños creados con Bootstrap son simples, limpios, intuitivos y brindan agilidad a la hora de cargar y adaptarse a otros dispositivos como tabletas y móviles a distintas escalas y resoluciones (Thornton, 2014).

Integración e intercambio de mensajes

RabbitMQ

Para el funcionamiento del sistema desarrollado es necesario orquestar la comunicación entre las aplicaciones que lo componen. Esta comunicación debe suceder de tal forma que los eventos relevantes acaecidos dentro de la frontera de cada una de las aplicaciones puedan ser notificados al resto de las aplicaciones, para que en el caso que corresponda se realicen las acciones necesarias. Para la gestión del intercambio de mensajes se utilizó a RabbitMQ, un sistema para el intercambio de mensajes entre procesos gestionado por la compañía Pivotal. RabbitMQ está basado en código abierto, y se encuentra disponible de forma gratuita bajo licencia *Mozilla Public License* (Ramírez, 2018).

El desarrollo de las aplicaciones siguiendo el diseño propuesto por RabbitMQ dota al sistema de un alto grado de flexibilidad, al utilizar a las colas de RabbitMQ como un punto de intercambio y almacenamiento temporal. Las aplicaciones que componen el sistema pueden ser instanciadas más de una vez para ajustar la capacidad de procesamiento en dependencia de la demanda que se tenga en un momento dado (Ramírez, 2018).

Se utiliza además **Celery** también escrito en Python, es una cola de tareas o trabajos asincrónicos basados en paso de mensajes distribuidos. Se centra en la operación en tiempo real y es muy fácil de integrar con los *frameworks* web. Las unidades de ejecución, llamadas tareas, se ejecutan simultáneamente en un único o varios servidores de trabajo utilizando el multiprocesamiento, se utiliza en los sistemas de producción para procesar millones de tareas de un día (Correaz, 2017).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

Herramienta para el control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Para garantizar el control de versiones en el sistema se emplea el repositorio *GitLab*, específicamente el cliente **Git** en su versión 2.8, el cual, según (Andújar, 2016):

- Es un sistema de control de versiones distribuido.
- No depende de acceso a la red o un repositorio central.
- Está enfocado a la velocidad, uso práctico y manejo de proyectos grandes.

Herramientas para las pruebas de software

Las pruebas de software, son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada (Pressman, 2010). La propuesta de solución de la investigación será validada a través de varios tipos y técnica de pruebas, algunas de ellas de forma manual y otras mediante el uso de herramientas que permite la realización de dicha tarea. Entre las herramientas existentes, se pretende utilizar la herramienta **Acunetix Web Vulnerability Scanner** para las pruebas de seguridad. *Acunetix* realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección *SQL*, *Cross site scripting* y otras vulnerabilidades que puedan ser explotadas por *hackers*. Ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (Guerrero, 2016).

JMeter

JMeter es una herramienta Java dentro del proyecto *Jakarta*, que permite realizar **Pruebas de Rendimiento sobre Aplicaciones Web**. Es una herramienta de carga para llevar a cabo simulaciones sobre cualquier recursos *software*. Se destaca por su versatilidad, estabilidad y por ser de uso gratuito. Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción. En este sentido, simula todas las funcionalidades de un Navegador ("*Browser*"), o de cualquier otro cliente, siendo capaz de manipular resultados en una determinada solicitud y reutilizarlos para ser empleados en una nueva secuencia (Loisel, 2016).

Fundamentación teórica sobre el análisis de los perfiles de usuarios en la red social Twitter

1.6 Conclusiones del capítulo

En el presente capítulo se abordaron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, en tal sentido se puede arribar a las siguientes conclusiones:

1. El análisis de los referentes teóricos sobre el sistema para el análisis de perfiles de usuario en la red social Twitter, así como el estudio de los principales conceptos asociados al problema planteado, permitió sentar las bases para el desarrollo de la investigación y conocer las características del objeto de estudio.
2. La caracterización y comparación de las diferentes aplicaciones que analizan perfiles de usuario en la red social Twitter demostró la necesidad de desarrollar una aplicación web porque las herramientas existentes no resuelven las necesidades que presenta DOWAI.
3. Se definió para la elaboración de la propuesta de solución el empleo de la metodología de software Variación de AUP para la UCI, que guiará la implementación de la solución con *Visual Paradigm 8.0* como herramienta de modelado, *PyCharm* como entorno de desarrollo integrado, Python como lenguaje de programación, Django como *framework* de desarrollo, el sistema gestor de bases de datos PostgreSQL, para la integración y el intercambio de mensajes RabbitMQ y para las pruebas de software la herramienta JMeter para evaluar rendimiento y *Acunetix Web Vulnerability Scanner* para la seguridad.

Capítulo 2: Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

Introducción

Una de las prioridades cuando se desea desarrollar un software, es establecer un entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr, realizando un correcto análisis y diseño de dicho sistema. En este apartado se especifican los requisitos del sistema con el objetivo de proporcionar una definición completa y útil del software, tanto funcional como no funcional. Además se evidencian un conjunto de artefactos definidos en la metodología seleccionada que servirán como guía durante la implementación de la propuesta de solución.

2.1 Propuesta de solución

A partir de entrevistas realizadas al cliente, se determinaron un conjunto de necesidades existentes en DOWAI. Estas necesidades están enfocadas en un sistema para el análisis de perfiles de usuarios en la red social Twitter. Dicho sistema debe permitir utilizar dos mecanismos de autenticación, como usuario del sistema y la autenticación utilizando su cuenta de Twitter; permitiendo así, realizar búsquedas recientes de un usuario determinado o por intervalo de tiempo. Debe generar informes basados en estos datos reales de la plataforma mediante su API, centralizando la fuente y garantizando la transparencia y confiabilidad de la información mostrada.



Figura 1 Propuesta de solución
Fuente: Elaboración propia

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

- **API:** Representa la API de Twitter. Es una aplicación que hace públicos los recursos de Twitter para que los desarrolladores puedan emplearlos en sus productos.
- **Json:** Formato de texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript.
- **Sistema:** Aplicación a desarrollar.
- **Usuarios:** Representa a los integrantes del departamento DOWAI. Solicitan información que los ayuda a realizar análisis de perfiles de usuarios.

2.2 Requisitos

El objetivo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez, 2015). Dentro de las tareas de esta disciplina está la Especificación de Requisitos de Software (ERS) que constituye un medio de comunicación entre clientes, usuarios, ingenieros de requisitos y desarrolladores. En la ERS deben recogerse tanto las necesidades de clientes y usuarios (necesidades del negocio, también conocidas como requisitos de usuario, requisitos de cliente y necesidades de usuario) como los requisitos que debe cumplir el software a desarrollar para satisfacer todas las necesidades (requisitos del producto, también conocidos como requisitos de sistema o requisitos de *software*) (Creed, 2017). A continuación se listan los requisitos funcionales y no funcionales definidos para la implementación de la propuesta de solución como resultado de entrevistas realizadas al cliente.

2.2.1 Requisitos funcionales

Un requisito funcional es la capacidad que debe tener un software para que sea útil en la realización de los procesos de negocio de una organización (Creed, 2017). En la siguiente tabla se listan los requisitos funcionales de la propuesta de solución, la prioridad para el cliente y su complejidad, determinada mediante el producto de trabajo Evaluación de Requisitos del expediente de proyecto 5.0 disponible para la actividad productiva de la universidad.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

Tabla 2 Listado de requisitos funcionales
Fuente: Elaboración propia

| No | Nombre | Prioridad para el cliente | Complejidad |
|-----|---|---------------------------|-------------|
| RF1 | Crear usuario del sistema | Media | Baja |
| RF2 | Eliminar usuario del sistema | Media | Baja |
| RF3 | Modificar permisos de usuarios del sistema | Media | Baja |
| RF4 | Autenticar usuario | Alta | Media |
| RF5 | Mostrar datos más recientes de un usuario dado | Alta | Alta |
| RF6 | Añadir usuario a seguir por intervalo de tiempo | Media | Baja |
| RF7 | Mostrar datos de un usuario seguido en un intervalo de tiempo | Alta | Alta |
| RF8 | Exportar información de los perfiles a formato PDF | Media | Baja |

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de recursos y del proceso de desarrollo, como impuestas por los estándares. Los requisitos no funcionales se aplican al sistema como un todo, más que a características o a servicios individuales del mismo (Creed, 2017).

Se obtuvo un total de ocho (8) requisitos no funcionales, distribuidos en especificaciones de usabilidad, seguridad, eficiencia, portabilidad, hardware y software, los cuales se relacionan a continuación:

RnF 1: Usabilidad

1.1: El sistema deberá ser una aplicación web.

1.2: La aplicación deberá mostrar una interfaz agradable e intuitiva para el usuario.

RnF 2: Seguridad

2.1: Para acceder a las funcionalidades del sistema el usuario debe estar correctamente autenticado.

2.2: El sistema deberá ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

RnF 3: Eficiencia

3.1: El sistema debe permitir que los usuarios interactúen con él de manera concurrente.

RnF 4: Portabilidad

4.1: El sistema se podrá utilizar indistintamente en diferentes sistemas operativos como Linux y Windows.

RnF 5: Hardware

5.1: Debido a que el sistema realiza peticiones al API de Twitter, se necesita acceso a Internet mediante una conexión de red con velocidad de 100MB/s.

5.2: Servidor: 8GB de memoria RAM como requerimiento mínimo, procesador Core i3 o superior.

RnF 6: Software

6.1: Para la ejecución del sistema es necesario que el entorno de despliegue cuente con las siguientes dependencias: Python en su versión 3.6, Django en su versión 2.1, PostgreSQL en su versión 11, Celery y RabbitMQ.

2.2.3 Historias de usuario

La metodología Variación de AUP para la UCI, en su escenario 4 para la disciplina Requisitos, genera como uno de sus artefactos a las Historias de Usuario (HU), que consiste en una técnica para encapsular los requisitos del software, a través de un conjunto de tablas que describen brevemente las características que desea el cliente para el sistema a desarrollar. Estas son lo suficientemente comprensibles y delimitadas para que los programadores puedan implementarlas (Rodríguez, 2015).

En la HU se define quién es el responsable de su concreción, la prioridad en función de otorgar funcionalidades para el desarrollo del sistema y una breve descripción del comportamiento de la o las funcionalidades que componen acompañado de un prototipo de interfaz.

Para la presente investigación se generaron un total de seis (6) HU, a continuación, solo se muestra una (1) de ellas en función de los requisitos funcionales de prioridad alta para el cliente.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

*Tabla 3 Historia de usuario Mostrar datos más recientes de un usuario dado
Fuente: Elaboración propia*

| Historia de usuario | |
|--|---|
| Número: 3 | Nombre: Mostrar datos más recientes de un usuario dado |
| Programador: Mayelín Argudín Ferrán | Iteración asignada: 1 |
| Prioridad en negocio: alta | |
| <p>Descripción: Permite mostrar datos específicos de un usuario dado.</p> <p>El usuario autenticado introduce el usuario de Twitter al que se le desea realizar la búsqueda, luego el sistema verifica la existencia del usuario. Si el usuario existe, el sistema muestra los datos específicos del mismo como: cantidad de seguidores, cantidad de seguidos, usuarios más mencionados, usuarios más retuiteados, los <i>tweets</i> que emite, los tweets marcados como favoritos, tweets más retuiteados y las etiquetas que más utiliza. En caso de no existir, el sistema muestra un mensaje notificando la no existencia del usuario.</p> | |
| <p>Prototipo de interfaz:</p> <div style="border: 1px solid gray; padding: 10px; margin: 10px auto; width: 80%; background-color: #f0f0f0;"> <div style="border-bottom: 1px solid gray; padding-bottom: 5px; margin-bottom: 10px;"> Mostrar datos recientes de un Usuario _ □ X </div> <div style="text-align: center; margin-bottom: 10px;">  </div> <div style="margin-bottom: 10px;"> <p style="text-align: center;">Nombre y Apellidos <input style="width: 100px;" type="text"/></p> </div> <div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> Cantidad de seguidores : 2000 Cantidad seguidos : 30 Cantidad de tweet que emite : 500 </div> <div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> <div style="width: 30%;"> <p>Usuarios más mencionados: <input style="width: 100%;" type="text"/></p> </div> <div style="width: 30%;"> <p>Usuarios más retuiteados: <input style="width: 100%;" type="text"/></p> </div> <div style="width: 30%;"> <p>Etiquetas más utilizadas: <input style="width: 100%;" type="text"/></p> </div> </div> <div style="display: flex; justify-content: center; margin-bottom: 10px;"> <div style="width: 60%;"> <p>Tweets más mencionados : <input style="width: 100%;" type="text"/></p> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p>Usuarios más retuiteados: <input style="width: 100%;" type="text"/></p> </div> <div style="width: 30%;"> <p>Tweets más mencionados : <input style="width: 100%;" type="text"/></p> </div> </div> </div> | |

La disciplina Requisitos permitió definir la propuesta de solución generando especificaciones correctas que describen con claridad, en forma consistente y compacta, las necesidades existentes como una base para el análisis y diseño de la propuesta de solución.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

2.3 Análisis y Diseño

En esta disciplina, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos (Rodríguez, 2015).

2.3.1 Diagrama de clases del diseño

Un diagrama de clases es una presentación gráfica de la vista estática, que muestra una colección de elementos declarativos (estáticos) del modelo, como clases, tipos, contenidos y relaciones. Contiene elementos materializados de comportamiento, como operaciones, pero cuya dinámica se puede representar en otros diagramas, como diagramas de estados o diagramas de colaboración (Ferrer, 2017). A continuación se presenta el diagrama de clases del diseño con estereotipos web para la HU Mostrar datos más recientes de un usuario dado.

Como se puede observar en la figura la acción comienza en el momento que el usuario autenticado accede al formulario Form_FormularioBuscar e introduce el usuario de Twitter del cual se desea obtener información. Este dato se envía a la clase controladora SP_views.py y es usada para a través de una instancia de la clase Conexión_Utiles conectarse con la API de Twitter. Esta última, devuelve un json con todo lo relacionado con ese usuario dado; luego se procesa el json en la SP_view.py y se muestra la información relacionada con el usuario en la CP_Listar_buscqueda.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

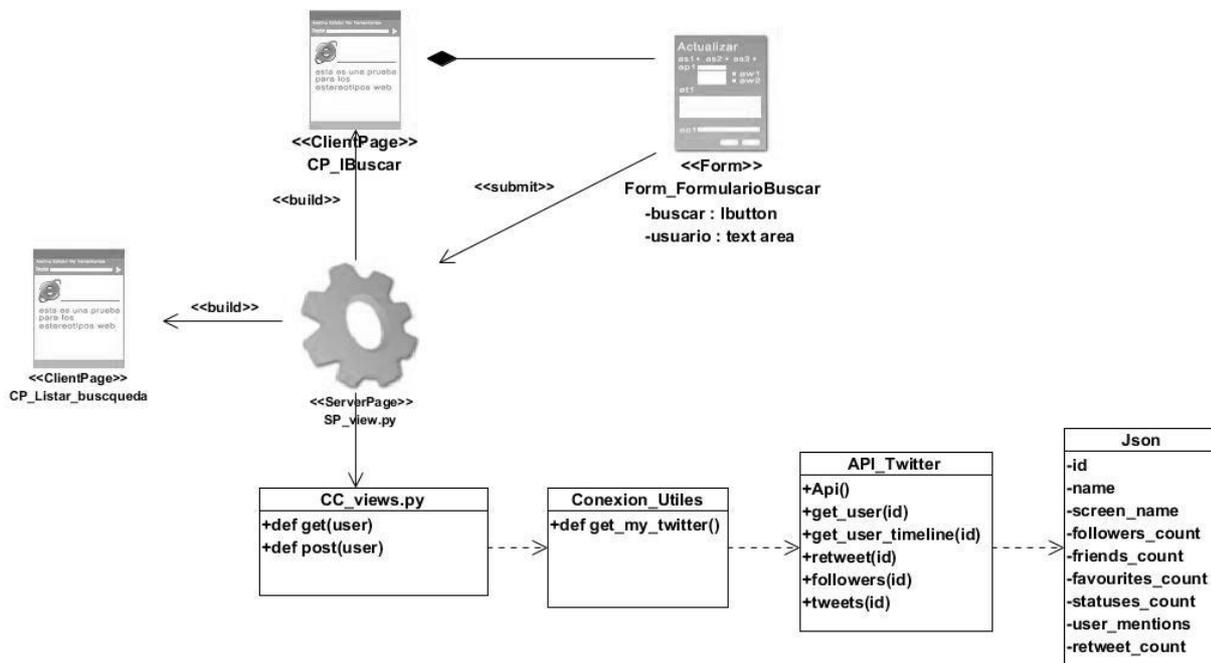


Figura 2 Diagrama de clases del diseño Mostrar datos más recientes de un usuario dado
Fuente: Elaboración propia

El diagrama de clases del diseño con estereotipos web permitió definir la estructura estática de las clases en el sistema, su contenido y las relaciones que se establecen entre ellas. Además, permiten identificar los patrones del diseño que serán utilizados en el desarrollo de la propuesta de solución.

2.3.2 Patrones del diseño de software

Los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí. Constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son descripciones de clases cuyas instancias colaboran entre sí. 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 (Quimera, 2015).

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

En el diseño del sistema para el análisis de perfiles en la red social Twitter se tuvo en cuenta los siguientes patrones:

Patrones GRASP

Bajo acoplamiento: Es la idea de tener las clases lo menos relacionadas 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. Se evidencia en la misma estructura 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, en este caso, Formularios.py y views.py se enfocan totalmente en realizar todas las operaciones relacionadas con formularios dentro del 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. Este patrón se evidencia en el uso de la clase controladora.

Patrones GoF

Decorador: Se evidencia en la capa plantilla del framework Django. La plantilla base.html contiene estructura y diseño común para todo el sistema. El sistema de bloques de las plantillas de Django permite añadir responsabilidades -en este caso fragmentos de código HTML- a las plantillas que heredan de base.html de manera dinámica y transparente, decorándola con contenido particular.

2.3.3 Modelado de datos

Un modelo de datos es un conjunto de conceptos y reglas que permiten estructurar los datos resultantes de la observación de la realidad, de forma que pueden ser representadas todas sus propiedades, tanto estáticas como dinámicas. Los modelos de datos de alto nivel o conceptuales utilizan conceptos muy cercanos a la forma en que los usuarios perciben los datos, mientras que los de bajo nivel o físicos describen detalles de cómo se almacenan los datos en la computadora (Gómez, 2015).

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

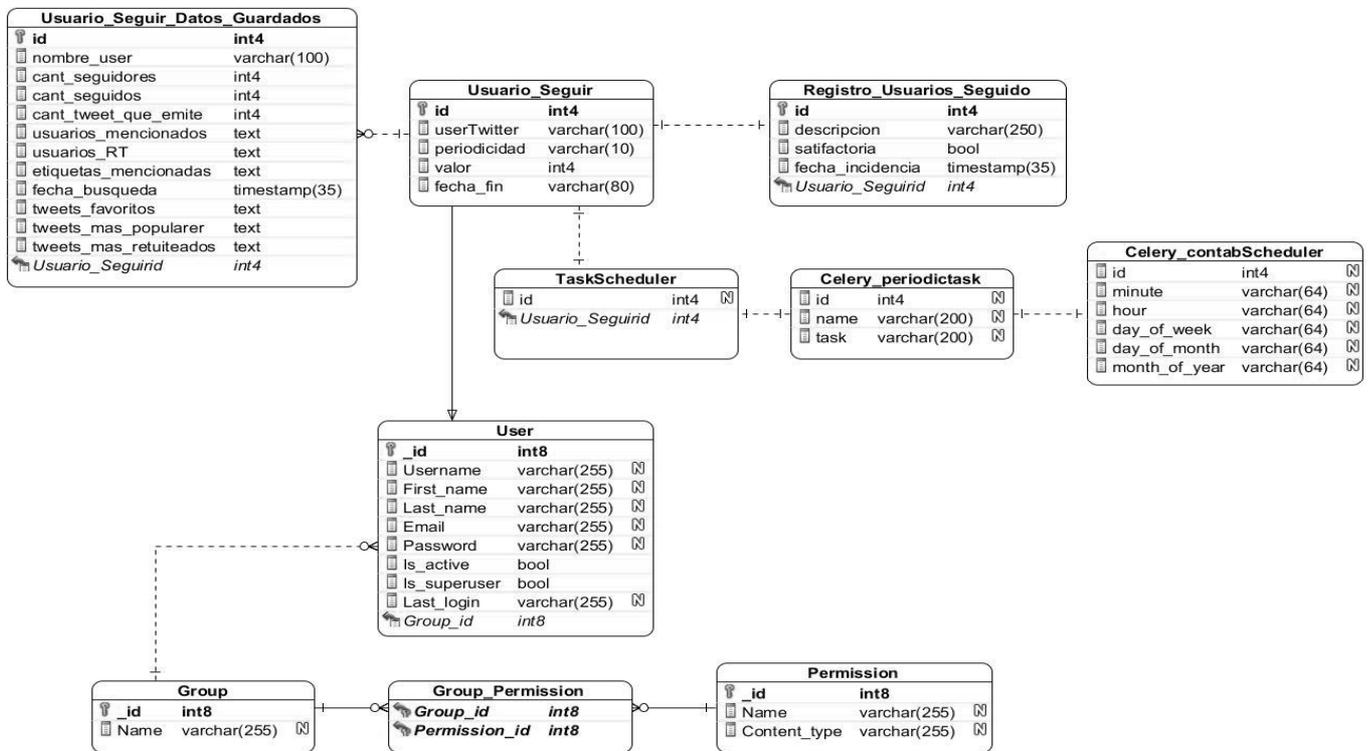


Figura 3 Modelo de datos
Fuente: Elaboración propia

2.3.4 Diseño arquitectónico

Descripción de la Arquitectura de Software

La arquitectura de un software se encarga de entender cómo debe organizarse el sistema y cómo tiene que diseñarse la estructura global de ese sistema. Es el enlace crucial entre el diseño y la ingeniería de requisitos porque identifica los principales componentes estructurales en un sistema y la relación entre ellos. La salida del proceso de diseño arquitectónico consiste en un modelo arquitectónico que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación. La arquitectura de software es importante porque afecta el desempeño y la potencia, así como la capacidad de distribución y mantenimiento de un sistema (Sommerville, 2011).

Según (Pressman, 2010), la arquitectura de un sistema puede basarse en un patrón o un estilo arquitectónico particular. Un patrón arquitectónico es una descripción de una organización del sistema y

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

expresa un esquema organizativo y estructural para sistemas de software especificando subsistemas con sus responsabilidades, detallando un grupo de recomendaciones para organizar sus componentes.

En el desarrollo de la solución se utiliza el patrón arquitectónico MVT (*Model-View-Template* o Modelo-Vista-Plantilla) utilizado por el marco de trabajo Django. En este marco, el Modelo es la capa de acceso a la base de datos y contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene y las relaciones entre los datos. La Vista es la capa de la lógica de negocios incluye la lógica que accede al modelo y la delega a la plantilla apropiada. Es una conexión entre los modelos y las plantillas. La Plantilla es la capa de presentación y comprende las decisiones relacionadas a la presentación: cómo son mostradas sobre una página web u otro tipo de documento (Condori, 2016).

El patrón MTV 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 (Kaplan-Moss, 2016).

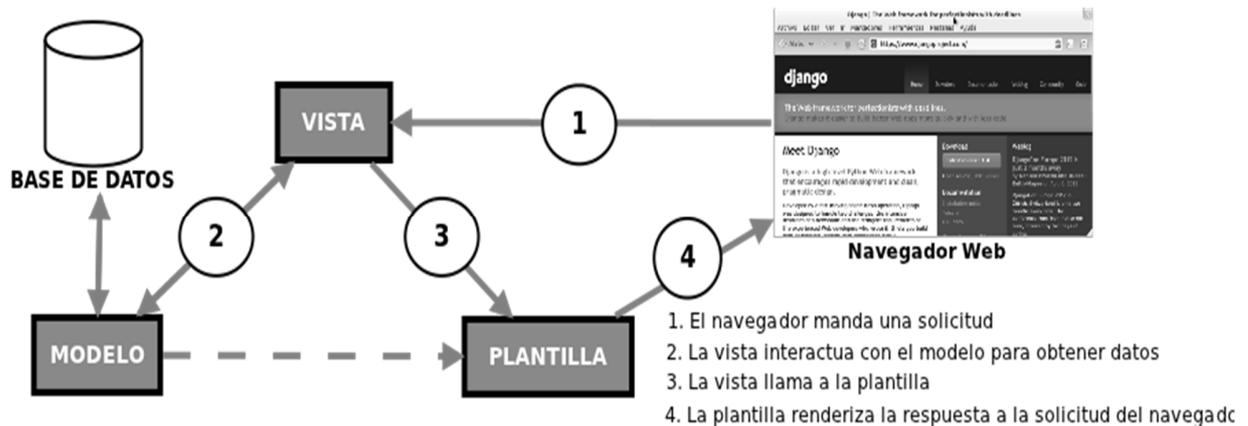


Figura 4 Patrón arquitectónico Modelo Vista Plantilla

Fuente: Django Framework de desarrollo web para perfeccionistas Basado en el Modelo MTV.

Para garantizar el cumplimiento del patrón arquitectónico anteriormente descrito, todos los ficheros de códigos y configuraciones necesarias para la propuesta de solución son definidos a partir de la propia estructura proporcionada por el marco de trabajo Django. Ver figura 5.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

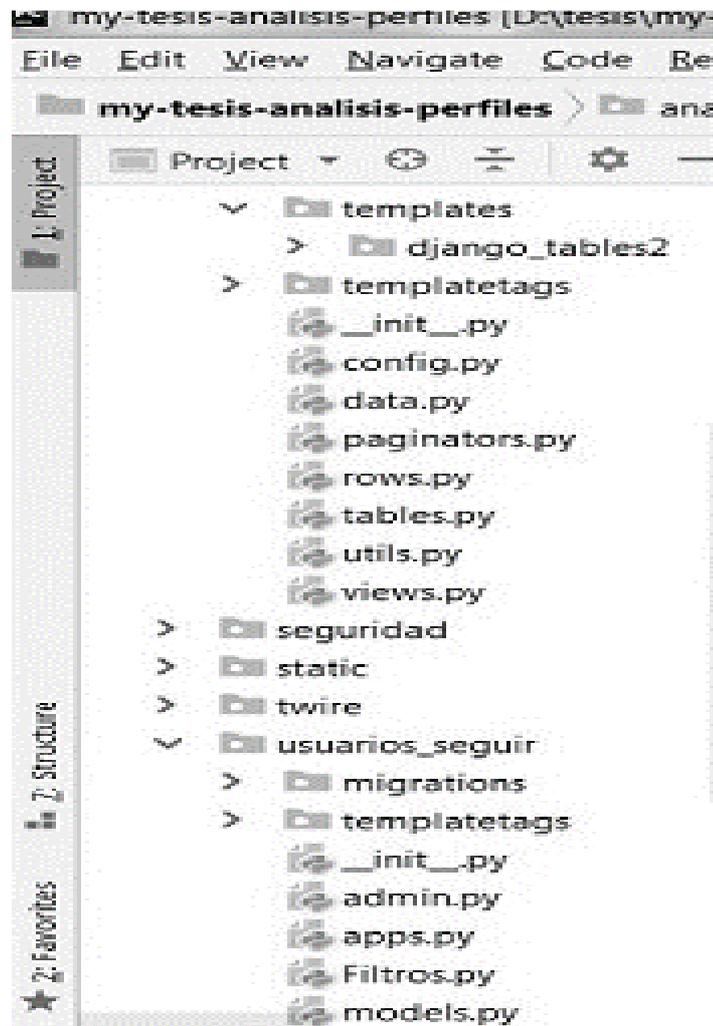


Figura 5 Arquitectura de la propuesta de solución
Fuente: Elaboración propia

2.3.5 Diagrama de Secuencia

Un diagrama de secuencia muestra la forma en que un grupo de objetos se comunican o interactúan entre sí a lo largo del tiempo. Consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada verticalmente (Baeza, 2017).

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

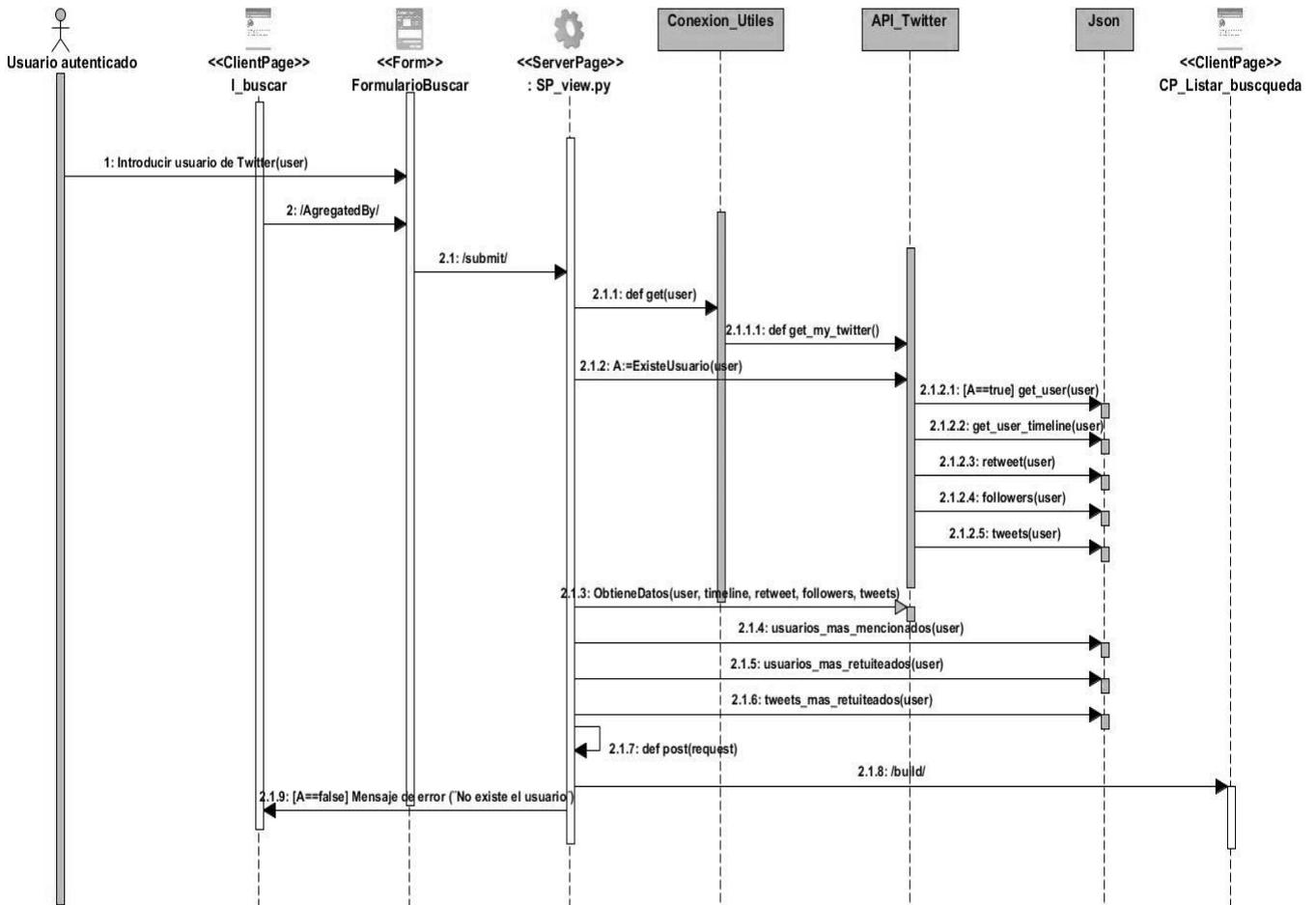


Figura 6 Diagrama de secuencia Mostrar datos más recientes de un usuario dado
Fuente: Elaboración propia

2.3.6 Diagrama de Despliegue

Un modelo de despliegue consiste en una representación estructural de la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue; definiendo a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo (Sarmiento, 2016).

A continuación, se muestra el diagrama de despliegue propuesto para el análisis de perfiles de usuarios en la red social Twitter. El mismo constituye una representación física de los nodos que componen el sistema y el reparto de los componentes en dichos nodos.

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

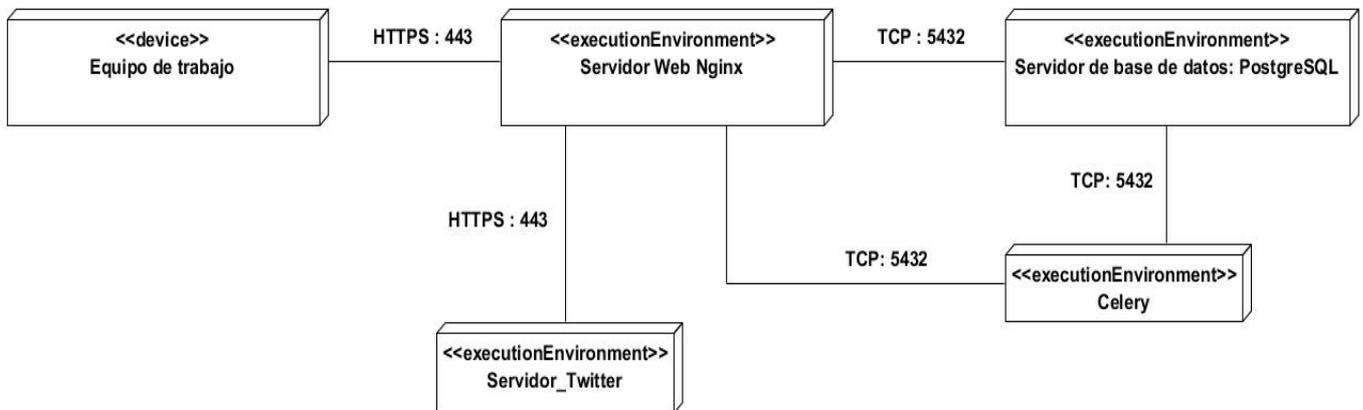


Figura 7 Diagrama de despliegue
Fuente: Elaboración propia

Descripción del diagrama de despliegue

Equipo de trabajo: Representa el conjunto de computadoras a través de las cuales los usuarios pueden consultar y actualizar la información que se encuentra en el Servidor Web Nginx. La comunicación entre las PC del Equipo de trabajo y el Servidor Web se establece utilizando el protocolo de comunicación HTTPS y el puerto 443.

Servidor Web Nginx: Es el nodo que funciona de intermediario entre el Equipo de trabajo y el servidor donde se encuentra la información. Este servidor web toma los datos, realiza sus funciones y presenta la información a las PC del Equipo de trabajo. La comunicación entre el Servidor Web y el servidor de base de datos se establece utilizando el protocolo de comunicación TCP y el puerto 5432.

Servidor de base de datos PostgreSQL: Es el nodo que contiene la información relacionada con los usuarios.

Servidor Twitter: Es el nodo que contiene la API de Twitter.

Celery: Es el nodo donde se ejecutan las tareas encargadas del intercambio de datos con Twitter de forma distribuida.

2.4 Conclusiones del capítulo

Después de realizado el análisis y diseño de la propuesta de solución, se puede concluir lo siguiente:

Análisis y diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter

1. La especificación de los requisitos funcionales y no funcionales del sistema, dieron paso a una mejor comprensión de las características que debe cumplir y el funcionamiento interno de la propuesta de solución.
2. La correcta identificación de los diferentes patrones de diseño y la arquitectura permitieron diseñar una solución considerando las buenas prácticas de desarrollo de *software*.
3. El diseño de diagramas de clases, facilitó la visión en cuanto a composición lógica del sistema.
4. La generación de todos los artefactos requeridos por la metodología de desarrollo Variación de AUP para la UCI, documentaron la solución propuesta facilitando su posterior implementación.

Capítulo 3: Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

Introducción

A partir de los artefactos generados en el capítulo anterior, se procede a la construcción del sistema. Se elabora el diagrama de componentes el cual representa cómo un sistema de software es dividido en componentes y las dependencias entre ellos. Para asegurar una mayor precisión y objetividad de la solución propuesta, se plasman los resultados obtenidos al aplicar una estrategia de pruebas con el fin de descubrir y corregir errores.

3.1 Implementación

La etapa de implementación del software es el proceso de concretar en términos de código las Historias de Usuarios. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente (Sommerville, 2011).

3.1.1 Diagrama de componentes

El diagrama de componentes permite visualizar la estructura de un sistema informático atendiendo a las partes que lo conforman. Cada componente debe ser tratado como una unidad de composición independiente e indispensable dentro de un sistema, pudiéndose encontrar dependencias entre estos. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables y binarios (Sommerville, 2011). Este diagrama provee una vista arquitectónica del sistema el cual permite lograr una mejor visualización de la implementación. Su realización posibilita tomar decisiones respecto a las tareas de implementación y los requisitos.

A continuación, se muestra el diagrama de componentes del sistema de análisis de perfiles de usuarios en la red social Twitter, organizado acorde a la arquitectura *Modelo-Vista-Plantilla* propuesta por el *framework* Django y descrita en el capítulo anterior. En la Plantilla se encuentran todos los componentes visuales con los que interactúa el usuario y define cómo se presentan los datos del modelo, la Vista contiene el archivo `views.py` y constituye una refinación de los datos del modelo los cuales se presentan en la plantilla y el Modelo es el componente que agrupa toda la información que es manejada por el sistema.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

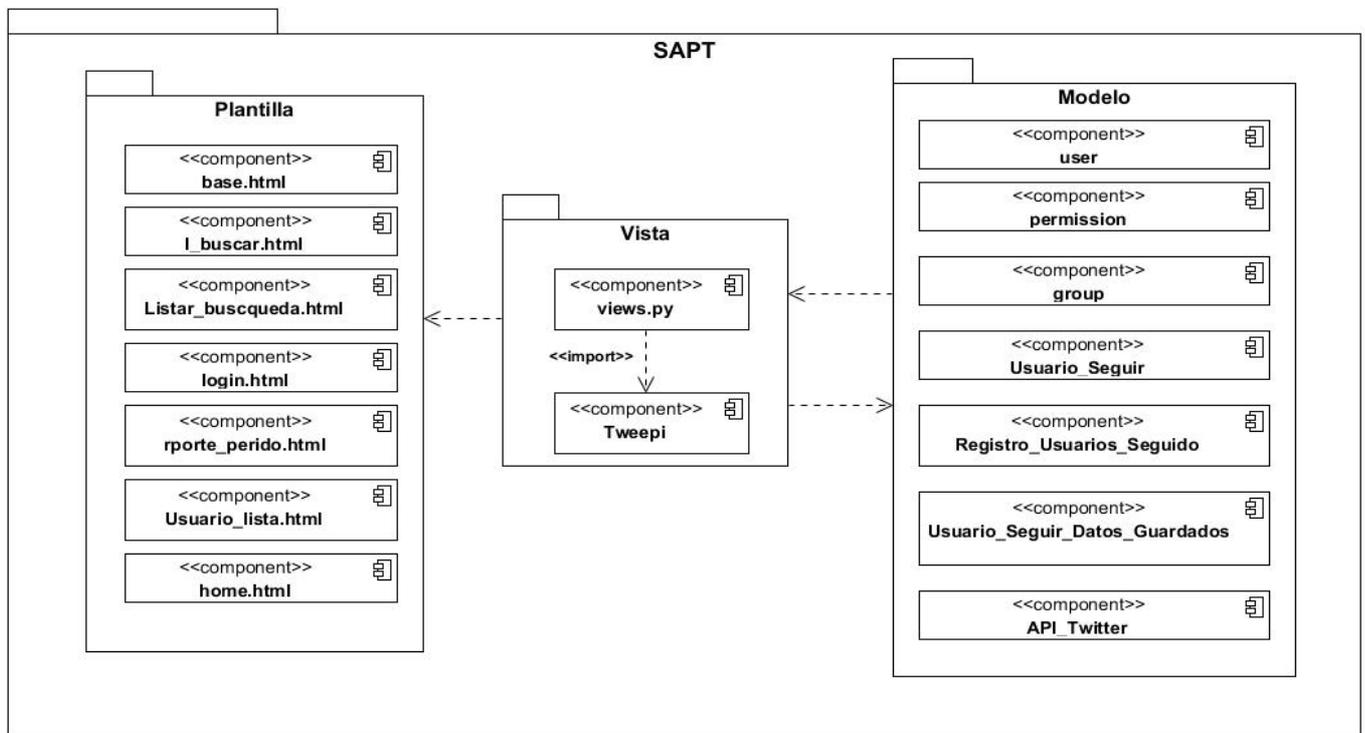


Figura 8 Diagrama de componentes
Fuente: Elaboración propia

3.1.2 Estándares de codificación

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. Ellos garantizan el mejoramiento de la comunicación en los equipos de desarrollo de software, reducen los errores de programación y mejoran la calidad del software. Todo esto repercute en la competitividad de las empresas de software y en la productividad de sus trabajadores porque se mejora su facilidad de mantenimiento teniendo un gran impacto en la reducción de los costos de mantenimiento (Vera, 2016).

Para la codificación de la propuesta de solución se utiliza el estándar de codificación para Python utilizado en el Universidad de las Ciencias Informáticas. En este documento se listan distintas convenciones utilizadas en el código Python comprendido en la librería estándar de la distribución principal de Python (Universidad de las Ciencias Informáticas, 2010).

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

A continuación se presenta cómo se evidencian los estándares de codificación en la implementación del sistema para el análisis de perfiles de usuarios en la red social Twitter.

Indentación

- Se utilizaron 4 espacios por cada nivel de indentación.

Tabuladores o espacios

- No se mezclaron tabuladores y espacios en la codificación. Las formas más populares de indentar en Python es utilizando sólo espacios o sólo tabuladores, el código indentado con una mezcla de tabuladores y espacios se reformateó y se usaron espacios exclusivamente.

Tamaño máximo de línea

- Todas las líneas están limitadas a un máximo de 79 caracteres.

Líneas en blanco

- Se separaron 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. Se usan líneas en blanco extras, de forma reservada para separar grupos de funciones relacionadas.

Convenciones de nombres

No se utilizaron los caracteres 'l' (letra ele minúscula), 'O' (letra o mayúscula) o 'I' (letra i mayúscula) como nombres de variables de un solo caracter debido a que en algunas fuentes, estos caracteres son indistinguibles de los numerales uno y cero.

- Se utilizó FormularioBuscar (palabras que comienzan con mayúsculas) para los nombres de las clases.
- Los nombres de funciones están en letras minúsculas, con palabras separadas mediante guiones bajos según sea necesario para mejorar la legibilidad.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

```
def crear_usuario(request, template_name='Modificar_usuario.html'):
    form = UserForm(request.POST or None)
    error = ''
    if form.is_valid() and request.user.has_perm('usuarios_seguir.add_usuario_seguir'):
        datos_user = form.cleaned_data
        try:
            usuarioTwire = get_my_twitter().get_user(datos_user['nombre'])
            if usuarioTwire.json['name'] == None:
                error = 'No existe el Usuario Buscado'
            return render(request, template_name, {'form': form.errors, 'error': error})
```

Figura 9 Ejemplificación del uso de los estándares de codificación
Fuente: Elaboración propia

Imports

- Los módulos importados se han colocados en distintas líneas.
- 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 importa una clase de un módulo, normalmente se realiza de la siguiente manera, por ejemplo:

```
from django_tables2 import RequestConfig
from .tabla import UserTable
from .Filtros import UserFilter
from twire.Utiles import get_my_twitter
from django.contrib.auth.decorators import login_required
```

Figura 10 Ejemplificación del uso de los estándares de codificación
Fuente: Elaboración propia

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

Nombres de paquetes y módulos:

- Los módulos tienen nombres cortos formados en su totalidad por letras minúsculas. Se utilizan guiones bajos en el nombre del módulo si mejora la legibilidad. Los paquetes Python también tienen nombres cortos formados por letras minúsculas.

Argumentos de funciones y métodos:

- Se usa siempre `'self'` como primer argumento de los métodos de instancia.

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 #.

Otras consideraciones

- Se rodearon siempre los siguientes operadores binarios con un espacio en cada lado: asignación (`=`), asignación aumentada (`+=`, `-=`, `*=`, `/=`), comparación (`==`, `<`, `!=`, `<>`, `<=`, `>=`).
- Se utilizaron espacios alrededor de los operadores aritméticos.

3.2 Validación de la propuesta de solución

Las pruebas de software son un conjunto de actividades que se realizan con el objetivo de detectar fallos y evaluar las características del software, regulan la ejecución de los proyectos y garantizan la calidad del producto desarrollado. Incluyen técnicas y métodos específicos del diseño de casos de prueba. Consisten en la verificación dinámica del comportamiento de un programa en un conjunto finito de casos de prueba, adecuadamente seleccionado de los posibles escenarios del sistema, para asegurarse que arroja el resultado definido en la especificación de requisitos (Pressman, 2010).

Luego del desarrollo de la propuesta de solución, se hace necesario verificar y validar el sistema implementado a través de una estrategia de pruebas que permitan comprobar el cumplimiento de las especificaciones del diseño y de la codificación e identificar los posibles errores cometidos.

La estrategia de pruebas definida por la autora para validar la propuesta de solución, está compuesta por tres (3) tipos de pruebas: las pruebas funcionales, la realización de las pruebas de rendimiento a través de

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

la herramienta Apache JMeter y las pruebas de seguridad con la herramienta Acunetix. Esto se realiza con el objetivo de comprobar el funcionamiento y probar la correcta implementación de los requisitos definidos.

3.2.1 Pruebas funcionales

Las pruebas funcionales se aplican a un software determinado, con el objetivo de verificar que las funcionalidades implementadas se desempeñen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Al conocer las funciones específicas que se le asignaron a una aplicación para su realización, pueden llevarse a cabo casos de pruebas que demuestren que cada función es completamente operativa mientras que al mismo tiempo se buscan errores en cada función (Pressman, 2010). Este tipo de pruebas es basado en requisitos y se vincula a los datos de entrada.

Con el objetivo de realizar este tipo de pruebas al sistema, se diseñó un conjunto de casos de pruebas, tres (3) en total (Ver Anexo 1), referentes a varias de las historias de usuarios obtenidas en la fase de diseño del capítulo anterior, pertenecientes a requisitos funcionales de prioridad alta para el cliente, también especificados en dicho capítulo. A continuación, se muestra solo uno (1) de los casos de pruebas mencionados.

*Tabla 4 Caso de prueba Mostrar datos más recientes de un usuario dado
Fuente: Elaboración propia*

| Entrada | Resultados | Condiciones |
|--|---|---|
| El usuario introduce el criterio de búsqueda (usuario) correcto. | El sistema muestra la interfaz de búsqueda con los datos correspondiente a la búsqueda realizada. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. |
| El usuario introduce el criterio de búsqueda (usuario) incorrecto. | El sistema muestra un mensaje informándole al usuario autenticado que el usuario buscado no existe. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. |
| El usuario autenticado deja el campo de búsqueda vacío. | El sistema le muestra un mensaje informando al usuario que el campo no puede estar vacío. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. |

Resultado de las pruebas

Como resultado final de las pruebas funcionales, se obtuvo, en una primera iteración, un total de nueve (9) no conformidades, divididas en cuatro (4) de ortografía, dos (2) de funcionalidad y tres (3) de validación.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

De estas, se resolvieron seis (6), y tres (3) quedaron pendientes. En una segunda iteración, se detectaron cinco (5) no conformidades, tres (3) de ellas pendientes de la iteración anterior y dos (2) nuevas de idioma, las cuales fueron resueltas en su totalidad. En una tercera iteración no se identifican nuevas no conformidades, obteniendo, de esta manera, resultados satisfactorios. La siguiente gráfica, muestra los resultados antes descritos:

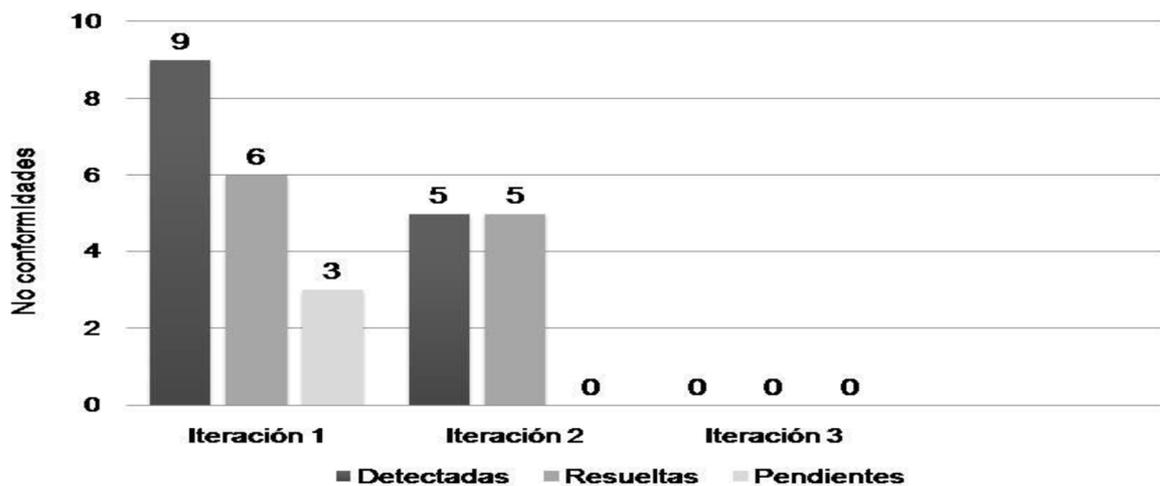


Figura 11 Resultado de las pruebas funcionales
Fuente: Elaboración propia

La realización de las pruebas funcionales posibilitó la identificación de errores en la aplicación, así como la corrección de cada uno de estos. Entre las faltas más comunes se encontraban la incorrecta validación de los datos y errores funcionales relacionados con la conexión entre la API de Twitter y el sistema.

3.2.2 Pruebas de rendimiento

Las pruebas de rendimiento se implementan y se ejecutan para caracterizar y evaluar las características relacionadas con el rendimiento del destino de la prueba, como los perfiles de tiempo, el flujo de ejecución, los tiempos de respuesta y la fiabilidad. A lo largo del ciclo de vida del desarrollo de software, se implementan varios tipos de prueba de rendimiento, cada una con un objetivo de prueba diferente (Brito, 2016).

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

Estas pruebas se centran en identificar y eliminar los cuellos de botella de rendimiento relacionados con la arquitectura. En las iteraciones de la construcción, los tipos adicionales de pruebas de rendimiento se utilizan para ajustar el software y el entorno (optimización de los recursos y el tiempo de respuesta), y para verificar que las aplicaciones y el sistema manejan de manera aceptable las condiciones de estrés y de carga elevada, como grandes números de transacciones, clientes y/o volúmenes de datos (Brito, 2016).

Entre los diferentes tipos de pruebas de rendimiento se incluyen los siguientes (Brito, 2016):

- **Pruebas de carga:** Mediante la ejecución de las pruebas de carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables, tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la aplicación.
- **Prueba de estrés:** Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de usuarios y/o procesos. Una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y procesos, que se verá reflejada en una óptima operación de negocio.

Para la realización de las pruebas de rendimiento se utiliza la herramienta JMeter, estas pruebas fueron realizadas en test de 50 hilos, los cuales simulan la cantidad de usuarios que acceden a las funcionalidades concurrentemente. Se tiene en cuenta el escenario donde se encuentra la aplicación y las condiciones de *hardware* y *software* de la computadora donde es ejecutado el muestreo; atendiendo a las siguientes características:

- **Hardware:** Memoria RAM de con 4GB, procesador Core i3 y una capacidad de disco duro de 500GB.
- **Software:** Python en su versión 3.6, Django en su versión 2.1 y PostgreSQL en su versión 11.

A continuación, se describen las variables que miden el resultado de las pruebas realizadas al sistema:

Muestra: Cantidad de peticiones realizadas por cada *URL*.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

Mediana: Tiempo promedio en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min: Tiempo mínimo que demora un hilo en acceder a la página.

Max: Tiempo máximo que demora un hilo en acceder a la página.

Línea 90%: Máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevó más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Tabla 5 Resultado de la prueba de rendimiento
Fuente: Elaboración propia

| Muestra | Media | Mediana | Línea 90% | Min | Max | % Error | Rend | Kb/s |
|---------|-------|---------|-----------|-----|------|---------|---------|------|
| 50 | 7092 | 537 | 3503 | 143 | 2103 | 0.0% | 4.5/sec | 40.3 |
| 50 | 6503 | 384 | 3433 | 97 | 1576 | 0.0% | 4.1/sec | 43.3 |
| 50 | 4295 | 261 | 1642 | 35 | 1019 | 0.0% | 4.2/sec | 44.0 |
| 50 | 7154 | 503 | 2311 | 36 | 2287 | 0.0% | 4.4/sec | 46.5 |

Se probó el sistema con un total de 200 peticiones enviadas en 50 hilos concurrentes, obteniéndose un tiempo promedio de respuesta de 6261 milisegundos (6.3 segundos aproximadamente), con un margen de error del 0.0 %.

3.2.3 Pruebas de seguridad

Las pruebas de seguridad intentan verificar que los mecanismos de protección que se construyen en un sistema lo protegerán realmente de cualquier irrupción inapropiada. Buscan medir la usabilidad, seguridad, eficiencia y portabilidad de los datos. Se diseñan para detectar las vulnerabilidades del entorno en el sistema (Pressman, 2010).

Para la realización de las pruebas de seguridad, se empleó la herramienta Acunetix Web Vulnerability Scanner, caracterizada en el epígrafe 1.5. En una primera iteración, se obtuvo un total de once (11) no conformidades, divididas en cinco (5) de nivel medio y seis (6) de nivel bajo. De las de nivel medio, se

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

destacó el uso del protocolo no seguro para el envío de datos. Las de nivel bajo estuvieron relacionadas con problemas para la protección contra ataques de fuerza bruta a la página de autenticación; así como directorios que pueden ser accesibles directamente sin pasar la autenticación. Esto fue resuelto utilizando siempre métodos POST en los procesos de autenticación, protegiendo las credenciales de acceso y asegurando un *login* persistente.

Todas estas deficiencias fueron corregidas en la primera iteración, y para una segunda, no se identificaron nuevas no conformidades, por lo cual se obtuvo finalmente una herramienta que cumple con los requisitos de seguridad definidos para la misma. Los resultados antes descritos, se muestran a continuación en la siguiente gráfica:

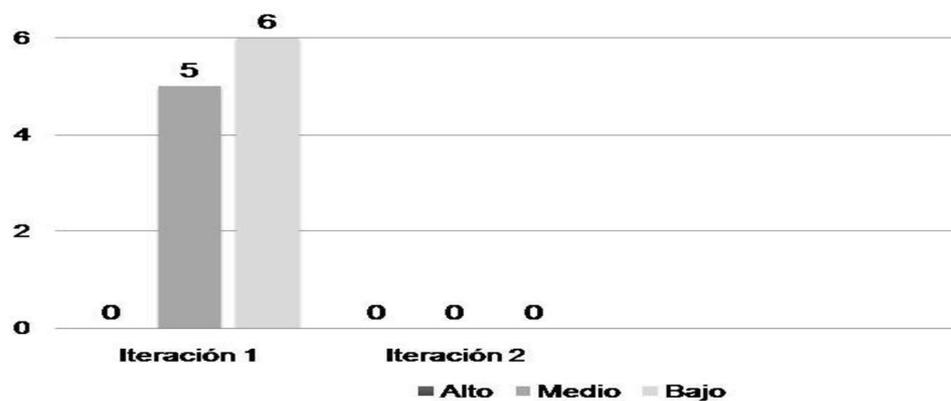


Figura 12 Resultado de las pruebas de seguridad
Fuente: Elaboración propia

3.3 Análisis de la aplicación de la técnica de ladov

Cuando se realiza una propuesta, es recomendable retroalimentarse con la opinión de los usuarios potenciales. Esta información es útil para conocer las debilidades de la propuesta y profundizar en sus fortalezas. Para valorar y/o constatar el nivel de aceptación y la aplicabilidad de la propuesta de solución se utiliza la técnica de ladov. Esta, es aplicada a una muestra de profesores y especialistas del Departamento de Operaciones Web y Análisis de Información (DOWAI) para medir mediante el Índice de Satisfacción Grupal (ISG) el nivel de aceptación del sistema desarrollado, por parte de los usuarios finales.

Mediante esta técnica se puede determinar, de forma indirecta, Índice de Satisfacción Grupal (ISG) de los individuos involucrados en el proceso que está siendo objeto de análisis. Para la aplicación de la técnica

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

de lado se define un cuestionario conformado por cinco preguntas, de las cuales tres son cerradas y dos abiertas (Ver Anexo 2). Según (Ramírez, 2018), las preguntas cerradas guardan una relación entre sí, que previamente no es de conocimiento por parte del sujeto al que se le aplica la técnica.

Estas tres preguntas se relacionan a través del "Cuadro Lógico de lado" el cual permite ubicar a cada encuestado en una escala de satisfacción, para luego calcular el ISG. Las respuestas a cada una de estas preguntas permiten determinar la posición de cada sujeto en la escala de satisfacción que toma valores desde 1 hasta 5, distribuido de la siguiente manera: 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definido o contradictorio, 4-Más insatisfecho que satisfecho y 5-Clara insatisfacción (Ramírez, 2018). El cuadro lógico utilizado en la investigación se muestra en la siguiente tabla.

*Tabla 6 Cuadro Lógico de lado utilizado
Fuente: Elaboración propia*

| ¿Qué tan satisfecho se siente con los resultados del sistema para el análisis de perfiles de usuarios en la red social Twitter? | ¿Estima provechosa la herramienta desarrollada para el trabajo que se realiza en el departamento? | | | | | | | | |
|---|--|-------|----|-------|-------|----|----|-------|----|
| | Sí | | | No sé | | | No | | |
| | ¿Utilizaría la herramienta desarrollada para el análisis de perfiles de usuarios en la red social Twitter? | | | | | | | | |
| | Sí | No sé | No | Sí | No sé | No | Sí | No sé | No |
| Clara satisfacción | 1 | 2 | 3 | 2 | 2 | 3 | 3 | 3 | 2 |
| Más satisfecho que insatisfecho | 2 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 2 |
| No definido o contradictorio | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Más insatisfecho que satisfecho | 2 | 3 | 3 | 3 | 4 | 4 | 3 | 3 | 4 |
| Clara insatisfacción | 2 | 3 | 3 | 3 | 4 | 4 | 3 | 3 | 5 |

Para medir el grado de satisfacción de los usuarios respecto al sistema desarrollado, se tomó como muestra a siete especialistas y profesores vinculados a DOWAI. La selección se realizó teniendo en cuenta la experiencia como analista en el trabajo con la red social Twitter (7 años como promedio) y el dominio de técnicas de análisis de redes sociales. Los resultados obtenidos para la satisfacción de forma individual se exponen a continuación.

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

$$ISG = \frac{A(+1)+B(+0.5)+C(0)+D(-0.5)+E(-1)}{N}$$

Las variables representan las cantidades de participantes agrupados por las escalas del índice de satisfacción individual. La cantidad de participantes que expresaron tener una clara satisfacción son representados por A, la cantidad que se sienten más satisfechos que insatisfechos se expresan mediante B, no definido o contradicción se evidencia mediante C, los que se sienten más insatisfechos que satisfechos mediante D, E es la cantidad de participantes que expresan una clara insatisfacción. El valor de N representa el total de participantes.

La escala numérica abarca el intervalo desde -1 hasta 1. Los valores que se encuentran comprendidos entre -1 y -0,5 indican insatisfacción; los comprendidos entre -0,49 y +0,49 evidencian contradicción y los que están entre 0,5 y 1 indican que existe satisfacción.

Resultados obtenidos

1. Los resultados obtenidos de la aplicación de la encuesta se presentan a continuación:

Tabla 7 Escala del Índice de Satisfacción Individual
Fuente: Elaboración propia

| Categorías grupales de satisfacción | Escala | Participantes en la escala |
|-------------------------------------|--------|----------------------------|
| Clara satisfacción | A | 4 |
| Más satisfecho que insatisfecho | B | 3 |
| No definido o contradictorio | C | 0 |
| Más insatisfecho que satisfecho | D | 0 |
| Clara insatisfacción | E | 0 |

2. Cálculo del Índice de Satisfacción Grupal

$$ISG = \frac{A(+1)+B(+0.5)}{N}$$

$$ISG = \frac{(4(+1)+3(+0.5))}{7}$$

$$ISG = 0.79$$

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

El formulario presentado a los participantes incluye dos preguntas abiertas, mostradas a continuación:

1. ¿Qué importancia le concede al sistema para el análisis de perfiles de usuarios en la red social Twitter?
2. ¿Qué aspectos a su juicio potencian o limitan el uso del sistema para el análisis de perfiles de usuarios en la red social Twitter?

Sobre la primera pregunta, los participantes manifestaron que resulta de gran importancia porque permite realizar estudios recientes de un perfil de usuario o por un período de tiempo determinado, además de brindar información en tiempo real de la evolución de un usuario específico en su relación con el entorno. Fue expuesta la importancia de las acciones realizadas por el usuario de forma activa sobre los *tweets*, como una expresión del interés o la sensibilidad del usuario ante la información implícita en el mismo. Además permite agilizar el trabajo y procesar mayores volúmenes de información en menos tiempo; ayuda a comprender el comportamiento de los usuarios a través del análisis de la información pública en sus perfiles, y describir estados de opinión, comportamiento y tratamiento acerca de determinados temas.

Sobre la segunda pregunta los participantes expresaron que la creciente necesidad de analizar la información que se publica en redes sociales, potencia el uso de esta herramienta. En cuanto a las limitaciones, consideran que todas se encuentran en el orden técnico, puede limitar su uso el despliegue del sistema en los servidores, su estabilidad y respaldo, y la inestabilidad de la conectividad a Internet. Igualmente, constituye una posible limitante, cualquier cambio que pueda realizar Twitter en su API de servicios.

3. Interpretación del resultado del ISG

El proceso de validación de la propuesta de solución mediante la aplicación de una encuesta a especialistas en el análisis de redes sociales Twitter y el análisis de los resultados obtenidos mediante la técnica de *ladov*, confirmó su posibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG= 0.79). Además, los criterios emitidos en las preguntas abiertas evidencian su satisfacción por la contribución del sistema, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad en función de su aporte práctico.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

3.4 Interfaz gráfica de usuario

La interfaz gráfica de usuario permite representar las acciones y la información disponible utilizando un conjunto de imágenes y objetos gráficos. Su función principal consiste en facilitar un entorno visual sencillo que permita la comunicación con el ordenador. A continuación se muestra la interfaz de la historia de usuario. Mostrar datos más recientes de un usuario dado, con el objetivo de visualizar el resultado obtenido una vez desarrollado el sistema.



Nombre: UCI_Cuba

Cantidad seguidores: 13704 Cantidad seguidos: 733 Cantidad tweet que emite: 27830

Usuarios más mencionados: Usuarios más retuiteados: Etiquetas más utilizadas:

| Usuario | Cantidad |
|-----------------------------|----------|
| UCI_Cuba | 33 |
| Miguel Diaz-Canel Bermúdez | 30 |
| Natalia | 15 |
| 18° CLAE • Venezuela 2019 • | 9 |
| MES | 9 |

| Usuario | Cantidad |
|---------------|----------|
| DiazCanelB | 19 |
| nataliauci16 | 12 |
| alicia_senra | 8 |
| dellylien1 | 6 |
| CortizoAndres | 5 |

| Etiqueta | Cantidad |
|------------------|----------|
| Cuba | 30 |
| SomosContinuidad | 11 |
| SomosCuba | 7 |
| DevCaUCI2019 | 5 |
| NoMásBloqueo | 5 |

Figura 13 Interfaz gráfica Mostrar datos más recientes de un usuario dado
Fuente: Elaboración propia

3.5 Conclusiones del capítulo

En este capítulo se han abordado los elementos de la implementación del Sistema para el análisis de perfiles de usuarios en la red social Twitter, así como las pruebas realizadas al mismo y los resultados obtenidos; lo cual permite arribar a las siguientes conclusiones:

- La elaboración del diagrama de componentes, facilitó la comprensión de la estructura general del sistema.
- El correcto uso de los estándares de codificación, permitió una mejor legibilidad y comprensión del código, facilitando su mantenimiento.

Implementación, pruebas y evaluación del sistema para el análisis de perfiles de usuarios en la red social Twitter

- La implementación del sistema permitió la obtención de una aplicación funcional que cumple con los requisitos definidos para la misma.
- La ejecución de la estrategia de pruebas especificada, permitió detectar y corregir deficiencias presentes en la solución y ofrecer una aplicación con mejor calidad, seguridad y rendimiento.
- Se evidenció la aceptación de la propuesta de solución a partir de la técnica de ladov por parte de los especialistas, la cual propició la evaluación satisfactoria del sistema para el análisis de perfiles de usuarios en la red social Twitter.

Conclusiones generales

La presente investigación permitió desarrollar un sistema para el análisis de perfiles de usuarios en la red social Twitter. Una vez respondidas todas las preguntas científicas planteadas, se concluye lo siguiente:

- El análisis de los referentes teóricos y de las aplicaciones informáticas estudiadas evidenció la necesidad de desarrollar un sistema informático para el análisis de perfiles de usuarios en la red social Twitter.
- La elaboración de los artefactos propuestos por la metodología de desarrollo Variación de AUP para la UCI y la identificación de requisitos permitieron una mayor comprensión en el diseño del sistema para el análisis de perfiles de usuarios en la red social Twitter.
- La correcta selección de las tecnologías de desarrollo, contribuyó a obtener información de forma centralizada del sistema para el análisis de los perfiles de usuarios en la red social Twitter.
- La aplicación de una estrategia de pruebas permitió validar el correcto funcionamiento del sistema desarrollado, simulando ambientes reales en diferentes entornos de ejecución.
- La validación científica a través de la técnica de ladov, evidenció el nivel de aceptación del sistema por parte de los especialistas encuestados.

Recomendaciones

Una vez concluida la presente investigación, se recomienda:

- ❖ Profundizar en el estudio del análisis de redes sociales basado en el actor, mediante el cálculo de métricas de centralidad con el objetivo de estimar la capacidad de influencia de un usuario sobre la base de las relaciones de mención, cita y *retweet*.
- ❖ Profundizar en el estudio del análisis de redes sociales para la evaluación del comportamiento y los hábitos de utilización de los usuarios de la red social Twitter.
- ❖ Investigar sobre la factibilidad de aplicar algoritmos de agrupamiento partiendo de las temáticas de interés de los usuarios y sus seguidores.
- ❖ Elaborar un índice de relevancia e influencia como método de evaluación del comportamiento de los usuarios en la red social Twitter.

Referencias Bibliográficas

- Abreu, Ramón. 2016.** Metodologías Tradicionales vs. Metodologías Ágiles. [En línea] 05 de 06 de 2016. [Citado el: 26 de Octubre de 2018.]
- Álvarez, Miguel Angel. 2017.** DesarrolloWeb.com--Lenguaje de Marcas de Hipertexto HTML 5. [En línea] 14 de 09 de 2017. [Citado el: 07 de 11 de 2018.]
- Andújar, Benisio. 2016.** Git-Control de versiones. [En línea] 2016. [Citado el: 27 de 02 de 2019.]
- Baeza, Vladimir Boltton. 2017.** UML Secuencia. [En línea] 18 de 08 de 2017. [Citado el: 13 de 02 de 2019.]
- Brito, Eugenio González. 2016.** Pruebas de rendimiento de software. [En línea] 29 de 08 de 2016. [Citado el: 21 de 03 de 2019.]
- Cabrera, Harold. 2015.** Red Social Twitter. [En línea] 18 de 04 de 2015. [Citado el: 15 de 10 de 2018.]
- Condori, Jose Luis. 2016.** *Python - DjangoFramework de desarrollo web para perfeccionistas* Basado en el Modelo MTV. 2016.
- Correaz, Asdrubal. 2017.** Celery: Distributed Task Queue. [En línea] 2017. [Citado el: 28 de 11 de 2018.]
- Creed, Manuel. 2017.** Marco de Desarrollo de la Junta de Andalucía. [En línea] 03 de 04 de 2017. [Citado el: 10 de 02 de 2019.]
- Fackensy, Acquia. 2016.** Standard search API Documentation. [En línea] 25 de 07 de 2016. [Citado el: 13 de 05 de 2019.]
- Fernández, Juan Manuel Pacheco. 2017.** Real Academia Española. [En línea] 2017. [Citado el: 07 de 10 de 2018.]
- Ferrer, Laurato. 2017.** *Propuesta de conceptualización de requisitos para proyectos software basados en formalismo de ingeniería de conocimiento.* 2017.
- Fintechs, Protobill. 2017.** API de Streaming en Twitter. [En línea] 19 de 01 de 2017. [Citado el: 20 de 04 de 2019.]

- Gasterol, Yoandri. 2015.** IDE PyCharm. [En línea] 17 de 09 de 2015. [Citado el: 15 de 11 de 2018.]
- Goebelbecker, Eric. 2017.** Rest API Tutorial. [En línea] 06 de 09 de 2017. [Citado el: 14 de 05 de 2019.]
- Gómez, Maria del Carmen. 2015.** *Notas del curso Base de Datos. Modelado de datos.* MÉXICO, D.F : s.n., 2015.
- Guerrero, Ronald. 2016.** Audit Your Web Security with Acunetix Vulnerability Scanner. [En línea] 2016. [Citado el: 22 de 03 de 2019.]
- Gutiérrez, Javier J. 2015.** Framework Django. [En línea] 2015. [Citado el: 14 de 10 de 2018.]
- Hugo, Jorge. 2014.** Analisis de perfiles de usuarios. [En línea] 28 de 11 de 2014. [Citado el: 17 de 10 de 2018.]
- Junco, Alejo. 2015.** PostgreSQL. [En línea] 2 de octubre de 2015. [Citado el: 25 de octubre de 2018.] <http://www.postgresql.org>.
- Kaplan-Moss, Adrian Holovaty y Jacob. 2016.** *The Definitive Guide to django.* 2016.
- Kerlinger, Fred. 2016.** *Perfiles de usuario en redes sociales.* México D.F : s.n., 2016.
- Kuzko, Richard. 2016.** *Lenguajes de programación.* 2016.
- Loisel, Jérôme. 2016.** JMeter Tutorial for Beginners. [En línea] 07 de 08 de 2016. [Citado el: 18 de 03 de 2019.]
- Magán, Paloma Ruiz. 2018.** *Desarrollo de una plataforma para la gestión de Twitter y el análisis de su repercusión.* 2018.
- Martinez, Alberto. 2015.** Las redes sociales en Internet. [En línea] 18 de 12 de 2015. [Citado el: 10 de 11 de 2018.]
- Mateo, Sergi. 2016.** Manual práctico de benchmarking. [En línea] 15 de 07 de 2016. [Citado el: 29 de 10 de 2018.]
- Montero, Richard. 2014.** Perfil de usuario en los sitios de redes sociales. [En línea] 25 de 06 de 2014. [Citado el: 08 de 12 de 2018.]

- Naramore, Elizabeth. 2016.** Servidor de Aplicaciones Web Nginx. [En línea] 2016. [Citado el: 21 de 10 de 2018.]
- Noguera, Bulmaro. 2016.** Culturación. Ventajas de utilizar HTML 5. [En línea] 16 de 08 de 2016. [Citado el: 2018 de 10 de 25.]
- Nuñez, Robert. 2014.** Importancia de las redes sociales en la sociedad actual. [En línea] 24 de junio de 2014. [Citado el: 05 de octubre de 2018.]
- Pressman, Roger. 2010.** *Ingeniería de Software. Un enfoque práctico.* 2010.
- Prist, Fernando. 2016.** Twitter-libraries. [En línea] 14 de 02 de 2016. [Citado el: 3 de 10 de 2018.]
- Python Social Auth's. 2018.** Welcome to Python Social Auth's documentation! [En línea] 19 de 02 de 2018. [Citado el: 15 de 05 de 2019.]
- Quimera, Felipe. 2015.** Patrones de diseño: qué son y por qué debes usarlos. [En línea] 2015. [Citado el: 18 de 02 de 2019.] <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos..>
- Ramírez, Adrián. 2015.** Reglas de calidad para la codificación estandarizada. 2015.
- Ramírez, José Gabriel Espinosa. 2018.** *Sistema para la detección de roles sobre colecciones de tuits. Maestría en Ciencias Avanzadas.* 2018.
- Ramírez, Marlon. 2018.** Aplicación de la técnica de ladov. 2018.
- Ramos, César Vinuesa. 2015.** *El auge y la importancia de las redes sociales en el mundo actual.* 2015.
- Rodríguez, Alejandro González. 2016.** *Librerías Twitter para Python.* 2016.
- Rodríguez, Tamara. 2015.** Metodología de desarrollo para la actividad productiva de la UCI. Universidad de las Ciencias Informáticas. La Habana. Cuba. 2015.
- Romero, Abel. 2017.** Mozilla Developer Network--CSS, Learn to style HTML using. [En línea] 23 de 11 de 2017. [Citado el: 25 de 05 de 2018.]
- Salazar, Patricia. 2016.** *El perfil de usuario de información.* 2016.

- Salinas, Adriana. 2017.** Origen e historia de la red social Twitter. [En línea] 24 de 10 de 2017. [Citado el: 15 de 05 de 2018.]
- Sarmiento, Johana. 2016.** Vision General de los diagramas de despliegue. [En línea] 22 de 03 de 2016. [Citado el: 14 de 01 de 2019.]
- Schmuller, Joseph. 2015.** *Visual Paradigm para UML*. 2015.
- Silega, Noris. 2017.** ¿Qué es la técnica de ladov? [En línea] 16 de 02 de 2017. [Citado el: 16 de 03 de 2019.]
- Sommerville, Ian. 2011.** *Ingeniería de Software*. 2011.
- Thornton, J. 2014.** Bootstrap Manual Oficial. [En línea] 08 de 05 de 2014. [Citado el: 23 de 10 de 2018.]
- Universidad de las Ciencias Informáticas. 2010.** *Programa de mejora. Estándar de codificación para Python*. 2010.
- Urrutia, Angélica. 2015.** Análisis de Información de Redes Sociales (Twitter). [En línea] 17 de 09 de 2015. [Citado el: 07 de 12 de 2018.]
- Velázquez, Luis Romero. 2015.** *Análisis y diseño orientado a objetos*. 2015.
- Vera, D., DANILO, E., V.B., Ramírez, A. del Á., BARRIOS, M. y ANTONIO, J.,. 2016.** *Reglas de calidad para la codificación estandarizada en Lenguaje Python: una propuesta para la enseñanza a nivel superior. Proyectos institucionales y de vinculación, vol. 3, no. 6, pp. 155–175*. 2016.

Anexos

Anexo 1: Pruebas funcionales. Casos de pruebas

*Tabla 8 Caso de prueba Autenticar usuario
Fuente: Elaboración propia*

| Entrada | Resultados | Condiciones |
|---|---|---|
| El usuario introduce los datos (usuario, contraseña) correctos para autenticarse. | El sistema reenvía al usuario a la interfaz predeterminada para realizar la búsqueda. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. |
| El usuario omite algunos de los datos (usuario, contraseña) 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 exista conexión para que el sistema pueda conectarse con la red social Twitter. |
| El usuario introduce algunos datos (usuario, contraseña) incorrectos para autenticarse. | El sistema muestra un mensaje de error informándole al usuario que alguno de los datos está incorrecto. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. |

*Tabla 9 Caso de prueba Mostrar datos de un usuario seguido en un intervalo de tiempo
Fuente: Elaboración propia*

| Entrada | Resultados | Condiciones |
|--|---|--|
| El usuario introduce el criterio de búsqueda (usuario) correcto. | El sistema muestra la interfaz de búsqueda con los datos correspondiente a la búsqueda realizada. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. Que la BD esté conectada. |
| El usuario introduce el criterio de búsqueda (usuario) incorrecto. | El sistema muestra un mensaje informándole al usuario autenticado que el usuario buscado no existe. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. Que la BD esté conectada. |
| El usuario autenticado deja el campo de búsqueda (usuario) vacío. | El sistema le muestra un mensaje informando al usuario que el campo no puede estar vacío. | Que exista conexión para que el sistema pueda conectarse con la red social Twitter. Que la BD esté conectada. |

