# DESARROLLO Y APLICACIÓN DE UNA HERRAMIENTA DE EXTRACCIÓN Y ALMACENAMIENTO DE DATOS DE TWITTER A UN CONTEXTO SOCIAL DE VIOLENCIA POLÍTICA


**UNIVERSIDAD CATÓLICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**



**BOGOTÁ D.C**

**2017**

# DESARROLLO Y APLICACIÓN DE UNA HERRAMIENTA DE EXTRACCIÓN Y ALMACENAMIENTO DE DATOS DE TWITTER A UN CONTEXTO SOCIAL DE VIOLENCIA POLÍTICA

**JOSÉ CAMILO BARRIGA MARIÑO**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE**

**INGENIERO DE SISTEMAS**

**DIRECTOR**

**DIEGO ALBERTO RINCÓN YÁÑEZ MCSc**

**INGENIERO DE SISTEMAS**

**UNIVERSIDAD CATÓLICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**BOGOTÁ D.C**

**2017**

**NOTA DE ACEPTACIÓN**

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de Ingenieros de Sistemas.

_____

Jurado

_____

Diego Alberto Rincón
Director

_____

Revisor Metodológico.

Fecha de Entrega

## AGRADECIMIENTOS

Agradezco a Dios por permitirme culminar con este proceso tan importante en mi vida.

A mi asesor de proyecto de grado el ingeniero Diego Alberto Rincón, ya que sin su gran apoyo y su amplio conocimiento, la culminación de este proyecto no hubiera sido posible.

A cada uno de los docentes que de buena forma aportaron su conocimiento y tiempo en mi ciclo académico por la Universidad Católica de Colombia.

A mi familia quien ha sido el apoyo primordial para poder culminar cada semestre además de ser gran parte de mi motivación cada día.

A cada uno de mis compañeros con los cuales aprendí cada día de este proceso y con los que compartí el esfuerzo aplicado en cada objetivo de cada asignatura.

Y a cada una de las personas que me brindó su apoyo de alguna manera en pro de mi bienestar durante esta importante etapa de mi vida.

## TABLA DE CONTENIDO

## TABLA DE ILUSTRACIONES

## TABLA DE ANEXOS

## ABSTRACT

This project was oriented to the construction of a web tool for extracting and storing data from the twitter social network, which will allow future with the support of external or integrated software, establish a statistical analysis of these data, focused on the need of the user. In this case, the functionalities of the tool are applied to a social context focused on the measurement of political violence on twitter. For its realization, theoretical bases were integrated focused on concepts related to BigData, Storage and data mining, with emphasis on data extraction, storage and curing processes. In addition, a software development process was carried out in which the PXP methodology was applied, which is an adaptation of the extreme programming methodology focused on the development carried out by a single programmer. This enabled the tool to be implemented quickly and to generate a set of unit tests with which the correct functioning of the software is identified. The connectivity of the tool with the social network specified involved the use of technologies such as programming interfaces and web services, which allowed processes linked to data extraction or web mining to be performed correctly and thus generate an information context open to analysis. Information that regardless of the limitations with which the tool was developed, exceeds in value of utility and variety the information offered by different tools of the market. The system extracts and stores the data of the required accounts, which are previously entered into the system, and allows the user to manage a classification focused on the social context treated, later this information provides a quality environment for the statistical analysis required and that opens the doors to the construction of different solutions.



**Keywords**: BigData, Data Mining, Data Storage, Extraction, Software Development.

## RESUMEN

Este proyecto se orientó a la construcción de una herramienta web para la extracción y almacenamiento de datos de la red social twitter, la cual permitirá a futuro con apoyo de un software externo o integrado, establecer un análisis estadístico de estos datos, enfocado en la necesidad del usuario. En este caso, las funcionalidades de la herramienta se aplican a un contexto social enfocado en la medición de violencia política en twitter. Para su realización se integraron bases teóricas enfocadas en conceptos relacionados con BigData, Almacenamiento y minería de datos, haciendo énfasis en los procesos de extracción, almacenamiento y curación de datos. Así mismo, se realizó un proceso de desarrollo de software en el cual se aplicó la metodología PXP, la cual es una adaptación de la metodología de programación extrema enfocada al desarrollo llevado a cabo por un solo programador. La cual permitió de forma ágil implementar la herramienta y generar como resultado, un set de pruebas unitarias con las cuales se identifica el correcto funcionamiento del software. La conectividad de la herramienta con la red social especificada implicó el uso de tecnologías como interfaces de programación y servicios web, las cuales permitieron que los procesos ligados a la extracción de datos o minería web se realizaran de forma correcta y así generar un contexto de información abierta al análisis. Información que sin importar las limitaciones con las cuales fue desarrollada la herramienta, supera en valor de utilidad y variedad a la información que ofrecen diferentes herramientas del mercado. El sistema extrae y almacena los datos de las cuentas requeridas las cuales son previamente ingresadas al sistema y permite al usuario gestionar una clasificación enfocada al contexto social tratado, posteriormente esta información provee un ambiente de calidad para el análisis estadístico requerido y que abre las puertas a la construcción de diferentes soluciones.



**Palabras Claves:** Almacenamiento de datos, BigData, Desarrollo de software, Extracción, Minería de datos.

# INTRODUCCIÓN

La llegada de las redes sociales a la web impactó al mundo de forma única, el hecho de poder establecer una comunicación de nivel global en cuestión de segundos, con la única condición de poseer acceso a internet y la creación de una cuenta (la cual es una acción gratuita en las redes más populares), genera que su uso sea constante y masivo. En consecuencia, el flujo de datos en las comunicaciones por chat, publicaciones y comentarios obtiene un volumen inagotable como lo plantean en (Logicalis, 2015), volumen que impulsa la idea de que estos datos pueden ser de utilidad y facilitar tareas fundamentales.

La información expuesta por los propietarios de cada cuenta, permite que se reproduzca un reflejo tanto de sus pensamientos como de estado de ánimo, lo que permite que pueda realizarse un análisis de sentimiento con el fin de encontrar patrones de comportamiento en diferentes situaciones, o respecto a diferentes temas. Esto suele ser aprovechado por la industria, marketing e inclusive por organizaciones del estado en el planteamiento de estrategias de comunicación y captación de público.

El análisis de redes sociales según (García García & Rodríguez, 2017) ha irrumpido en muchas ciencias sociales en los últimos veinte años como una nueva herramienta de análisis de realidad social. Al centrarse en las relaciones de los individuos (o grupos de individuos) y no en las características de los mismos (raza, edad, ingresos, educación, entre otras), ha sido capaz de abordar algunos temas con un éxito insospechado.

Pero ¿Cuál sería el paso inicial que se debe realizar para poder analizar estos datos y obtener un beneficio? ¿Cómo hacer de estos datos una estructura definida la cual se pueda operar?

Este documento tiene como objetivo definir formalmente el desarrollo de una herramienta encargada de responder a estas preguntas, una herramienta funcionalmente enfocada en la extracción de datos en la red social twitter y su posterior almacenamiento. Todo con el fin de construir una base de datos que pueda ser operada con facilidad, en la cual el contenido se pueda definir como limpio y adecuado para un posterior análisis.

Este desarrollo está acogido por un proyecto padre el cual tiene un enfoque psicológico y busca plantear la efectividad de un protocolo de re experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política. La primera fase de este proyecto padre requiere la construcción de la herramienta definida, que a diferencia de algunos desarrollos ya

existentes, evitará el sesgo de información y se adaptará a los objetivos específicos del proyecto social.

El proceso de desarrollo implicado comprende varias etapas en las cuales, a través de un estándar definido, se construye la estructura requerida para el funcionamiento óptimo de la herramienta, esto involucra el uso de tecnologías propias del desarrollo de software, las cuales permitirán definir el diseño de la herramienta y consolidar los conceptos en la implementación general.

Sin embargo, con el fin de obtener el resultado deseado, se debe realizar una previa investigación y estudio acerca de las diferentes metodologías, conceptos, antecedentes y demás información relevante para el desarrollo del proyecto. Esta información de carácter formativo se expone en el presente documento, relacionado al punto de vista de nuestro entorno, lo que permite darle un contexto de integración a esta información y captar él porque de la necesidad en la construcción de este tipo de herramientas.

# 1. GENERALIDADES

## 1.1 ANTECEDENTES

Es algo común pensar que desde que las redes sociales dieron su aparición y con el impacto que siempre han causado, diferentes entidades y organizaciones se han podido interesar sobre la información que se expone en estos medios de comunicación por parte de las personas. Desde el año 1994 el mundo conoció el concepto de redes sociales a través del lanzamiento de GeoCities una red que permitía a los usuarios crear sus propias páginas web y alojarlas en determinados sectores.(Marketing Directo, 2011), este solo fue el inicio de una evolución de las redes sociales que llevo con ella el incremento de información y por lo tanto la labor de recolectar la información generada por estas redes se empezó a llevar a cabo.

Con esta información y el apoyo de la investigación realizada en la primer fase del presente proyecto sobre herramientas de extracción y almacenamiento de datos en las redes sociales, podemos definir que desde el comienzo y en el desarrollo de este milenio, al igual que las redes sociales han evolucionado, el número de herramientas para recolectar datos de ellas han incrementado y estas han mejorado con la llegada de nuevas tecnologías.

Dentro de las diferentes herramientas investigadas existen algunas enfocadas a la extracción de cierta información, la mayoría para apoyar la toma de decisiones de las organizaciones o clientes interesados. Entre algunas se encuentran:

- **Hootsuite.**

Es una aplicación web y móvil lanzada en el 2008, la cual se origina a base de la necesidad de diferentes organizaciones de adaptarse tecnológicamente al entorno actual y de aprovechar las diferentes fuentes de información en la red, para gestionar de una mejor manera sus servicios y evolucionar, a través de diferentes análisis aplicados a los datos recolectados de diferentes fuentes de información en la web.

Esta herramienta gestiona redes sociales por parte de personas u organizaciones, es una de las más reconocidas a nivel mundial por sus diferentes recursos de análisis y se enfoca en el marketing global aportando en el hallazgo de clientes potenciales como se especifica en su página principal (Hootsuite, 2017). Permite utilizar, entre otras, las siguientes redes sociales: Facebook, Twitter, LinkedIn, GooglePlus, Instagram, YouTube, Foursquare y adicionalmente está integrada con Google Analytics y Facebook Insights. Actualmente posee aproximadamente 2 millones de usuarios registrados.

- **Ucl's socialstorm.**

En el año 2014, Wood, Zheludev y Treleaven elaboran una plataforma diseñada para minar datos provenientes de Twitter, Facebook, fuentes RSS (un formato XML para compartir contenido en la web.) y blogs a la cual le dieron el nombre de UCL's SocialSTORM, Además ejecuta un monitoreo de datos puestos a disposición a través del tiempo.

Para sacar provecho de la gran cantidad de datos actualmente disponibles en la web, para los fines de la investigación académica, la Universidad de Londres, University College London (UCL) ha construido, y continúa desarrollando, un motor de datos de medios de comunicación social que soporta el scraping y análisis de una amplia gama de datos, los cuales son almacenados con una latencia inferior a 1 segundo lo que permite 4000 entradas a la base de datos cada segundo.

Esta plataforma como ilustra (R. Wood, Zheludev, & Treleaven, 2012), incluye instalaciones para ejecutar modelos de simulación en la que los datos permiten la identificación de las tendencias cambiantes, sentimientos generales y la propagación de historia.

- **Polyphonet.**

El sistema de extracción de datos en redes sociales llamado POLYPHONET, basándose en (Matsuo et al., 2007), emplea varias técnicas avanzadas para extraer las relaciones de las personas, a detectar grupos de personas, y para obtener palabras clave. En primer lugar, se reducen los métodos relacionados en Pseudocódigos simples usando Google para que se puedan construir sistemas integrados.

En segundo lugar, desarrollan novedosos algoritmos para la minería de redes sociales, tales como las relaciones de clasificar en categorías para hacer la extracción escalable, y para obtener y utilizar las relaciones de persona a palabra. Cada módulo que se implementa en POLYPHONET, se ha utilizado en cuatro conferencias académicas, cada una con más de 500 Participantes.

- **Automap.**

Es una herramienta enfocada en la extracción de información textual (minería de texto) elaborada por CASOS en Carnegie Mellon. Esta herramienta utiliza diferentes métodos de SNA (Análisis de redes sociales) y permite extraer datos de varios tipos.

Esta herramienta se elaboró en Java 1.7 de forma que su funcionamiento cubra tanto el front end como el back end, por lo tanto, en cualquiera de las dos caras se puede hacer uso de la herramienta.

Los diferentes datos que puede extraer se definen en (Eisenberg et al., 2010) como datos de contenido analítico (palabras y frecuencias), datos de red semántica (la red de conceptos), los datos de meta-red (la clasificación cruzada de los conceptos en su categoría ontológica como personas, lugares y cosas y las conexiones entre estos conceptos clasificadas), y los datos de confianza (actitudes, creencias).

## 1.2 PLANTEAMIENTO DEL PROBLEMA

Cada día que transcurre, la cantidad de información generada a través de las redes sociales aumenta drásticamente. Este medio de comunicación permite tener un enfoque característico de las múltiples cuentas asociadas, donde cada una de ellas puede pertenecer a una entidad reconocida en el medio social por su labor o posición en diferentes áreas (espectáculo, religión, política, etc.) y donde la información compartida permite que exista una conexión entre las personas y sus pensamientos, en este caso plasmados en cada publicación.

Dicha información tiene un crecimiento exponencial, y para poder establecer un análisis de estos datos, antes se debe realizar una debida recolección y almacenamiento de estos datos. Procesos en los cuales para realizarse de una forma adecuada deben incluir el tratamiento de los datos en sus diferentes estructuras, para posteriormente generar una composición fácil de almacenar a través de métodos específicos y ligados al tipo de almacenamiento.

Como producto de la investigación de los antecedentes del proyecto, se identifica que la mayoría de herramientas de extracción y almacenamiento de datos de redes sociales pertenecientes al mercado actual, se centran en un contexto específico, generalmente enfocado al marketing y los negocios. Esto no permite que se definan como un conjunto de herramientas que pueden atacar la mayoría de problemáticas que se presenten incluyendo el factor social, por otro lado aquellas herramientas que poseen las cualidades para hacerlo, son herramientas que para poder ser utilizadas implican un costo de adquisición superior y algunas limitaciones en su uso.

Luego de identificar estos factores y si se pretendiera realizar un tipo de análisis, estadístico, de varianza con enfoque al análisis de sentimientos y con carácter predictivo, que involucre los datos generados por diferentes cuentas en una red social tan influyente como lo es twitter, teniendo en cuenta la necesidad de que la

obtención de estos datos pueda generarse de una forma sencilla e implique bajos costos monetarios, además de tomar como contexto para la aplicación funcional de la herramienta la problemática social referente a la violencia política que se genera a través de este medio de comunicación ¿A través de que medio podría recolectarse y almacenarse dicha información para un posterior análisis?

## 1.3 DESCRIPCIÓN DEL PROBLEMA

Hoy en día la mayoría de las grandes fuentes de información como lo son las redes sociales, se ven mayormente aprovechadas por organizaciones comerciales, las cuales generan análisis de estos datos para apoyar la toma de decisiones con vista a un progreso o beneficio propio. Las herramientas utilizadas para realizar la extracción y almacenamiento de estos datos, se limitan en su mayoría a un conjunto de datos enfocados al marketing y los negocios.

No es igual de notable el aporte en el desarrollo y uso de las herramientas de extracción y almacenamiento de datos de redes sociales para un contexto general. Un contexto que implique un bajo costo por su adquisición y que permita ofrecer soluciones al mundo basadas en problemáticas sociales, ecológicas o científicas.

La mayoría de herramientas de bajo costo poseen un conjunto reducido de funcionalidades que no permiten establecer un proceso completo de extracción y almacenamiento de datos, algunas veces los datos extraídos no poseen la calidad suficiente para llevar a cabo un análisis concreto. Por otro lado los costos de licencia de las herramientas reconocidas en el mercado, pueden ser elevados para el investigador o la entidad que desee adquirir una herramienta con estas cualidades. Como ejemplo de herramienta de extracción de datos de la red social Twitter, **Insights for Twitter** es un producto de software de IBM Social Media Analytics, el cual ofrece un servicio de extracción y almacenamiento de tweets, además de generar búsquedas a través de un conjunto de parámetros de consulta y palabras clave. Este producto tiene un plan básico el cual permite extraer como máximo 1 millón de tweets por cuenta por un costo de 2.000,00 USD mensuales (IBM Bluemix, 2015), lo que equivale a aproximadamente 6’000.000 COP. Como se puede apreciar el costo de la herramienta aumentará según el tiempo por el cual se utilice.

De esta manera se apoya la formulación y desarrollo de la herramienta expuesta en el presente proyecto la cual permitirá de una forma sencilla extraer una cantidad de datos que se puedan utilizar en diferentes análisis de temas específicos según la necesidad del usuario y el planteamiento de la investigación a realizar.

Como desarrollo inicial se otorga para la aplicación de la herramienta, una problemática social de nivel nacional, donde se busca obtener una solución concreta a través de datos de una red social especifica y la interacción de varios componentes, donde diferentes áreas y conceptos de investigación se integran para enmarcar un solo proyecto.

En un entorno donde la sociedad se ve afectada constantemente por decisiones de nivel político y la violencia generada a través de los diferentes medios de comunicación, el área de psicología y la ingeniería de sistemas buscan desde la aplicación de sus ramas del conocimiento generar una solución que contribuya al bienestar social, dadas las diferentes experiencias que lo han perjudicado en el contexto expuesto.

Para ello, la facultad de psicología realiza un estudio cuantitativo con dos fases. La primera, tiene como propósito identificar los estilos lingüísticos asociados al reconocimiento del conflicto, situaciones pre-traumáticas y creencias identitarias en el contexto de violencia política en Colombia mediante el análisis estadístico de los comunicados conjuntos difundidos por la mesa de conversaciones en el marco del proceso de diálogo y negociación entre las Fuerzas Armadas Revolucionarias de Colombia (Farc-Ep) y el Gobierno colombiano y las conversaciones de distintos actores del contexto político colombiano durante el año 2016 mediante mensajes textuales publicados en la red social local Twitter.

En esta fase, la ingeniería de sistemas se hace partícipe a través de la presente investigación e implementación de una herramienta con la función de extraer y almacenar datos de la red social propuesta, la cual se lleva a cabo con el fin de poder recuperar datos de forma ágil acerca de alguna entidad, que puede representarse como una unidad empresarial, personas reconocidas en un área específica o cualquier propietario de una cuenta en la red social, para posteriormente ejecutar un proceso de análisis sobre los datos extraídos. Esta herramienta se define como un producto semi-dependiente que aunque apoye el contexto social enmarcado por el proyecto de la facultad de psicología, tiene la capacidad para llevar sus funciones a cualquier problemática.

Claramente, para que se pueda llevar a cabo un análisis correcto y confiable de estos datos, el proceso de extracción y almacenamiento debe ser implementado de tal forma que no exista una corrupción en la información recuperada, es decir que a cada tarea, como por ejemplo (la limpieza de los datos), esté ligada a un control de calidad y verificación que consientan la viabilidad de su uso.

Visto desde una perspectiva un poco más global y con visión futura del proyecto, la visión consiste en generar diferentes soluciones a las necesidades de los diferentes campos del conocimiento e industria en los que la herramienta pueda aportar. Soluciones provenientes de los análisis aplicados a los datos extraídos de la red social y que permitan producir información valiosa en la toma de decisiones aplicando un carácter predictivo para un objetivo definido.

La segunda fase del proyecto general, tiene como objetivo establecer la eficacia de un protocolo de re-experimentación emocional y mindfulness mediante un diseño factorial con grupo control con medidas pre y pos prueba con 105 participantes (expuestos a situaciones traumáticas en un contexto de violencia política).

La primer fase del proyecto permitirá identificar las diferencias del tratamiento entre grupos mediante diferentes análisis de varianza (univariados y multivariados) y, el segundo, permitirá comparar los cambios psicológicos (cognitivos y emocionales) expresados a través del lenguaje mediante el análisis de estilos lingüísticos. Se espera que los participantes que integraron la exposición repetida a un mismo trauma y a claves contextuales que enmarcan la revelación desde un foco social con entrenamiento virtual en mindfulness mejoren los niveles de sintomatología de estrés postraumático y salud auto percibido.

## 1.4 DELIMITACIÓN

**1.4.1. Alcance.** Se define el alcance según los siguientes criterios:

#### 1.4.1.1. Espacio.

- El proyecto será elaborado en una residencia propia, y a la vez, se aprovechará el espacio universitario como apoyo para el proceso de desarrollo.

#### 1.4.1.2. Tiempo.

- El periodo de tiempo de implementación es menor a 1 año (6 meses aproximadamente)

#### 1.4.1.3. Contenido.

- La elaboración de esta herramienta permitirá una extracción limpia de los datos y metadatos de las diferentes cuentas de la red social twitter, gracias a la aplicación de diferentes métodos de curación de datos aportados por la librería de extracción utilizada (twitter4j), aprovechando la accesibilidad y obtención por medio del API proveído por la comunidad de desarrolladores. Esto permitirá en un futuro establecer un análisis de forma sencilla debido a la calidad de estos datos.

- La herramienta permitirá un proceso de extracción además de un almacenamiento ágil gracias al estudio y aplicación de conceptos de bases de datos, tanto relacionales como no relacionades enfocados en la optimización y características propias de los tipos de datos recolectados, adicionalmente el almacenamiento se realizará de una forma parametrizable.

- La extracción de los datos, conllevará a un posterior almacenamiento de estos en un sistema de base de datos especializado (NoSQL).

**1.4.2. Limitaciones.** Se definen las limitaciones según los siguientes criterios:

#### 1.4.2.1. Espacio.

- El desarrollo de la herramienta se desarrolla en su mayoría en una residencia la cual posee como servicio una conexión a internet con velocidad moderada, la otra parte de la herramienta se desarrolla en las instalaciones de la universidad católica de Colombia.

#### 1.4.2.2. Conceptual (contenido).

- La visión planteada por el proyecto padre implica que este se desarrolle por fases, en las cuales el proyecto propuesto tomaría el papel de la primer fase, la cual consiste en la extracción y almacenamiento de datos en la red social seleccionada.

- Herramientas utilizadas como el API de twitter poseen limitaciones en cuanto al volumen de información extraído a través de las credenciales creadas.

- La infraestructura o vista de despliegue comprometida en la implementación es básica.

- El desarrollo y la documentación de esta herramienta será ejecutado por una sola persona.

#### 1.4.2.3. Tiempo.

- La implementación del prototipo se llevará a cabo en el primer semestre del año 2017.

- El tiempo de desarrollo y documentación del proyecto está limitado por el tiempo definido del periodo académico.

- La actualización o retroalimentación de la herramienta se ve comprometida por tiempo definido.

# 2. OBJETIVOS DEL PROYECTO

## 2.1 OBJETIVO GENERAL

Desarrollar una herramienta web para la extracción y almacenamiento de datos de la red social Twitter que permita en un futuro con el apoyo de herramientas externas, establecer un análisis de estos datos, enfocado a una necesidad en un contexto específico.

## 2.2 OBJETIVOS ESPECÍFICOS

- Analizar diferentes métodos, para llevar a cabo la extracción y almacenamiento de datos de redes sociales.
- Identificar los requerimientos para la elaboración de la herramienta de extracción y almacenamiento además de definir el diseño de su arquitectura.
- Implementar la herramienta de software acoplada a la arquitectura definida, que permita la extracción y el almacenamiento de los datos recolectados de la red social twitter.
- Realizar una fase de pruebas unitarias que permita garantizar un buen desarrollo de la herramienta y así proveer un conjunto de datos aplicables a un posterior análisis.

# 3 MARCO DE REFERENCIA

## 3.1 ESTADO DEL ARTE

Actualmente, el valor de la información para las organizaciones se ha incrementado en cifras significativas. Claramente la información es y ha sido una muy importante herramienta de construcción global y toma de decisiones, pero con el paso del tiempo y el incremento de avances tecnológicos, esta ha tomado un impulso que ha llevado a que se valorice aún más.

Todo se reduce en el desmesurado crecimiento de fuentes de información y la posibilidad de generar un análisis específico, que se adapte a los objetivos de cualquier tipo de organización que desee obtener un beneficio a través de los resultados de dicho análisis.

Diferentes ciencias y tecnologías se han enfocado en este tema, generando una variedad de técnicas para extraer, almacenar y manipular diferentes tipos de datos provenientes de cualquier fuente. Cada una tiene sus métodos y algunas se adaptan a diferentes tipos de datos. Es allí donde nos encontramos con diferentes conceptos utilizados comúnmente al hablar de datos y su manejo, como lo son la minería de datos, inteligencia de negocios (BI) y datos masivos (BigData) donde se busca obtener ese beneficio que la información brinda y que realmente para muchas organizaciones no es visible hasta que se puede consolidar utilizando los conocimientos adecuados.

Aclarando un poco estos conceptos, podemos empezar basándonos en aquel que está definido bajo el nombre de minería de datos (García García & Rodríguez, 2017) Lo describen como aquel donde se engloba todo un conjunto de técnicas encaminadas a la extracción de conocimiento procesable, implícito en las bases de datos. En otras palabras, nos indican que entre sus funciones esta la preparación, el sondeo y la exploración de los datos para obtener información oculta en ellos, pero en realidad es ¿información oculta?, muchos autores definen que más que información oculta, todo se centra en encontrar las relaciones entre estos datos, es decir, armar un rompecabezas con ellos o seguir un camino que permita la construcción de esta información.

Es allí donde debemos aclarar que tanto el concepto de BI y el de BigData no se refieren a un estilo de minería de datos, más allá de esto podemos definirlos como un conjunto de procesos, métodos y tecnologías para permitir la captura, almacenamiento, distribución, administración y análisis de la información, los

cuales buscan alcanzar una meta especifica cómo lo es comúnmente en las organizaciones el mejorar la toma de decisiones(SAIMA Solutions, 2013).

Esta definición se construye tomando en cuenta los diferentes aportes de algunos autores, donde por ejemplo en (Gandomi & Haider, 2015) se refieren a BigData como alto volumen, alta velocidad y / o activos de información de alta variedad, que exigen formas innovadoras y rentables de procesamiento de la información que permiten una visión mejorada, toma de decisiones, y la automatización de procesos. Mientras que en el mismo glosario de TI de Gartner (Gartner Inc., 2013) se refieren a Business intelligence (BI) como un término general que incluye las aplicaciones, la infraestructura y las herramientas y mejores prácticas que permiten el acceso y análisis de la información para mejorar y optimizar las decisiones y el rendimiento.

Aunque los conceptos dan a entender que tienen un objetivo y un procedimiento similar, existen muchas diferencias entre ambos, por ejemplo, difieren tanto en el modo en el que se realizan sus procesos, como en el tipo de datos que analizan.

Mientras que los datos que utiliza la inteligencia de negocios proceden de bases datos relacionales, es decir, son datos estructurados los cuales pueden ser fácilmente ordenados y procesados por las herramientas de minería de datos y estos generalmente son almacenados localmente por una organización para su posterior análisis, los datos que utiliza BigData son en su mayoría datos no estructurados(Schroeck, Michael; Shockley, Rebecca; Smart, 2012). Seth Grimes, un analista líder en la industria en la confluencia de las fuentes de datos estructurados y no estructurados, publicó un artículo que decía, “el 80% de la información relevante para el negocio se origina en forma no estructurada, principalmente texto.”. La información que esconden los datos ya no requiere un formato específico, pueden ser fotos, videos, audios o incluso mensajes cortos de texto.

El autor y periodista Kenneth Cukier reconocido por su obra (2013) .Big Data: A Revolution That Will Transform How We Live, Work, and Think, en una conferencia dada en Berlín en el año 2014 (TEDSalon Berlín 2014) la cual se tituló: Big Data is better data, expuso el verdadero valor de este concepto y su influencia en el tiempo actual y el desarrollo global. Entre las ventajas de utilizar BigData se enfocó en dejar claro que “más datos no solo nos permiten ver más de lo que ya sabemos, nos permiten ver cosas nuevas, diferentes y con mejor perspectiva”.

Según el reconocido blog de desarrollo y tecnología Plazi, en (Rojas & Platzi, 2016) se expone que precisamente de eso se trata el Big Data, la facilidad de acceder a información en tiempo real sobre lo que está pasando en las redes sociales, es una cuarta parte de los datos que la Inteligencia de Negocios (Business intelligence) convencional está ignorando, y el mercado lo está

evidenciando. En pleno siglo XXI donde cualquier cambio social, económico o político nos permite tomar decisiones instantáneamente para analizar, y poder reaccionar a tiempo. Está, es una de las múltiples ventajas del Big Data. ¿Qué utilidad podemos dar a todos estos datos si no entendemos que nos están diciendo? Más allá de si existe una diferencia entre los términos, el Big Data hoy se está utilizando para hacer análisis de información de forma más desarrollada, y por eso está llevando al Business Intelligence a otro nivel. Las técnicas de Big Data son una evolución del Business Intelligence, pues permiten transformar una gran cantidad de datos en información significativa para la toma de decisiones de negocios.

Sin duda alguna BigData es un concepto que se impone y se convierte en una tendencia a nivel de manejo de información global “Datos de todo el mundo, de todas partes, de todas formas”, por lo tanto conocer que atributos lo conforman es necesario para el desarrollo del proyecto expuesto en el presente documento. BigData se define y distingue actualmente por medio de diferentes dimensiones que en un principio fueron tres pero que varias organizaciones y autores han planteado como cuatro y hasta cinco, estas son las V del BigData.

Abarcando el concepto de BigData propuesto por IBM en (Schroeck, Michael; Shockley, Rebecca; Smart, 2012), se puede apreciar cómo se plantean diferentes dimensiones, en este caso cuatro y las características de cada una. Volumen, Variedad, Velocidad y Veracidad son las dimensiones que proponen en esta decisión, donde adicional a las tres V propuestas previamente por diferentes autores, se define la veracidad como cuarta V, la incertidumbre de los datos, donde de una forma concreta que algunos datos son intrínsecamente inciertos, por ejemplo, los sentimientos y la sinceridad de los seres humanos; los sensores GPS; las condiciones climáticas; los factores económicos; y el futuro.

A la hora de tratar con estos tipos de datos, ninguna limpieza de datos puede corregirlos. Aun así, y a pesar de la incertidumbre, los datos siguen conteniendo información valiosa. La necesidad de reconocer y abordar esta incertidumbre es una de las características distintivas de big data. La incertidumbre se manifiesta en big data de muchas formas. Se encuentra en el escepticismo que rodea a los datos creados en entornos humanos como las redes sociales; en el desconocimiento de cómo se desarrollará el futuro y cómo las personas, la naturaleza o las fuerzas ocultas del mercado reaccionarán a la variabilidad del mundo que les rodea.(Schroeck, Michael; Shockley, Rebecca; Smart, 2012)

Algo similar en estos conceptos es que incluyen en la operación grandes cantidades de datos y la relación que tienen con el concepto de minería de datos, esta relación se traduce en que uno de los métodos utilizados en la inteligencia de negocios y el BigData como ciencia para extraer y analizar datos es el Datamining,

cabe aclarar que es una opción entre varias y sus técnicas son las más utilizadas comúnmente en este tipo de procesos.

Esto nos indica que ciencias y tecnologías como la inteligencia de negocios y el BigData generalmente implementan técnicas de minería de datos, la cual es un campo de la estadística con relación a las ciencias de la computación para la extracción y análisis de grandes volúmenes de datos. De esta manera se extrae información de un conjunto de datos y a través de diferentes técnicas se altera haciéndola comprensible a un contexto definido.

Una de las áreas aplicables más importantes en la actualidad de la minería de datos consiste en la minería web o webmining. Claramente la web es interpretada como una fuente inagotable de datos, una fuente superior a cualquier otra y que cada día crece a un mayor nivel. Sin dudarlo, la minería de datos es un concepto tan robusto debido a sus fases de análisis, filtrado de datos, etc.

El web mining como lo expone (García García & Rodríguez, 2017) Usa herramientas de la minería de datos para extraer información tanto del contenido de las páginas, de su estructura de relaciones (enlaces) y de los registro de navegación de los usuarios.

Posterior mente plantean una definición del Web mining a través de tres diferentes clasificaciones:

- Minería del contenido de la Web, o Web Content Mining;
- Minería de la estructura de la Web, o Web Structure Mining;
- Minería de los registro de navegación en la Web o Web Usage Mining.

En el desarrollo de este proyecto y la herramienta propuesta se adopta el concepto de Web Content Mining, esto debido a que simplemente se extraerá información de las publicaciones de la cuenta deseada, la estructura no implica un factor determinístico en este fin.

Precisamente la posibilidad de acceso a esta gran cantidad de datos en este entorno permite que diferentes proyectos se pongan en marcha para dar una solución de objetivos específicos, aplicada a algún área del conocimiento y con el fin de aportar al descubrimiento global.

En este caso conceptos como BigData y minería web son aplicados al desarrollo de una herramienta web la cual se enfocará en la extracción y almacenamiento de los datos provenientes de una de las mayores fuentes de datos en la web, las redes sociales, especialmente enfocada en la red social twitter. Ya que la mayoría de datos que recibiremos son datos semi estructurados y la cantidad de datos a

extraer y almacenar son de gran magnitud, el concepto de BigData será nuestra base aprovechando que podemos aplicarlo en el desarrollo de la herramienta y las diferentes fases del proceso las cuales se enfocan en la data y se han definido como (Extracción, Curación y Almacenamiento).

En cuanto a la red social seleccionada, Twitter está catalogada como una de las más importantes actualmente, con aproximadamente más de 500 millones de usuarios, la cual ofrece un sencillo servicio que puede ayudar a un negocio a estar en contacto con seguidores y clientes a través de su servicio de mensajería con un máximo de 140 caracteres por mensaje.(Facchín, 2016)

En cualquier caso Twitter ofrece mucho más que la posibilidad de envío de mensajes y es la red con mayor número de aplicaciones. Para las empresa quizá sea la más desaprovechada de las “grades redes populares”. Adicionalmente, podemos identificar esta red social como una de las más utilizadas por entidades y personalidades reconocidas, adicionalmente los perfiles son verificados y confiables para la consulta de información de cada cuenta.

Las redes sociales como servicio se imponen como un medio de comunicación de nivel global, el cual permite el contacto entre personas a través de una red, generalmente internet. Esta red puede definirse según ((INTECO), 2009) como los servicios prestados a través de Internet que permiten a los usuarios generar un perfil público, en el que plasmar datos personales e información de uno mismo, disponiendo de herramientas que permiten interactuar con el resto de usuarios afines o no al perfil publicado.

Esto implica en las redes sociales habituales, que un alto porcentaje de la información de las cuentas públicas sea expuesta como un conjunto de datos abiertos, los cuales son datos con libertad de acceso, que pueden ser captados y distribuidos ateniéndose a las diferentes normativas de tratamiento de datos personales, tema un poco más permisivo respecto a cuentas en redes sociales de entidades públicas, publicitarias, personalidades etc. que representan un contexto más amplio de datos compartidos.

En cuanto a la privacidad de estos datos (Vásquez Vélez, 2012) aclara que para acceder a una red social esta pide una serie de datos que en principio son públicos, estos sirven para empezar a interactuar con la red, pero algunos de estos datos podrían estar catalogados como datos privados en las diferentes leyes de los países, son los mismos usuarios los que sin darse cuenta exponen su vida privada a toda la comunidad, hacen esto con el fin de tener alguna afinidad y pertenecer a la red social.

La importancia de la comunicación y la facilidad de obtenerla por medio de una red de este tamaño, genera que su utilidad sea mayor a todo pronóstico, por lo tanto el

flujo de información dentro de estas redes es casi incalculable. Si se deseara captar una pequeña parte de este flujo de información de forma rápida, sea referente a un sector de industria en específico, publicaciones de un propietario de cuenta, factores religiosos o políticos, con el objetivo de realizar un análisis o archivo de datos, es realmente necesaria una herramienta que permita llevar a cabo el proceso de extracción y almacenamiento de estos datos por lo tanto, la propuesta planteada en este documento es la elaboración de una herramienta que cumpla con este funcionamiento y que se adapte a las fases del desarrollo de software según el ciclo de vida estándar.

El ciclo de vida del desarrollo de software (SDLC), es una base estable, que toma papel como marco de referencia al definir un conjunto de diferentes fases, que permiten que el desarrollo de software garantice el cubrimiento total del sistema cumpliendo con los objetivos propuestos.

Varias metodologías y métricas definidas para el desarrollo de estas fases, a lo largo del tiempo han declarado como factor elemental, el seguimiento de una secuencia estructurada y bien definida del SDLC, establecido formalmente en (NTP-ISO/IEC 12207, 2006) como un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

El método para llevar a cabo el SDLC en forma general, consta de 5 procesos principales en los cuales la vida del sistema se describe y posteriormente define el curso de su implementación. Estos procesos se definen de forma general según (NTP-ISO/IEC 12207, 2006) como:

**1) Proceso de adquisición**: Es lo más similar a la ingeniería de requerimientos en el proceso de desarrollo de software, este proceso se concentra en encontrar las necesidades del adquiriente enfocadas a un producto que le permita satisfacer dichas necesidades, además de esto en este proceso se plantean las diferentes propuestas y las diferentes actividades del adquiriente.

**2) Proceso de suministro**: Es el proceso que comienza luego de recibir las propuestas del adquiriente y consiste en establecer los pasos a seguir y los diferentes bienes o recursos que estarán implicados en la realización del proyecto con el fin de asegurar su éxito y completitud todo esto comprende todas las actividades del proveedor hasta la entrega del sistema, producto o servicio de software.

**3) Proceso de desarrollo**: Se plantea como el proceso en el cual el desarrollador se encarga de realizar las correspondientes actividades del análisis de los requerimientos, el diseño, la codificación, la integración, las pruebas, la instalación y la aceptación del sistema, producto, o servicio de software solicitado. Se especifica claramente que las actividades del desarrollador llegaran hasta donde esté estipulado en el contrato.

**4) Proceso de operación**: Es un proceso enfocado al funcionamiento operacional del sistema por lo tanto cubre todos los procesos del producto y el apoyo a la operación del usuario.

**5) Proceso de mantenimiento**: Proceso que involucra el correcto funcionamiento del sistema luego de realizar un cambio debido a un nuevo requerimiento o alguna adaptación necesaria, apoyando su estabilidad operacional y garantizando el rendimiento en cada una de sus funciones. El proceso incluye la migración y retirada del producto de software.

La extracción de datos en la web se realiza con el fin de obtener de estos, alguna clase de información favorable a un objetivo fijo, también para almacenarlos en caso de un repositorio remoto y dinámico. Entre los conceptos de extracción de datos en la web se encuentra el Web Scraping el cual (Glez-Peña, Lourenzo, López-Fernández, Reboiro-Jato, & Fdez-Riverola, 2013) define como el proceso de extracción y la combinación de contenidos de interés de la Web de una manera sistemática. En tal proceso, un agente de software, también conocido como robot Web, imita la interacción de navegación entre los servidores web y el humano en un recorrido Web convencional. Paso a paso, el robot tiene acceso a la mayor cantidad de sitios web según sea necesario, analiza su contenido para encontrar y extraer datos de interés y las estructuras de los contenidos si lo deseas. Las tareas de scraping de datos más comunes en la web, involucran en lograr los objetivos de recuperación.

En cuanto al acceso al sito, el scraping de datos de Web establece la comunicación con el sitio web de destino a través del protocolo HTTP, de acuerdo a (IETF, 1999) Un protocolo de Internet basado en texto sin estado que coordina las operaciones de petición-respuesta entre un cliente, normalmente un navegador Web y un servidor Web.

El análisis de HTML y contenidos de extracción: Una vez que se recupera el documento HTML, el extractor de datos Web puede extraer el contenido de

interés. Para este propósito, expresiones regulares, solo o en combinación con lógica adicional, es ampliamente adoptado.

El scraping es más robusto cuando el sitio implementa micro formatos o micro datos de salida: El objetivo principal es transformar el contenido extraído en una representación estructurada que es adecuada para su posterior análisis y almacenamiento. Algunas herramientas son conscientes del resultado post-procesamiento, proporcionando estructuras de datos en memoria y soluciones basadas en texto, tales como cadenas o archivos (normalmente archivos XML o CSV).

Adaptando este ciclo y con el fin de concretar la funcionalidad de la herramienta, en cuanto al trabajo de extracción de los datos, se adoptan de manera investigativa, diferentes conceptos relacionados con la identificación y recuperación de datos en la red. Entre las diferentes fases del proceso general, existe una fase que permite que los datos recolectados tengan la capacidad de ser reconocidos y adaptados a un formato para su posterior almacenamiento y la correspondiente utilidad que brindan tanto para su consulta como para los diferentes análisis y soluciones, este proceso se ha definido como fase de curación de datos, un concepto que permite la calidad de la información resultante y analizada. Gartner estima que más del 25% de los datos críticos en las principales empresas del mundo es defectuoso (Gartner Inc., 2007). Los problemas de calidad de los datos pueden tener un impacto significativo en las operaciones comerciales, especialmente cuando se trata de los procesos de toma de decisiones dentro de las organizaciones (D. Wood, 2010). La curación de datos proporciona el soporte de gestión de datos metodológicos y tecnológicos para abordar problemas de calidad de datos que maximizan la usabilidad de los datos.

Volviendo al proceso general del proyecto y sus diferentes fases, se comprende que la recuperación de estos datos requiere normalmente escribir y ejecutar comandos de recuperación o extracción de datos o consultas en una base de datos. Sobre la base de la consulta prevista, la base de datos busca y recupera los datos solicitados. Aplicaciones y software en general utilizan diversas consultas para recuperar datos en diferentes formatos. Además de los datos simples o más pequeños, la recuperación de datos también puede incluir la recuperación de grandes cantidades de datos, por lo general en forma de informes.

Las bases de datos implican una importancia de alto grado en el funcionamiento de la herramienta, sin una base de datos, la extracción de estos por parte del software no tendría un sentido ya que no existiría un proceso de almacenamiento ordenado que permita posterior mente la utilización de los datos recolectados. Pero, ¿Cuál es el tipo de base de datos necesario para una herramienta de este

tipo? ¿Cómo dependiendo de esta declaración, se verá afectado el proyecto teniendo en cuenta los atributos de calidad?

En cuanto a las bases de datos relacionales, hay que decir que hasta el momento son a las que se les da un mayor uso, llevan un buen tiempo en el mercado, esto implica que tengan un mayor nivel de soporte y mejoras. A través de claves primarias asignadas a los atributos más relevantes, las diferentes tablas en estas bases se relacionan y permiten mantener un nivel de integridad de la información, además del contexto general, esto genera una dependencia a los tipos de datos de las diferentes tablas al momento de relacionarse.

Sin embargo, tomando como base la mayoría de desarrollos a través del tiempo, los cuales utilizan bases de datos, se ha probado que en muchos entornos, las bases de datos relaciónales no resultan ser tan productivas como parecen, esto debido a que un gran porcentaje de aplicaciones, actualmente tienen la necesidad de atender millones de clientes y al mismo tiempo brindar calidad en aspectos como disponibilidad, fiabilidad y consistencia.

Allí es donde las relaciones establecidas en las diferentes tablas generan un conflicto, ya que si aumentan los usuarios, entonces los datos aumentan proporcionalmente, estos datos tienen que ser almacenados en muchos servidores debido a su volumen, además deben estar distribuidos en diferentes ubicaciones. Esta distribución hace que sea difícil mantener las relaciones entre los datos, además las aplicaciones web requieren la capacidad de escalar en muchos servidores. En este punto es donde se presentan las bases de datos no relacionales (NoSQL) como una solución, ya que al no establecer relaciones el almacenamiento se convierte en una estructura escalable la cual puede manejar volúmenes enormes de datos.(Lotfy, Saleh, El-Ghareeb, & Ali, 2016)

Hay que resaltar que NoSQL no prohíbe estrictamente el uso de SQL, sin embargo lo usual es que la mayoría de las bases de datos NoSQL evitan utilizar este tipo de lenguaje o lo utilizan como un lenguaje de apoyo. Casos de base de datos NoSQL como MongoDB la cual es la líder en su categoría utiliza JSON como lenguaje de consultas que permite que no se siga estrictamente un esquema estático y adicionalmente trata los archivos de datos como si estuvieran en memoria.

Este tipo de base de datos son utilizadas frecuentemente en la práctica de BigData, donde debido a la excesiva cantidad de datos extraídos, debe existir una infraestructura de almacenamiento adecuada e independiente de los datos recolectados.

Un concepto a tener en cuenta es el de DataScience el cual definen en (Weigend, 2014) como la ciencia que estudia la extracción de conocimiento a partir de los

datos y como generar esa información a partir de diferentes prácticas. Adoptando estos conceptos, la aplicación de técnicas específicas de minería de datos, enfocadas en minería web en redes sociales, además de un almacenamiento adecuado utilizando bases de datos no relacionales (esto debido a las condiciones de volumen de datos y requerimientos del sistema), todos integrados en un proceso de desarrollo de software definido por una metodología ágil, generarán la construcción de una herramienta eficaz para la extracción y almacenamiento de datos de redes sociales, enfocada en brindar los elementos necesarios para realizar un posterior análisis con el fin de alcanzar un objetivo o beneficio.

En cuanto al proceso de extracción de datos, además de tener conocimiento de que debe existir una fase de desarrollo basado en algún estándar, marco de trabajo o metodología, no está demás tener en cuenta que dependiendo del tipo de extracción que se necesite realizar, y el entorno de la información implicada, existen algunas herramientas que pueden funcionar como una llave de acceso o acercamiento a estos datos de una forma más sencilla y dinámica.

Las APIs o interfaces de programación de aplicaciones, son una herramienta que nos permite crear este vínculo con diferentes funciones que pertenecen a un servicio web de manera segura y confiable, este vínculo se puede establecer desde nuestra propia aplicación, lo que permite que un proceso de extracción de datos en la mayoría de casos tenga un alto grado de eficiencia y una estructura definida, que en ciertos casos ayuda a la curación de estos datos generando un ambiente prospero para los diferentes procesamientos que puedan aplicarse.

En este caso el API de Twitter para desarrolladores se convierte en un servicio que se integra al proyecto. Este nos permite establecer una conexión con la red social para posteriormente extraer los datos públicos necesarios y almacenarlos en una base de datos. Claramente esta aplicación implica acciones específicas y concretas que se encontrarán una mejor definidas en la sección de desarrollo del proyecto.

Por otra parte es importante resaltar la participación de Liferay como portal gestor de contenidos en cuanto al entorno web donde estará posicionada la estructura funcional de la herramienta. Liferay (Liferay, 2017) es un portal que nos permite facilitar el diseño de interfaces de usuario ya que nos provee de un conjunto de herramientas sencillas de utilizar y con una gran variedad de funcionalidades, también nos permite integrar nuestra aplicación con otras por medio de métodos como SOAP, REST, RSS y APIs propietarias. Su uso disminuye la complejidad del desarrollo de la herramienta y nos permite tener una plataforma fácil de utilizar y modificar ya que permite la personalización por parte de los usuarios de cada página y también se adapta a los diferentes roles del sistema. La implementación

de Liferay se acomoda a las necesidades del proyecto y a la metodología de desarrollo ágil planteada.

# 4 MARCO CONCEPTUAL

## 4.1 RECUPERACIÓN DE DATOS (DATA RETRIVAL)

Basados en el concepto expuesto por (Techopedia, 2016) cuando nos referimos en bases de datos a la recuperación de datos o data retrieval, estamos hablando del proceso de identificar y extraer datos de una base de datos, con base a una consulta proporcionada por el usuario o la aplicación. Este proceso permite la búsqueda de datos desde una base de datos con el fin de visualizar en un monitor y / o utilización en una aplicación.

Para poder lograr una recuperación de datos de una base de datos principalmente se opta por realizarlo a través de sentencias de lenguaje específico de recuperación o extracción de datos o consultas en una base de datos. Generalmente la mayoría de aplicaciones o sistemas que utilizan bases de datos están ligadas fuertemente con el data retrieval ya que en el funcionamiento general del sistema es necesaria la comunicación de los datos locales y las diferentes operaciones implicadas.

## 4.2 DATOS ABIERTOS (OPEN DATA)

Datos abiertos es un concepto que ha tomado fuerza con el avance de la tecnología y el crecimiento masivo de la información. Este concepto se aplica a un entorno el cual proporciona métodos mecánicos para publicar y obtener datos, cuando nos referimos a los datos como abiertos, hacemos énfasis en su libertad de acceso, de allí se pronuncian como un conjunto de datos disponible para cualquier persona que pertenezca o consuma servicios de este entorno, permitiendo operarlos y distribuirlos sin restricción alguna.

El acceso abierto permite una difusión de conocimiento a nivel global desde que no existan clausulas legales de esta práctica en el entorno o que no permitan que sea un acceso abierto de gran magnitud.

## 4.3 RASTREADORES WEB (WEB CRAWLES)

Los rastreadores web o Web Crawlers son recopiladores de información que recorren la web con alguna trayectoria predeterminada según (Olston & Najork, 2010), Un rastreador web (también conocido como un robot o una araña) es un

sistema para la descarga masiva de páginas web. Estos rastreadores Web se utilizan para una variedad de propósitos. De manera prominente, son uno de los principales componentes de los motores de búsqueda web, sistemas que reúnen un cuerpo de las páginas web, el índice de ellas, y permiten a los usuarios emitir consultas en el índice y encontrar las páginas web que coincidan con las consultas. Un uso relacionado es el archivo web, donde grandes grupos de páginas web se recogen y archivan para la posteridad periódicamente. Un tercer uso es la minería de datos de la web, donde se analizan las páginas web de las propiedades estadísticas, o cuando el análisis de datos se lleva a cabo en ellos.

## 4.4 SERVICIOS WEB (WEB SERVICES)

El consorcio W3C (Web, 2011) define los Servicios Web como sistemas software diseñados para soportar una interacción interoperable maquina a máquina sobre una red. En esta definición se expone que en situaciones generales los servicios web principalmente suelen ser APIs Web las cuales son interfaces que nos permiten consumir funciones de un servicio específico a través de nuestra propia aplicación o producto de software.

En los últimos años se ha popularizado un estilo de arquitectura Software conocido como REST (Representational State Transfer). Este nuevo estilo ha supuesto una nueva opción de estilo de uso de los Servicios Web (Marset, 2007).

## 4.5 DATOS MASIVOS (BIG DATA)

Big Data es un concepto el cual se refiere a la acumulación masiva de datos y los procedimientos utilizados para identificar patrones recurrentes dentro de los datos. Para analizar esta gran cantidad de datos, Big Data se apoya de diferentes técnicas tales como la asociación, minería de datos, agrupación (clustering) y predicción.

Adicionalmente, existen muchos tipos de datos y cada uno tiene un proceso de análisis diferente. Los datos pueden ser texto, audio, video o diferentes medios de comunicación social, en el caso de los datos de texto, estos pueden analizados a través de la extracción de la información, aplicando técnicas de integración de texto, búsqueda de respuestas y análisis de sentimientos. Estos datos pueden ser sometidos a un análisis predictivo que utiliza métodos estadísticos para llevar a cabo una predicción descriptiva y de decisión, lo que genera un beneficio a futuro en el contexto y entorno en los cuales se utilice esta tecnología.

## 4.6 MINERÍA DE CONTENIDO WEB (WEB CONTENT MINING)

La minería de contenido web es el proceso de extraer información útil del contenido de los documentos web. Puede consistir en texto, imágenes, audio, vídeo, o registros estructurados, tales como listas y tablas.

La aplicación de la minería de texto de contenido web ha sido el más ampliamente investigado. Los temas abordados en la minería de texto incluyen descubrimiento y seguimiento de temas, la extracción de patrones de asociación, agrupación de documentos web y clasificación de páginas web.

Investigaciones sobre este tema se han basado en gran medida en las técnicas desarrolladas en otras disciplinas tales como la Recuperación de Información (RI) y Procesamiento del Lenguaje Natural (PLN).(Srivastava, Desikan, & Kumar, 2006)

# 5 METODOLOGÍA

La metodología utilizada en el desarrollo de la herramienta de software es un concepto actualmente desarrollado que toma como base la metodología de desarrollo ágil XP o programación extrema y a través de un conjunto de ajustes, da una aplicación enfocada al desarrollo de manera singular o exclusiva para un solo programador la cual se expone en la sección de implementación.

Aprovechando sus características de implementación que la clasifican dentro del grupo de metodologías de desarrollo ágiles, las cuales implican métodos de ingeniería del software basados en el desarrollo iterativo e incremental, estas metodologías son necesarias en un mundo dinámico e impredecible como el nuestro, además de plantear un manifiesto ágil el cual según (Issi, 2003) valora o da prioridad al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas (Es más importante construir un buen equipo que construir el entorno). Además plantea la importancia de desarrollar software que funcione más que conseguir una buena documentación, La colaboración con el cliente más que la negociación de un contrato y responder a los cambios más que seguir estrictamente un plan.

## 5.1 METODOLOGÍA XP

La programación extrema (XP) tomando como base los conceptos planteados en (Bustamante & Rodríguez, 2014) es una metodología de desarrollo de software planteada por Kent Beck en el año (1999). Este tipo de programación se diferencia de las metodologías comunes principalmente en que propone como factor primario la adaptabilidad antes que la previsibilidad. Posee un conjunto de valores que la identifican además de un conjunto de etapas definidas.

Valores como la simplicidad a la hora de agilizar el desarrollo, comunicación en cuanto a la forma de codificar, retroalimentación establecida por el cliente al dar su opinión de los avances del proyecto y coraje al diseñar y programar para hoy y no para mañana, hacen parte de esta metodología. Cada uno de estos valores se ve reflejado en un conjunto de fases propias de la metodología. Aunque el concepto expuesto en este caso para el desarrollo de la herramienta web es ligeramente diferente en algunas prácticas ya que la programación se realiza de manera singular, las fases de la metodología no cambian y por lo tanto se definen continuación tomando como base la definición expuesta por (Bustamante &

Rodríguez, 2014), La cual expone como fases de la metodología XP:

- **Planificación del proyecto**

  Esta etapa comprende la definición de historias de usuario con el cliente, las historias tienen una gran similitud a los casos de uso relacionándose con el levantamiento de requerimientos, pero con algunas diferencias, estas historias son líneas escritas por el cliente en lenguaje no técnico y son usadas para la estimación de tiempos, también se definen las iteraciones y tiempos en cuanto a la velocidad del proyecto.

- **Diseño**

  Consta del planteamiento de diseños simples y sencillos pero de calidad, también promueve la utilización de glosarios de términos para facilitar el entendimiento del diseño y abrir paso a la reutilización de código por ejemplo en posteriores actualizaciones, adicionalmente, plantea el tratamiento de riesgo para su reducción de forma rápida, el descarte de la funcionalidad extra y la refactorización de código sin alterar su funcionalidad con el fin de optimizar el funcionamiento.

- **Codificación**

  Esta fase comprende la programación o codificación de las historias o requerimientos, esta programación debe hacerse controlada por un estándar definido esto mantiene el código consistente y facilita su comprensión y escalabilidad.

- **Pruebas**
  Como fase final, implica el testeo del código para verificar su funcionalidad, sometiendo a estas pruebas distintas clases del sistema omitiendo los métodos más triviales, estos test no deben tener ninguna dependencia del código que en un futuro evaluará.

Algunas de las ventajas de esta metodología comprenden atributos que aportan al desarrollo eficiente de un producto de software. Esta metodología resalta una programación sumamente organizada, impulsa la comunicación cliente-desarrollador, permite ahorrar tiempo y dinero además de ser una metodología aplicable a cualquier lenguaje de programación y en la cual el nivel de errores es

bajo. Por otra parte algunos autores resaltan como desventaja su variedad de dificultad ya que no siempre es más fácil aplicarla que realizar un desarrollo más tradicional, adicionalmente se recomienda su uso en proyectos a corto plazo.

En este caso estas cualidades se ajustan perfectamente a las condiciones para el desarrollo de la herramienta web de extracción y almacenamiento ya que requiere un desarrollo ágil sin basta documentación y con las fases expuestas previamente.

## 5.2 IMPLEMENTACIÓN DE METODOLOGÍA (PXP)

La metodología XP fue definida con un grupo de cualidades y procesos, entre los cuales se planteó que el desarrollo pertinente seria llevado a cabo por parejas, dándole una optimización de tiempo a las etapas de la metodología, sin embargo se han desarrollado con el tiempo un conjunto de prácticas que definen un concepto de aplicación de esta metodología enfocándose en el desarrollo realizado por un solo programador. Tal como (Agarwal & Umphress, 2008) examinan y proponen el concepto de Personal Extreme Programming (PXP), el cual se basa en los valores de la metodología XP, la práctica de implementar este concepto, está impulsada por el hecho de poder implementar y recibir la mayoría de ventajas que proporciona la metodología XP propuestas anteriormente.

(Agarwal & Umphress, 2008) definen a través de una tabla de procesos, todas aquellas actividades que deben ser ejecutadas para poder adaptar la metodología XP a un desarrollo realizado por un solo programador. En esta tabla se definen tres tareas principales las cuales son la planeación, el desarrollo y el después de la muerte (post mortem), donde cada una de ellas contiene actividades específicas que permitirán al programador solitario aplicar el concepto de (PXP).

Entre las fases de la metodología, se comprende en la planeación el proceso de obtención y especificación de requerimientos, fue implementada para este proyecto de una forma en la cual se define la metáfora del sistema, se crean las historias de usuario y se plantean los requisitos funcionales y no funcionales de la herramienta, definiciones que se pueden encontrar en el documento de especificación de requerimientos de software (Véase el Anexo A ).En esta misma fase se establecen las tareas a realizar, tareas que fueron aplicadas al cronograma del proyecto planteado en la investigación de anteproyecto.

El control de versiones hace parte de esta metodología, existen diferentes herramientas que permiten llevar este control almacenando los diferentes archivos

o carpetas de forma histórica, de manera que cualquier persona involucrada en el desarrollo pueda recuperar versiones anteriores de estos documentos. Adicionalmente estas herramientas permiten llevar un control de los diferentes cambios que se realicen en cada sincronización a través de comentarios generados y la identificación de la persona que realizo el cambio.
En el proyecto actual se lleva un control de versiones a través de la plataforma GitHub, la cual es una plataforma de desarrollo colaborativo de software que nos permite subir nuestros proyectos o codificaciones utilizando el sistema de control de versiones Git este último creado por el ingeniero Linus Torvalds en el 2005.(git, 2015)

Como apoyo a la fase de desarrollo del proyecto se ha creado un repositorio en GitHub de forma privada, el cual fue proveído por el docente implicado como tutor en este proyecto. Repositorio en el cual se fueron realizando las diferentes actualizaciones según el avance semanal del desarrollo y se controlan las versiones con cada commit hecho al repositorio.

En la presentación del concepto de PXP dado por (Agarwal & Umphress, 2008) se puede identificar también un conjunto comparativo de doce prácticas propias de la metodología XP y como varían para la aplicación de PXP. Básicamente hacen referencia al juego de roles que debe realizar el desarrollador, al igual que el orden de las diferentes tareas diarias de desarrollo. Una de las ventajas frente a XP es la independencia a la hora de tomar decisiones estructurales en el desarrollo, la simpleza en el diseño incrementa sin afectar la funcionalidad del software y también permite una concreta planificación y control de tiempos durante el proceso general.

Por su funcionalidad, condiciones, ventajas y técnicas de implementación el concepto de PXP como metodología ágil fue implementado durante el desarrollo del proyecto y se promueve su aplicación en posteriores desarrollos con las características apropiadas para su aplicación.

# 6 DESARROLLO DEL PROYECTO

Se presentan en esta sección del documento los pasos, métodos, técnicas y herramientas utilizados en la elaboración del proyecto desde fase de investigación hasta la fase final.

## 6.1 FASE DE INVESTIGACIÓN

El desarrollo del proyecto inicia a través de una fase de investigación, en la cual el objetivo es indagar, comparar y aprender sobre diferentes conceptos que pudieran ser necesarios en la implementación de la herramienta web, conceptos referentes al tratamiento de datos de la web, herramientas, métodos, técnicas de extracción de datos y su almacenamiento se postularon como los temas principales de investigación.

La información es actualmente uno de los bienes más valiosos y preciados en todo el mundo, esta puede ser generada a través de la correcta interpretación de los datos obtenidos de diversas fuentes, en este caso el proyecto posee como fuente de datos la red social Twitter, donde la magnitud de estos aumenta constantemente.

La mayoría de organizaciones buscan como aprovechar la información que circula en internet, miles de páginas, blogs, journals, todas juntas tienen una cantidad de información incalculable a la cual con la aplicación de un análisis adecuado y enfocado en un tema específico puede sacársele un máximo provecho que casi siempre se traduce a la toma de decisiones fundamentadas por parte de la organización.

En este caso aprovechando la gigantesca cantidad y calidad de datos proveniente de las redes sociales y enfocadas en la descripción del problema a tratar, se realizó una investigación profunda sobre recuperación de datos de redes sociales, extracción, curación de datos, almacenamiento y métodos por los cuales se pudieran aplicar o enfocar estos conceptos durante el desarrollo de la herramienta.

La información consultada en esta fase es proveniente de bases de datos científicas, bibliotecas virtuales y diferentes recursos web, en los cuales se realizaron las diferentes búsquedas luego de identificar los conceptos importantes de la investigación y sus palabras clave, teniendo en cuenta las técnicas de investigación aprendidas y expuestas en diferentes seminarios universitarios.

Este proceso de investigación comenzó en el mes de julio del año 2016, donde basados en la información recolectada y estudiada desde esa fecha hasta el mes

de noviembre del mismo año se realizó la propuesta del presente proyecto, es decir, es un proyecto que ha tenido un proceso extenso de investigación contemplando fuentes de información certificadas y catalogadas con información verídica. La fase de investigación no es un proceso con cierre absoluto, ya que durante las otras fases, la investigación sigue siendo un factor importante y participativo.

## 6.2 PLANIFICACIÓN DEL PROYECTO

Posteriormente luego de la fase de investigación se da inicio a la fase de planificación, donde se especifican las bases del proyecto y se define el camino correcto a seguir para su implementación, en la planificación se tuvieron en cuenta todas las posibles actividades y todos los posibles procesos que se requieren para llevar a cabo el proyecto basados en la investigación realizada hasta el momento.

Una parte esencial de esta fase consistió en el levantamiento de requerimientos de la herramienta, ya que allí es donde la información del cómo desarrollarla está ligada totalmente con las necesidades del estudio que en este caso sería nuestro cliente. Por lo tanto antes de definir los requerimientos de la herramienta, era necesario definir la metodología de desarrollo a seguir, esto debido a la necesidad de establecer la mejor manera de realizar el proyecto desde la planificación a la fase de pruebas, como lo vimos en el marco teórico y en la definición metodológica.

Luego de seleccionada la metodología que en este caso fue el concepto de programación extrema llamado Personal Extreme Programming, se obtuvo la base o guía para poder llevar a cabo el levantamiento de requerimientos, el cual como lo dicta la metodología aplicada debe contener algunas historias de usuario en las cuales basarse y una especificación formal que se puede encontrar en el documento de especificación de requerimientos (Véase el Anexo A).

Conjunto a la definición de los requerimientos debemos definir las herramientas base a utilizar para la construcción de la herramienta, cuestiones como ¿Qué arquitectura de software es la ideal? ¿Qué lenguaje de programación utilizar en la codificación? ¿Qué tipo de bases de datos deberían implementarse? ¿Cómo estarán relacionados los datos dentro del sistema? Y varias más surgen en esta fase de planificación y se abordan en la siguiente la cual corresponde a la fase de diseño.

## 6.3 DISEÑO

Al diseñar nosotros podemos aclarar un poco más el concepto que teníamos de la herramienta al realizar el levantamiento de requerimientos, ya que se contempla la estructura que tendrá la herramienta desde diferentes aspectos. Al iniciar esta fase lo primero a definir fue la arquitectura del proyecto, la cual fue definida con el apoyo del lenguaje unificado de modelado UML, el cual permite modelar la arquitectura de un sistema por medio de diferentes diagramas enfocados específicamente en un aspecto funcional o estructural.

En esta fase, con vista a la construcción de la herramienta se han establecido los siguientes diagramas los cuales se describen teniendo como base el concepto definido en (Clases et al., 2001)

- **Diagrama de casos de uso:** el cual modela la funcionalidad del sistema (representa los requerimientos funcionales) y se concentra en plasmar gráficamente las interacciones con el desarrollo es decir lo que el sistema puede brindar al usuario.

- **Diagrama de distribución o despliegue**: Este diagrama permite modelar la arquitectura física de un sistema informático, sus diferentes equipos físicos, conexiones y también los componentes de software en un nivel muy básico.

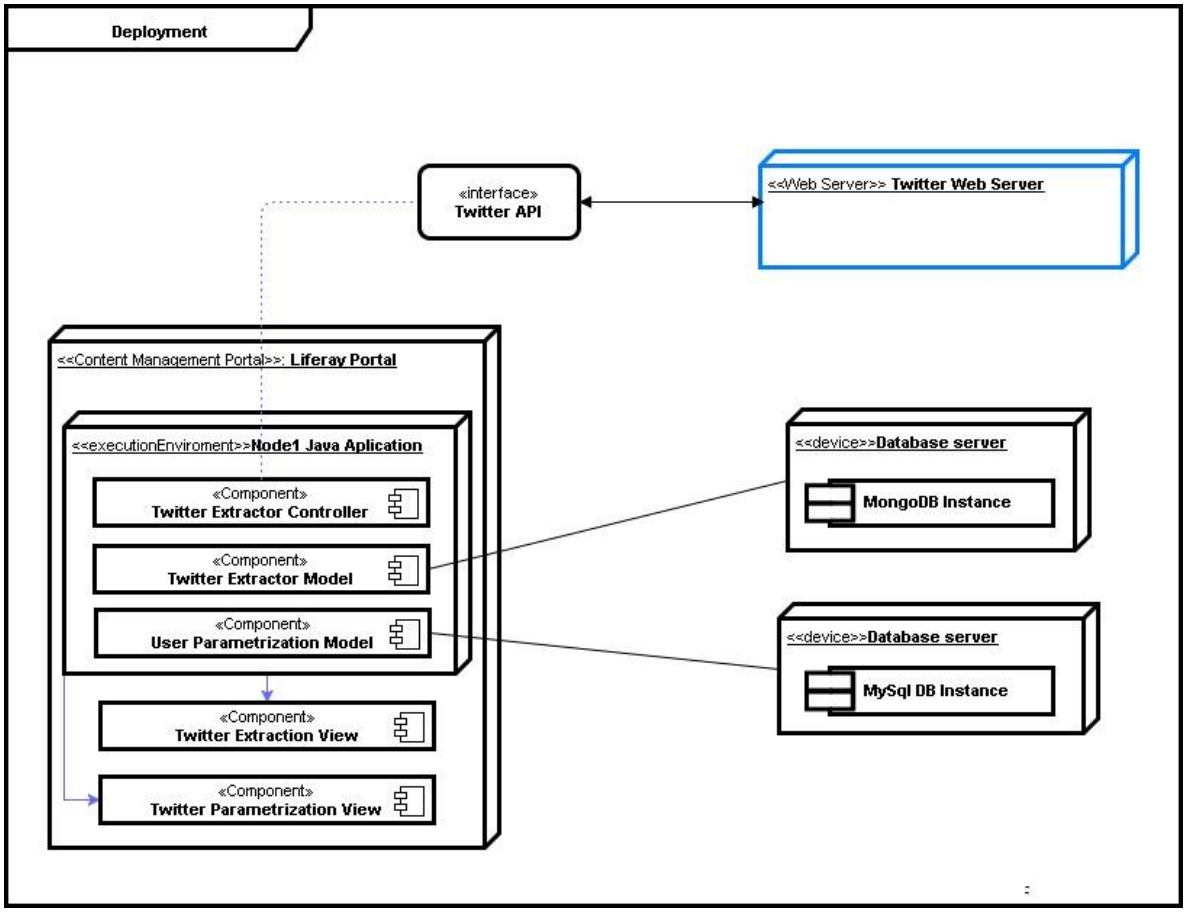
- **Diagrama de componentes:** El cual modela la organización de los diferentes componentes de software del sistema, sus diferentes dependencias y como se relacionan entre ellos.

Adicionalmente se ha generado un diagrama de procesos **BPMN** el cual se encuentra junto con los **diagramas de caso** de uso y el diagrama de **entidad relación** en el documento de especificación de requerimientos de software (Véase el Anexo A).

A continuación se ilustran y describen los diagramas de distribución y despliegue de la herramienta de software:

### 6.3.1 Diagrama de Despliegue.

Ilustración 1 - Diagrama de despliegue.




En el diagrama de despliegue podemos apreciar los diferentes componentes físicos del sistema y que permiten el funcionamiento de la herramienta. Todos juntos representan la arquitectura física de la herramienta.

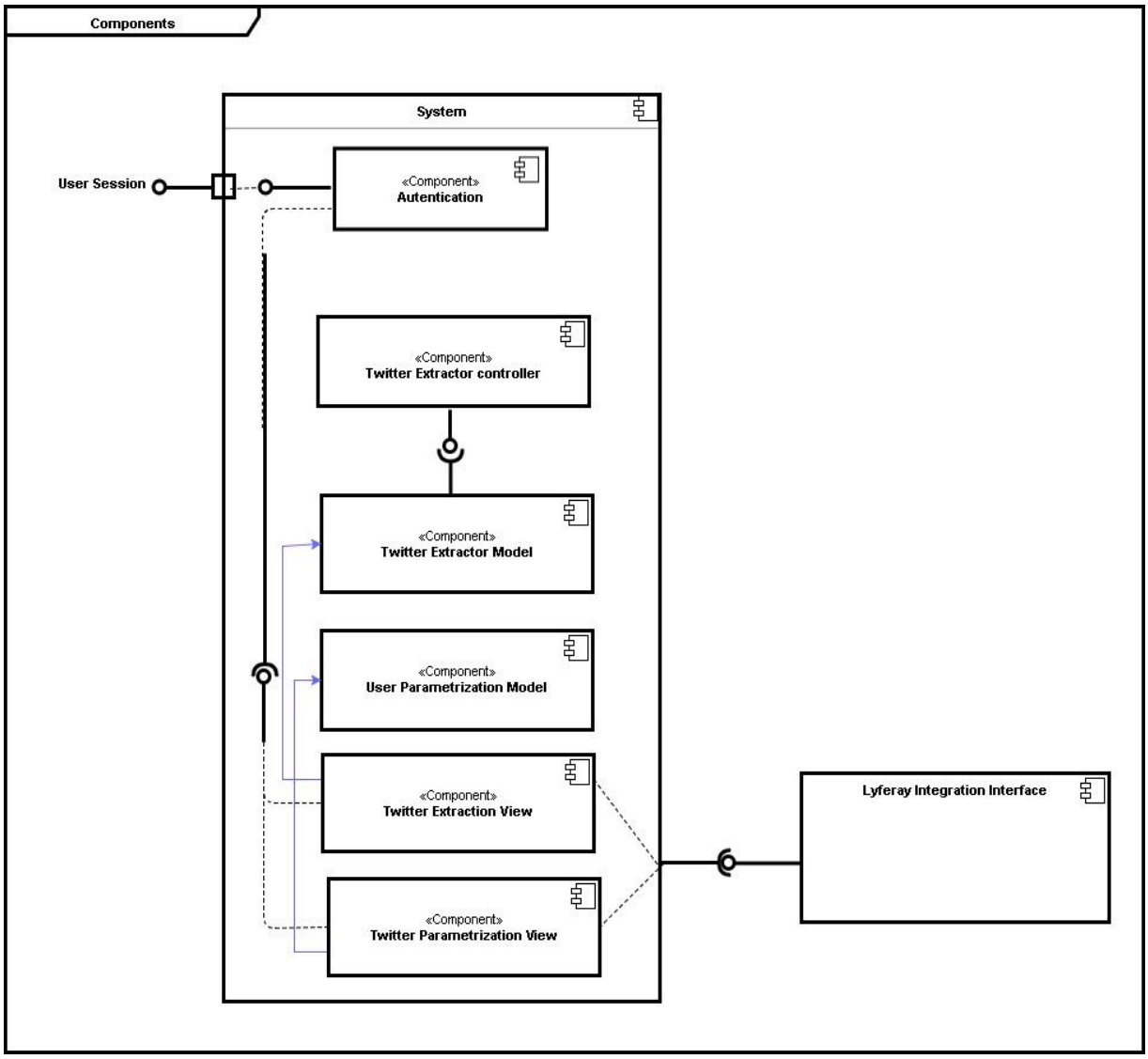
Se representa a través de un nodo contenedor el portal gestor de contenidos Liferay el cual es la plataforma donde la aplicación está alojada, dentro de este nodo se encuentra otro nodo que representa el ambiente de ejecución o la aplicación en un contexto de integración. Allí es donde nuestros componentes de software se encuentran (relevantes), tanto el componente de extracción como los componentes de los modelos tanto el relacional como el no relacional. Adicionalmente se presentan los componentes de visualización que en lugar de

estar en el mismo entorno de ejecución, está directamente ligado con el gestor Liferay.

Estos componentes involucrados con el almacenamiento de datos provenientes de twitter y la herramienta, están conectados con diferentes servidores de bases de datos, tal como se representa tenemos un servidor de base de datos relacional donde la base de datos de MySQL almacena los datos propios de la herramienta y la clasificación de las diferentes cuentas, la cual es generada por los diferentes usuarios. También se representa el servidor de bases de datos no relacional el cual contiene la base de datos Mongo que contiene la información de los perfiles de cada cuenta extraída y sus correspondientes líneas de tiempo las cuales son un conjunto de estados realizados por la cuenta específica y agrupados en una sola estructura.

### 6.3.2 Diagrama de Componentes.

Ilustración 2 - Diagrama de componentes.




En el diagrama de componentes podemos detallar los diferentes componentes de software de nuestra herramienta y como se relacionan en un mismo entorno de ejecución, adicionalmente a esto vemos la interfaz de inicio de sesión por parte del usuario la que lleva posterior mente un control establecido por el proceso de autenticación. Este control verificara el usuario y su rol frente al sistema de allí el sistema le permitirá ingresar al componente **Twitter Parametrization View** la vista

donde el usuario podrá generar la clasificación de las cuentas deseadas en las categorías y subcategorías que se definan.

Claramente el componente de vista de extracción, **Twitter Extraction View,** también se representa como accesible desde que se realice la autenticación, esto depende de la actividad que necesite realizar el usuario ya que allí tendrá acceso al componente que permitirá extraer información de las diferentes cuentas introduciendo sus identificadores. Como se puede apreciar, este componente está ligado al componente de modelo de datos no relacionales **Twitter Extractor Model** el cual almacenara todos los datos provenientes del componente de extracción.

Adicionalmente vemos una interfaz de integración con Liferay nuestra plataforma gestora de contenidos la cual nos permitirá gestionar las interfaces de la herramienta y se puede ver como el consumo de otro componente externo a la codificación de la herramienta, ya que el software de la plataforma es un producto o servicio el cual se decidió utilizar en el presente proyecto.

## 6.4 DESARROLLO (CODIFICACIÓN)

En la fase de codificación o desarrollo en su expresión máxima, se establecieron un conjunto de procesos para llevar a cabo la elaboración de cada componente de software. Lo primero en esta fase fue la definición simple de los componentes, de tal manera que al inicio se enfocó en las funcionalidades de Extracción, Almacenamiento relacional y Almacenamiento no relacional, lo que involucró el repaso de algunos conceptos trabajados en la fase de investigación.

Como primer paso como se planteó en el desarrollo metodológico, se realizó la construcción de un repositorio privado en GitHub para llevar un control de versiones en la codificación. Esto nos permite llevar un mejor control durante el desarrollo de la herramienta.

Conceptos de minería de datos, extracción y almacenamiento de grandes volúmenes de información fueron la base del desarrollo inicial, sin embargo lo primero a definir era la forma en la cual se iba a realizar la conexión con la red social Twitter y a partir de que métodos podría extraer la información necesaria definida por los requerimientos de la herramienta.

Por lo tanto el primer concepto y parte de desarrollo se basó en la conexión con el API de Twitter la cual le brinda al desarrollador la oportunidad de crear una aplicación la cual está ligada directamente con el servidor web de twitter y los datos de cuentas públicas.

Para crear dicha conexión primero se ingresó a la página de comunidad de desarrollo twitter llamada Twitter Developers, posterior mente se debe crear una App que simplemente requiere ingresar a una cuenta personal de twitter y posteriormente introducir alguna información referente a su objetivo y páginas de acceso.

Ilustración 3 - API de Twitter.

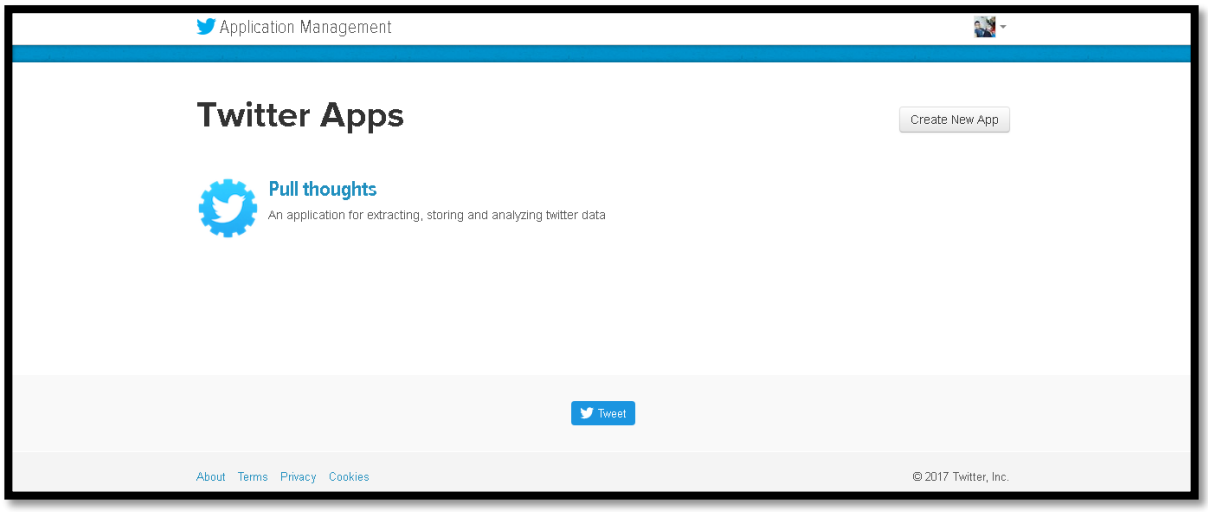
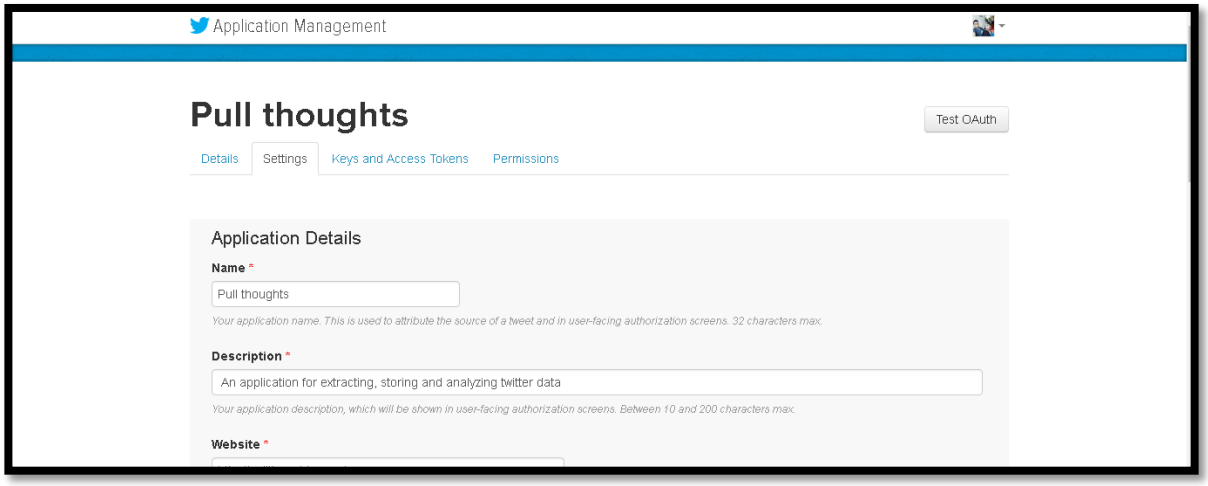



Ilustración 4 - Creación - API de Twitter.



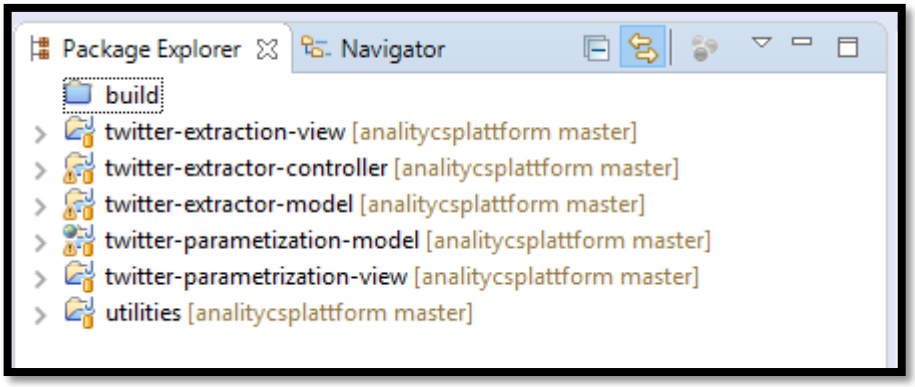
Posteriormente se generarán unas llaves de acceso al API y Tokens los cuales con los códigos con los cuales dentro de la herramienta yo puedo establecer la conexión con la red social.

Tal cual se define en el estado del arte del proyecto un API es una interfaz que me da la oportunidad de consumir un servicio web, en este caso la utilización de esta API nos permitirá consumir el servicio de twitter. Cabe aclarar que para poder establecer una extracción de datos de la red social no basta solo con crear dicha conexión con el API. Adicionalmente necesitamos utilizar una herramienta que ligado al lenguaje de programación nos permitirá desde nuestra aplicación realizar la conexión pasando las llaves de acceso y Tokens.

En este caso la herramienta fue programada bajo lenguaje Java y por lo tanto la herramienta seleccionada luego de una previa investigación de métodos, ventajas y aplicación fue la famosa librería Twitter4j, la cual también nos permite utilizar métodos de recolección de datos twitter lo que facilita el proceso de extracción ya que puede generarse de forma específica como lo menciona (Rubira, 2011) esta herramienta permite gestionar tweets, usuarios, hilos, listas, mensajes directos, relaciones, favoritos, subscripciones, bloqueos, realizar y guardar búsquedas, reportar spam y más estructuras que sin lugar a duda proveen al usuario de la información necesaria para recolectar los datos de cuentas específicas.

Su implementación es realmente sencilla, retomando el ciclo de desarrollo del proyecto, se decide crear en el IDE Eclipse tres proyectos independientes (estructura desacoplada) los cuales representarán los componentes principales de extracción y almacenamiento tanto relacional como no relacional.

Ilustración 5 - Componentes en proyectos



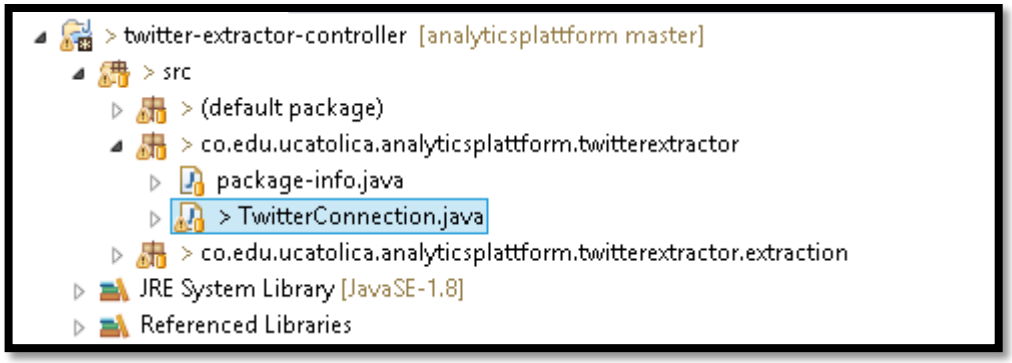
Como se puede ver se identifica la sincronización con el repositorio de Github en el primer componente, en cuanto a los otros dos componentes se puede apreciar el símbolo de almacenamiento. Esto debido a que se crearon con configuración JPA.

### 6.4.1 Componente de extracción.

Posteriormente en el componente de extracción twitter-extractor-controller creamos los paquetes necesarios los cuales están debidamente documentados con archivos de información (package-info) en este caso uno para la conexión a través del API de twitter y otro para la clase de extracción.

En el paquete de conexión creamos una clase TwitterConnection, en donde con ayuda de la librería twitter4j pasamos a la API los datos de acceso generados, los cuales fueron almacenados y posteriormente pueden ser accedidos para enviar los diferentes datos de acceso.

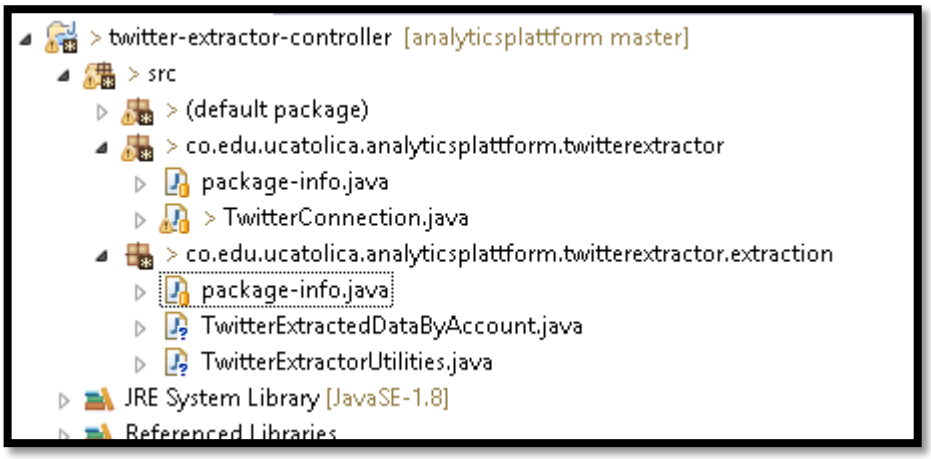
Ilustración 6 - Componente de extracción




Al realizar la conexión con el API de Twitter y tener acceso a la red social y a los métodos que nos provee la librería twitter4j anteriormente agregada al Build Path del componente de software, se puede iniciar con el proceso de codificación enfocado a la extracción de datos.

Para definir la funcionalidad, creamos dos clases en el paquete definido para la extracción (.extraction)

Ilustración 7 - Componente de extracción.




Una clase contiene la programación necesaria para extraer la información de cada cuenta requerida **TwitterExtractedDataByAccount**, esta clase depende de los métodos de la otra clase creada en este paquete **TwitterExtractorUtilities**, esto debido a que esta última clase tiene los métodos principales de extracción en los formatos necesarios para que la clase TwitterExtractedDataByAccount los consuma y genere así un conjunto ordenado de datos provenientes de cada una de las cuentas de una manera consolidada y más formal.

En cada método de extracción se definen los parámetros requeridos para su ejecución.

#### 6.4.1.1 Métodos de TwitterExtractorUtilities.

- **extractTimeline:** Método que recibe por parámetros un objeto que representa la conexión con la red social y una cadena llamada account name que representa el identificador de la cuenta a la cual se le extraerá su timeline.

  Este genera una lista de estados de cada cuenta los cuales se extraen por 16 páginas cada una con 200 estados propios de la cuenta especificada. Devolverá la lista tipo Status o un mensaje de error especificando que no pudo obtenerse el timeline.

- **extractTimelineAndStoreJSON** : Método booleano que recibe por parámetros un objeto que representa la conexión con la red social, una cadena con la ruta de un archivo JSON generado y una cadena de texto que corresponde al account name o nombre de la cuenta de twitter a la cual se le quiere extraer la información en este caso el timeline.

  Su funcionalidad consiste en extraer el timeline de la cuenta identificada con el account name recibido y posteriormente después de convertir estos datos a formato JSON, y almacenarlos en un archivo de texto plano con el nombre de la cuenta y que se encuentra en la dirección indicada en la cadena llamada "filepath". Devuelve un valor booleano de confirmación.

- **loadStatusfromJSON** : Método que recibe por parámetros un objeto que representa la conexión con la red social, una cadena con la ruta de un archivo JSON generado y una cadena de texto que corresponde al file name o nombre del archivo el cual contiene el timeline de la cuenta en formato JSON.

  Su funcionalidad consiste en extraer los estados del timeline de la cuenta previamente generado y almacenado en un archivo con formato JSON. Devuelve un objeto tipo Status o un error informativo.

- **loadStatusesfromJSON:** Método que recibe por parámetros un objeto que representa la conexión con la red social, una cadena con la ruta de un archivo JSON generado y una cadena de texto que corresponde al file name o nombre del archivo el cual contiene el timeline de la cuenta en formato JSON.

  Su funcionalidad consiste en extraer los estados del timeline de la cuenta previamente generado y almacenado en un archivo con formato JSON. Devuelve una lista tipo Status con todos los estados del timeline propio de la cuenta o un error informativo.

- **getFollowingUsers:** Método que recibe por parámetros un objeto que representa la conexión con la red social y una cadena llamada account name que representa el identificador de la cuenta a la cual se le extraerá una lista con los perfiles de cada usuario seguido por la cuenta, el método devuelve esta lista o un error informativo.

- **getFollowers:** Método que recibe por parámetros un objeto que representa la conexión con la red social y una cadena llamada account name que representa el identificador de la cuenta a la cual se le extraerá una lista con los perfiles de cada usuario que sigue la cuenta, el método devuelve esta lista o un error informativo.

- **getHashTags:** Método que recibe por parámetro un objeto tipo status que representa un estado de una cuenta específica con el cual opera y devuelve una lista de cadenas de texto con los hashtags realizados en ese estado específico.

#### 6.4.1.2 Métodos de TwitterExtractedDataByAccount.

- **initalize:** Método que crea la conexión con twitter e inicializa todas las variables a utilizar en la clase.

- **TwitterExtractedDataByAccount:** Posee tres métodos constructores con el mismo nombre, uno sin parámetros, uno que recibe el nombre de la cuenta (accountName) y el otro recibe la ruta de un archivo JSON con el timeline de alguna cuenta específica y el nombre del archivo, el cual también es el nombre de la cuenta. Este último método carga de forma automática los estados de dicha cuenta en una lista tipo status para su posterior operación. Los tres constructores llaman el método initalize ().

- **extractTimelineFromTwitter:** Método que recibe como parámetro el nombre de la cuenta (accountName) y utilizando el método de la clase TwitterExtractorUtilities llamado extractTimeline almacena el timeline en una lista de tipo status, posteriormente calcula su tamaño y lo guarda en una variable llamada totaltweets. Retorna un valor booleano que verifica el éxito o fracaso de su uso.

- **populateEstructureData:** Método que se encarga de generar información variada del timeline, status y de la propia cuenta y almacena esta información en diferentes variables.

  Primero se define una variable result de tipo booleano que arrojara un valor dependiendo de si se pudo extraer el timeline de una cuenta específica.

Posteriormente calcula la cantidad de veces que cada estado del timeline fue citado y almacena este valor en la variable entera totalQuotedTweets, también calcula el número de retweets obtenidos por cada estado almacenado en la variable entera totalReTweets.
Luego obtiene los hashtags de cada estado, los cuales son almacenados en una lista string para la cual se utiliza el método getHashTags de la clase TwitterExtractorUtilities y también genera listas de perfiles de seguidores y seguidos donde se implementan los métodos getFollowers y getFollowingUsers de la clase TwitterExtractorUtilities.

**Nota:** Los demás métodos consisten en los set y get de cada variable definida en la clase.

### 6.4.2 Componente del Modelo de extracción (Base de datos No relacional).

Una vez obtenidos los datos solicitados de las cuentas es necesarios almacenarlos, para esto se decide utilizar un modelo de datos no relacional debido a la cantidad y estructura de los datos. El sistema de base de datos seleccionado para cumplir con este papel fue MongoDB, el cual está orientado a documentos lo cual se adapta a los datos recolectados los cuales en algunos casos simplemente podrían ser almacenados directamente en la base de datos pero para poder definir un modelo con una estructura concreta se ha construido un componente de software llamado **twitter extractor model** que permite almacenar información de los perfiles de cada usuario y también su línea de tiempo ajustada a los límites de utilización del API mencionados en la descripción del componente de extracción.

Teniendo en cuenta estas funcionalidades a implementar, se construye dentro del proyecto en el IDE, dos paquetes nuevos (debidamente documentados), donde en el primero se albergará la clase principal, y en el segundo se crearan tres clases diferentes.

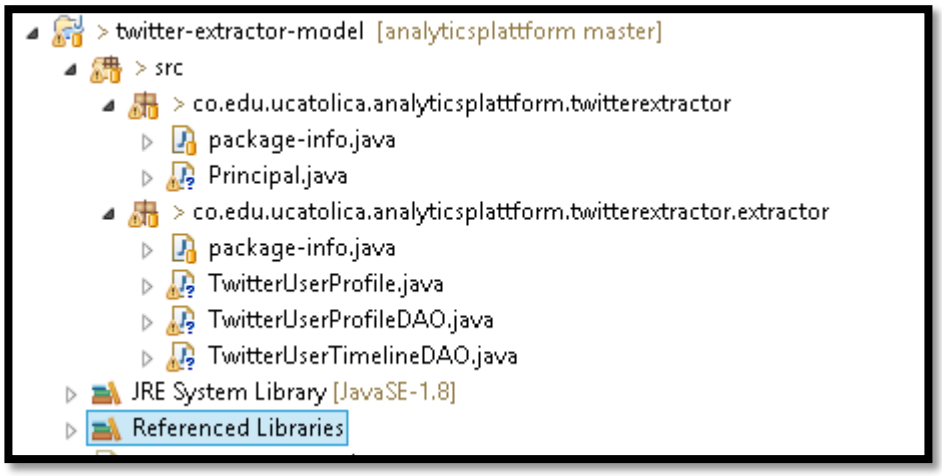
La primer clase va a definir la estructura del perfil de un usuario, es decir la información relevante del perfil de cada usuario será extraída y almacenada con la estructura propuesta en esta clase, la cual se asemeja al concepto de entidad en bases de datos. Esta “entidad” llevara el nombre de **TwitterUserProfile**.

La segunda clase será lógicamente será el DAO de esta entidad, la clase **TwitterUserProfileDAO** (Data Acces Object) es suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos. Allí se

establece la conexión con la base de datos MongoDB y se declaran las diferentes operaciones de un CRUD a través de sus métodos.

La tercer clase de este componente representa también un DAO pero en este caso perteneciente a los timelines extraídos. **TwitterUserTimelineDAO** es una clase que permite el almacenamiento de los timelines de cada cuenta y define las diferentes operaciones de un CRUD a través de sus métodos.

Ilustración 8 - Componente de almacenamiento (No relacional)




A continuación (Véase ilustración 9 Clase TwitterUserProfile) se expone un fragmento de la clase TwitterUserProfile (entidad), donde se aprecia su estructura a través de los datos definidos como apropiados para ser parte de un perfil de usuario.

Ilustración 9 - Clase TwitterUserProfile

```
public class TwitterUserProfile implements Serializable{

    /**
     *
     */

        private long id;

        private String accountName;

        private String screenName;

        private String profileimageurl;

        private String profilebannerurl;

        private long noFollowers;

        private byte[][] profilePicture;

        private byte[][] bannerPicture;

        private User userReference;

        /**
         * Status of the User (active or inactive)
         */
        private Integer status = 1;



        private boolean downloadPicture(){
            return true;
        }
```

En cuanto a la clase TwitterUserProfileDAO cabe aclarar que se implementa la conexión con la base de datos, luego con la colección de perfiles de usuario y se definen los siguientes métodos, métodos que no serán descritos en detalle ya que hacen referencia a las operaciones comunes de un CRUD.

Simplemente enfocándonos en los métodos, primero se define un método de creación de perfil llamado **createTwitterUserProfile** el cual inserta el perfil recibido por parámetro en la base de datos.

Luego se definen los métodos de lectura los cuales seleccionan desde la colección los perfiles solicitados, existe un método que los selecciona por el id del usuario **getTwitterUserProfileById**, otro que los selecciona por screenName **getTwitterUserProfileByScreenName** y un último método de lectura que selecciona todos los perfiles de la colección **ReadAllTwitterUserProfiles**. Los dos primeros reciben el perfil, el último no recibe ningún objeto o variable por parámetro.

En cuanto al método de actualización nombrado **updateTwitterUserProfileById**, es un método que recibe por parámetro el perfil a actualizar y el perfil con la información que se quiere reemplazar. La funcionalidad está en identificar el perfil a actualizar por el id del objeto y posterior mente eliminarlo e insertar el perfil nuevo con todos sus nuevos atributos.

Finalmente para los métodos de eliminación al igual que los de escritura se puede realizar la eliminación por Id, ScreenName o en general eliminar toda la lista de perfiles estos métodos se nombraron como **deleteTwitterUserProfileById**, **deleteTwitterUserProfileByScreenName** y **deleteAllTwitterUserProfiles**.

Por otro lado en cuanto a la clase TwitterUserTimelineDAO es una clase que en cuanto a funcionalidad es muy similar al DAO de perfiles sin embargo esta clase permite crear una colección por cada timeline extraído, no obstante, si el timeline de una cuenta ya fue extraído previamente, el DAO no genera una colección nueva, en lugar de esto se van añadiendo los diferentes timelines en la colección construida la primer vez que se realizó el almacenamiento.

Los métodos son similares a los de TwitterUserProfileDAO, con la diferencia de que la mayoría de ellos que realiza operaciones de manera singular reciben como parámetro un objeto tipo status representante de un estado de una cuenta específica y también se diferencia en que este DAO no posee método de actualización ya que debido a que un objeto tipo status tiene una gran cantidad de información que no tiene sentido modificar, preferiblemente se debe eliminar el estado que presente algún problema o así lo requiera el usuario.

Adicionalmente los métodos de operaciones masivas como eliminar todos los timelines o seleccionarlos todos tampoco se tienen en cuenta ya que al realizar estas operaciones con screenname como parámetro obtendré el mismo resultado para cada cuenta.

Los métodos de esta clase son los siguientes:

- **InsertStatusToTwitterUserTimeline** – Método de inserción, recibe Status por parámetro.

- **getStatusToTwitterUserTimelineById** – Método de lectura por id recibe Status por parámetro (lee un estado del timeline).

- **getStatusToTwitterUserTimelineByScreenName**– Método de lectura por screenname recibe Status por parámetro (lee un timeline).

- **deleteSatusToTwitterUserTimelineById**– Método de eliminación por id recibe Status por parámetro (elimina un estado del timeline).

- **deleteTwitterUserTimelineByScreenName** – Método de eliminación por screenname recibe Status por parámetro (elimina una timeline).

Estas operaciones permiten llevar un control de los diferentes perfiles de cada cuenta y cada uno de sus timelines y status almacenados en las diferentes colecciones de la base de datos no relacional.

### 6.4.3 Componente del modelo de parametrización (Base de datos relacional).

Por último, el componente de almacenamiento relacional llamado **User-Parametrization-Model**, es el componente de software enfocado al modelo de parametrización y clasificación de cada una de las cuentas en subcategorías y categorías. Un ejemplo de esta clasificación puede verse representado en la siguiente ilustración:

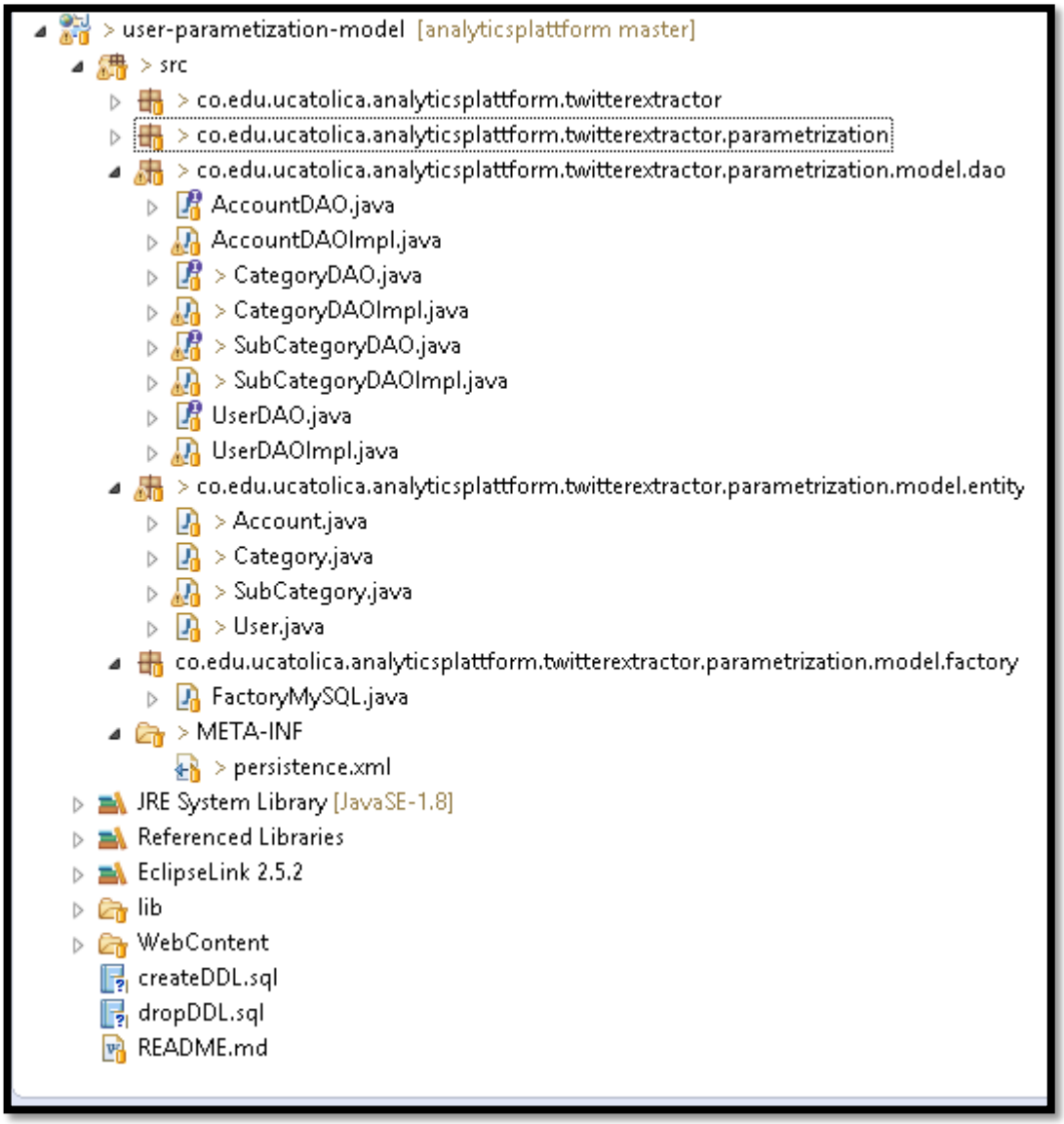
Ilustración 10 - Clasificación de cuentas

| Cuentas Twitter | Categoria | Subcategoria |
|---|---|---|
| JuanManSantos | Presidencia | Presidente |
| MinjusticiaCo | Presidencia | Ministerio |
| reintegracion | Presidencia | AgenyUni |
| UnidadVictimas | Presidencia | AgenyUni |
| centromemoria | Presidencia | AgenyUni |
| MinSaludCol | Presidencia | Ministerio |
| MinInterior | Presidencia | Ministerio |
| RafaelPardo | Presidencia | Ministerio |
| riveraguillermo | Presidencia | Ministerio |
| infopresidencia | Presidencia | Presidente |
| ComisionadoPaz | Presidencia | Ministerio |
| FuerzasMilCol | GALegales | FFMM |
| COL_EJERCITO | GALegales | FFMM |
| armadacolombia | GALegales | FFMM |
| FuerzaAereaCol | GALegales | FFMM |
| CeDemocratico | Partido Politico | Centro Democratico |

El componente posee un paquete para las diferentes entidades creadas en este caso se hace uso del concepto de JPA en java, otro para los diferentes DAO de cada entidad y sus implementaciones y un último paquete para el Factory de Mysql el cual es el gestor de bases de datos utilizado en este componente debido al modelo relacional que representan los datos de las entidades. Externamente en

el fichero META-INF se crea una persistencia donde se especifican los datos de conexión con el gestor (autenticación) y su base de datos

Ilustración 11 - Componente de almacenamiento (Relacional)




Haciendo énfasis en las diferentes entidades de este modelo, cada una tiene atributos que las identifican y relacionan entre ellas, esta relación puede detallarse mejor en el diagrama entidad relación (Véase el Anexo A ).

Por ahora cabe resaltar el contexto general y las relaciones entre entidades. La definición de la entidad User o usuario, comprende al individuo que hará uso de la herramienta, la información de esta entidad será importante a la hora de realizar una autenticación e indagación sobre las personas con acceso a la herramienta.

Sus atributos se conforman por un identificador único tipo double auto incrementable, un nombre de usuario y una contraseña (datos de autenticación) tipo string, un estado el cual se define como estado del usuario en el sistema (1= activo 0= inactivo) tipo integer, y un tipo de usuario el cual representa el rol del usuario en el sistema, el cual es de tipo string. Los métodos tratados en esta clase son todos los get y set de cada atributo.

Las operaciones de su DAO se pueden apreciar en la siguiente ilustración:

Ilustración 12 - Métodos de la clase UserDAO.

UserDAO.java

```
1  package co.edu.ucatolica.analyticsplattform.twitterextractor.parametrization.model.dao;
2
3  import java.util.List;
11
12 public interface UserDAO {
13
14     public boolean createUser(User user);
15
16     public boolean updateUser(User user);
17
18     public boolean deleteUser(User user);
19
20     public User getUserbyId(double iduser);
21
22     public User getUserbyUserName(String username);
23
24     public List<User> getAllUsers();
25
26 }
27
```

La implementación de este DAO necesita de la conexión a la base de datos, utilización del Factory creado y también comprende el concepto de transacciones para cada operación, además de utilizar objetos como administrador de entidades, query (consultas) para realizar las operaciones correspondientes y especificadas en la imagen anterior, en la cual también se aprecia los diferentes parámetros que reciben.

En cuanto a la entidad Account, se define como la entidad que representa una cuenta especifica de twitter, a esta entidad se le definieron como atributos un id auto incrementable de tipo double, un accountname el cual es un string que

representa el nombre de la cuenta, un screenname el cual es el identificador de la cuenta (tipo string), una descripción (tipo string) y un estado que al igual de todas las entidades sirve para definir su estado en la base de datos (tipo integer).

Las operaciones de su DAO se pueden apreciar en la siguiente ilustración:

Ilustración 13 - Métodos de la clase AccountDAO.

AccountDAO.java

```
1  package co.edu.ucatolica.analyticsplattform.twitterextractor.parametrization.model.dao;
2
3  import java.util.List;
8
9  public interface AccountDAO {
10
11     public boolean createAccount(Account account);
12
13     public boolean updateAccount(Account account);
14
15     public boolean deleteAccount(Account account);
16
17     public Account getAccountbyId(double id);
18
19     public Account getAccountbyAccountName(String accountName);
20
21     public List<Account> getAllAccounts();
22
23     public List<Account> getAccountsByScreenName(String screenName);
24
25     public Category getCategorybyAccountName(String accountName);
26
27     public SubCategory getSubCategorybyAccountName(String accountName);
28 }
29
```

Podemos apreciar las diferentes operaciones y los parámetros que recibe cada método, crear, leer, actualizar y eliminar, son las operaciones generales y las cuales varían en cuanto al parámetro de identificación. Adicionalmente se crearon dos métodos que se relacionan con las entidades Category y Subcategory, obteniendo a cuál de las categorías y subcategorías definidas pertenece esa cuenta.

La entidad Subcategory, se define como la entidad que representa una subcategoría especifica definida según los requerimientos funcionales, a esta entidad se le definieron como atributos un id auto incrementable de tipo double, un subcategoryname el cual es un string que representa el nombre de la subcategoría, una descripción (tipo string), una lista de cuentas pertenecientes a la subcategoría (tipo List<Account>) y un estado que al igual de todas las entidades sirve para definir su estado en la base de datos (tipo integer).

Las operaciones de su DAO se pueden apreciar en la sigu  iente imagen:

Ilustración 14 - Métodos de la clase SubCategoryDAO.

SubCategoryDAO.java

```
1  package co.edu.ucatolica.analyticsplattform.twitterextractor.parametrization.model.dao;
2
3  import java.util.List;
15
16 public interface SubCategoryDAO {
17
18     public boolean createSubCategory(SubCategory subcategory);
19
20     public boolean updateSubCategory(SubCategory subcategory);
21
22     public boolean deleteSubCategory(SubCategory subcategory);
23
24     public SubCategory getSubCategorybyId(double idsubcategory);
25
26     public SubCategory getSubCategorybySubCategoryName(String subcategoryname);
27
28     public List<SubCategory> getAllSubCategories();
29
30     public List<Account> getAccountsbySubCategoryName(String subcategoryname);
31
32 }
```

Podemos apreciar las diferentes operaciones y los parámetros que recibe cada método, crear, leer, actualizar y eliminar, son las operaciones generales y las cuales varían en cuanto al parámetro de identificación. Adicionalmente se crearon dos métodos que se relacionan con las entidades Category y Account, obteniendo a cuál de las categorías definidas pertenece y las cuentas que pertenecen a dicha subcategoría.

Por último la entidad Category, se define como la entidad que representa una categoría especifica definida según los requerimientos funcionales, a esta entidad se le definieron como atributos un id auto incrementable de tipo double, un categoryname el cual es un string que representa el nombre de la categoría, una descripción (tipo string), una lista de subcategorías pertenecientes a la categoría (tipo List<Subcategory>) y un estado que al igual de todas las entidades sirve para definir su estado en la base de datos (tipo integer).

Las operaciones de su DAO se pueden apreciar en la siguiente imagen:

Ilustración 15 - Métodos de la clase CategoryDAO.

CategoryDAO.java

```java
package co.edu.ucatolica.analyticsplattform.twitterextractor.parametrization.model.dao;

import java.util.List;

/**
 * @author CamiloBM
 *
 */
public interface CategoryDAO {

    public boolean createCategory(Category category);

    public boolean updateCategory(Category category);

    public boolean deleteCategory(Category category);

    public Category getCategorybyId(double idcategory);

    public Category getCategorybyCategoryName(String categoryname) ;

    public List<Category> getAllCategories();

    public List<SubCategory> getSubCategoriesbyCategoryName (String categoryname);

    public List<Account> getAccountsbyCategoryName(String categoryname);

}
```

Podemos apreciar las diferentes operaciones y los parámetros que recibe cada método, crear, leer, actualizar y eliminar, son las operaciones generales y las cuales varían en cuanto al parámetro de identificación. Adicionalmente se crearon dos métodos que se relacionan con las entidades Subcategory y Account, obteniendo las subcategorías definidas que pertenecen a ella y las cuentas que pertenecen a dicha categoría también.

En cuanto a los componentes de **utilidades** y **construcción**, son componentes desarrollados previamente por el asesor implicado en el proyecto y en general se describen como utilidades para la programación de los componentes y sus diferentes métodos adicionalmente de la construcción automática o despliegue del entorno en el gestor de contenidos.

# 7 RESULTADOS

Como consolidación de la fase de desarrollo y parte de resultados del proyecto, se presenta a continuación una etapa de aplicación de pruebas donde se evalúa el rendimiento de la herramienta y su correcta funcionalidad por cada uno de sus componentes.

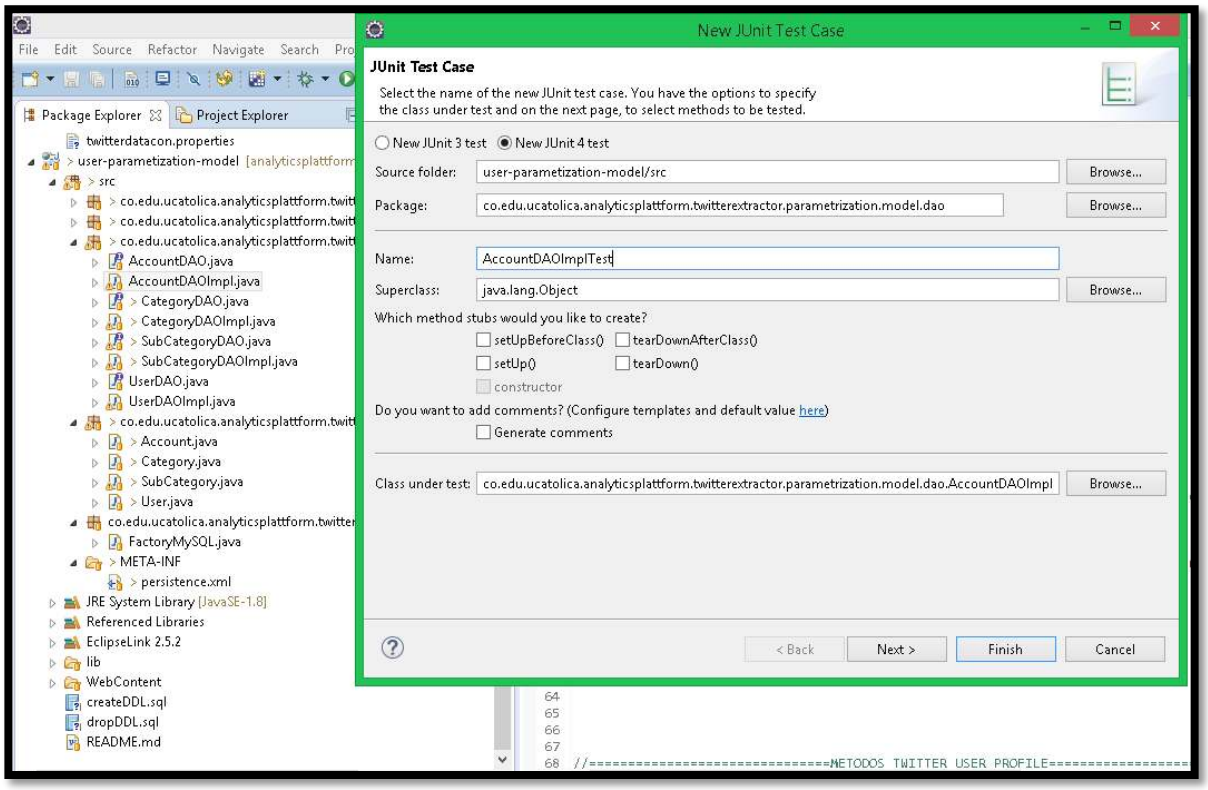
Esta etapa del proyecto permite determinar si la herramienta cumple con los objetivos propuestos o si ésta realiza correctamente su labor durante la ejecución. Consiste en una fase donde propuestos los diferentes componentes de la fase de codificación, el siguiente paso es implementar las pruebas necesarias, generar los reportes correspondientes y compartir los resultados obtenidos.

En cuanto a las pruebas unitarias, estas son una práctica que permite verificar la correcta funcionalidad de los diferentes módulos de software de un programa (uno por uno), esto tal como se menciona en (IA, 2014) permite que la calidad del desarrollo sea mucho mayor y nos permite asegurarnos que los diferentes módulos se comportan de la manera planificada frente a diferentes situaciones plasmadas en las pruebas.

Para la realización de estas pruebas en la herramienta propuesta, se optó por la herramienta JUnit, la cual basado en (IA, 2014) es una librería para Java que permite evaluar el funcionamiento de las clases y métodos de un programa a través de casos de prueba organizados en suites de prueba definidas. Su integración con java en este caso trabajando con el IDE de trabajo eclipse al igual que twitter4j, consiste en la implementación de un fichero JAR, el cual contiene las clases necesarias a la hora de implementar y ejecutar los casos de prueba. Posteriormente luego de realizar las configuraciones necesarias se crean lo casos de testeo o de prueba JUnit test case e implementa las diferentes pruebas unitarias con valores definidos para cada una de las funcionalidades de la herramienta.

Luego de configurar JUnit en nuestro eclipse, basta con un clic derecho sobre la clase en este caso la entidad a la que vamos a generarle un testeo de sus métodos, posteriormente hacer clic en New y luego en JUnit Test Case.

Ilustración 16 - Creación caso de prueba en JUnit



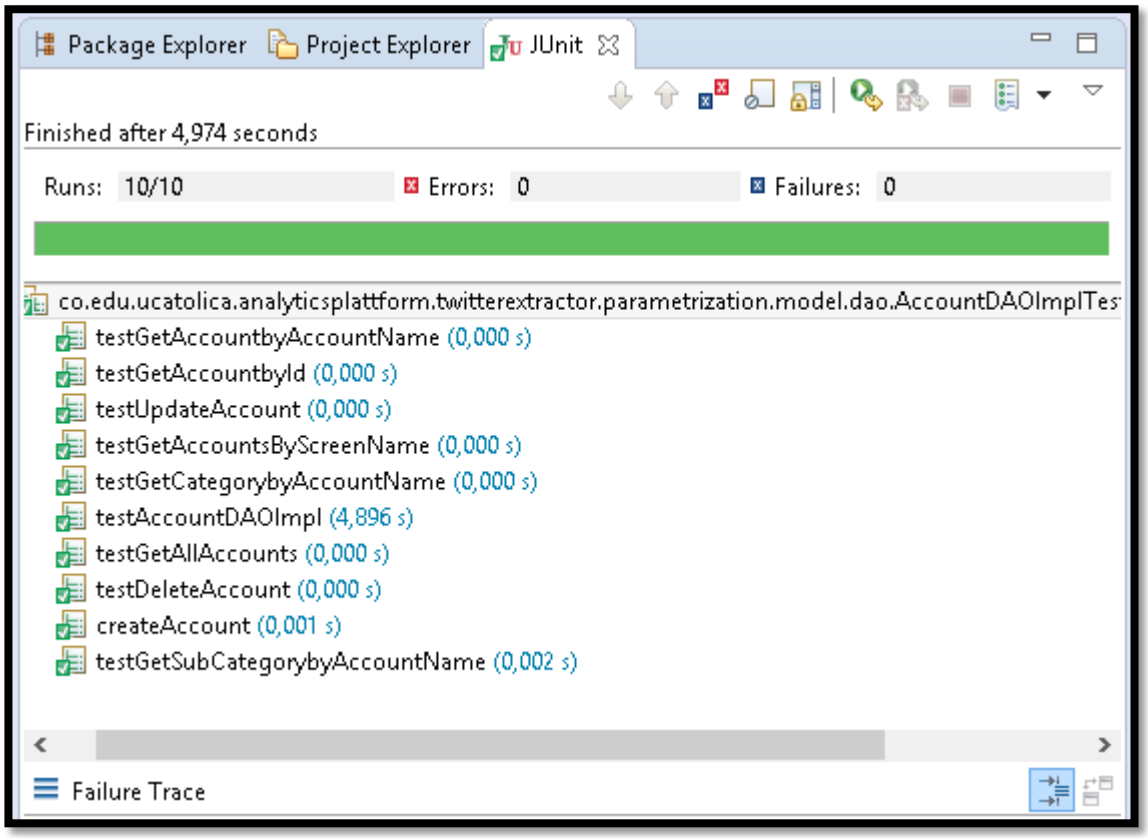

En este caso para probar nuestro software se realizaron casos de pruebas para cada uno de los métodos de cada componente explicado previamente en el anterior ítem, casos de testeo de entidades como en el caso de nuestro modelo relacional programado en el componente user-parametrization-model, no se llevan a cabo ya que son solo las definiciones de la entidad. Sin embargo las implementaciones de los DAO de cada entidad y sus diferentes métodos fueron probados para constatar que las operaciones que lo conforman se estén realizando de una manera correcta.

Como ejemplo de estas pruebas decidimos tomar el caso de prueba del DAO de la entidad de cuentas (Account) llamado AccountDAOImpl donde se probaron los 10 métodos u operaciones, los cuales fueron programados de la misma manera para los demás DAO de cada entidad.

El resultado o reporte arrojado por la herramienta es el siguiente:

**AccountDAOImplTest**

Ilustración 17 - Caso de prueba AccountDAO.




Notamos que el mayor consumo de tiempo en los test que sin embargo es un tiempo óptimo se genera en el método principal donde creamos el Factory el cual a su vez crea el EntityManager haciendo referencia a la persistencia de la base de datos relacional.

Resultado en el cual podemos apreciar cada método de la implementación del DAO de la entidad, su tiempo de ejecución de prueba y si fueron casos exitosos o alguno fallo a través de la prueba.

Los fallos principalmente se generan ya que el valor definido como esperado en la prueba no se cumple al momento de ejecutarla.

Para cada una de las entidades de este modelo generamos este mismo proceso de pruebas obteniendo resultados positivos y tiempos realmente cortos y similares.

A continuación los resultados de las demás pruebas en el modelo relacional.

Ilustración 18 - Caso de prueba CategoryDAO

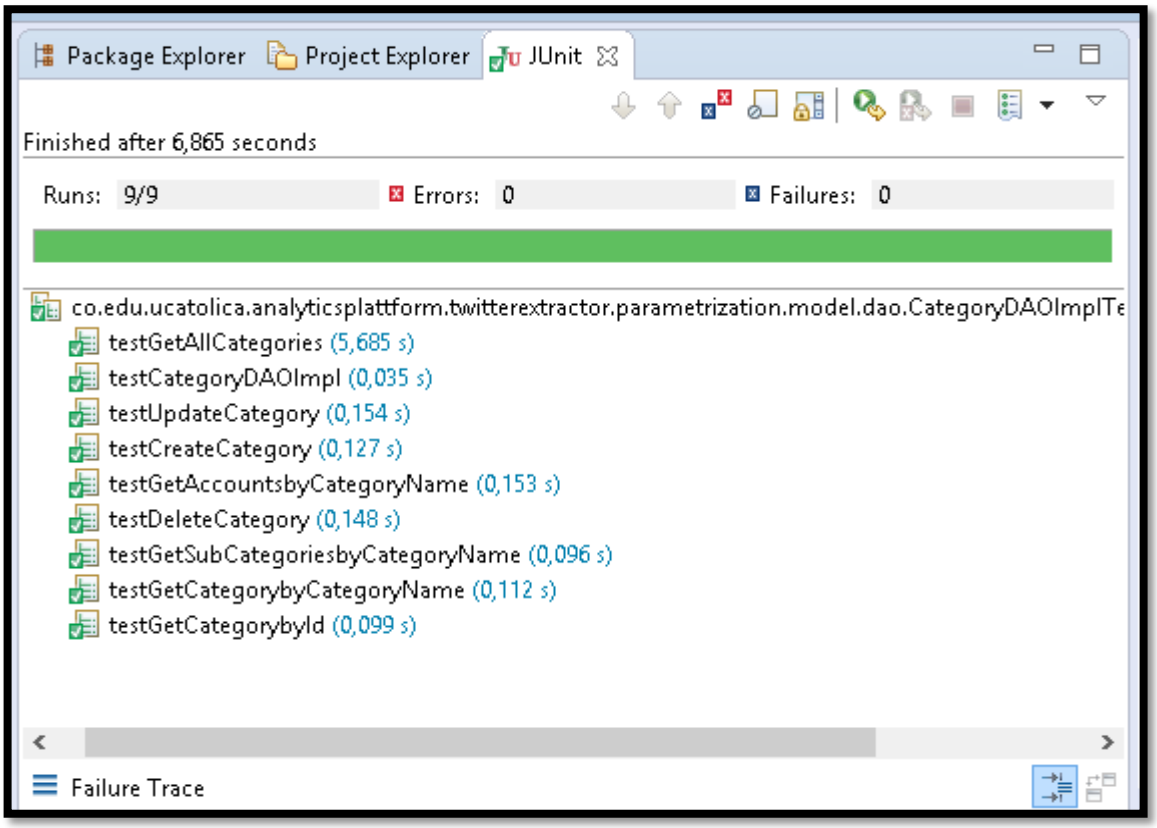



Ilustración 19 - Caso de prueba UserDAO.

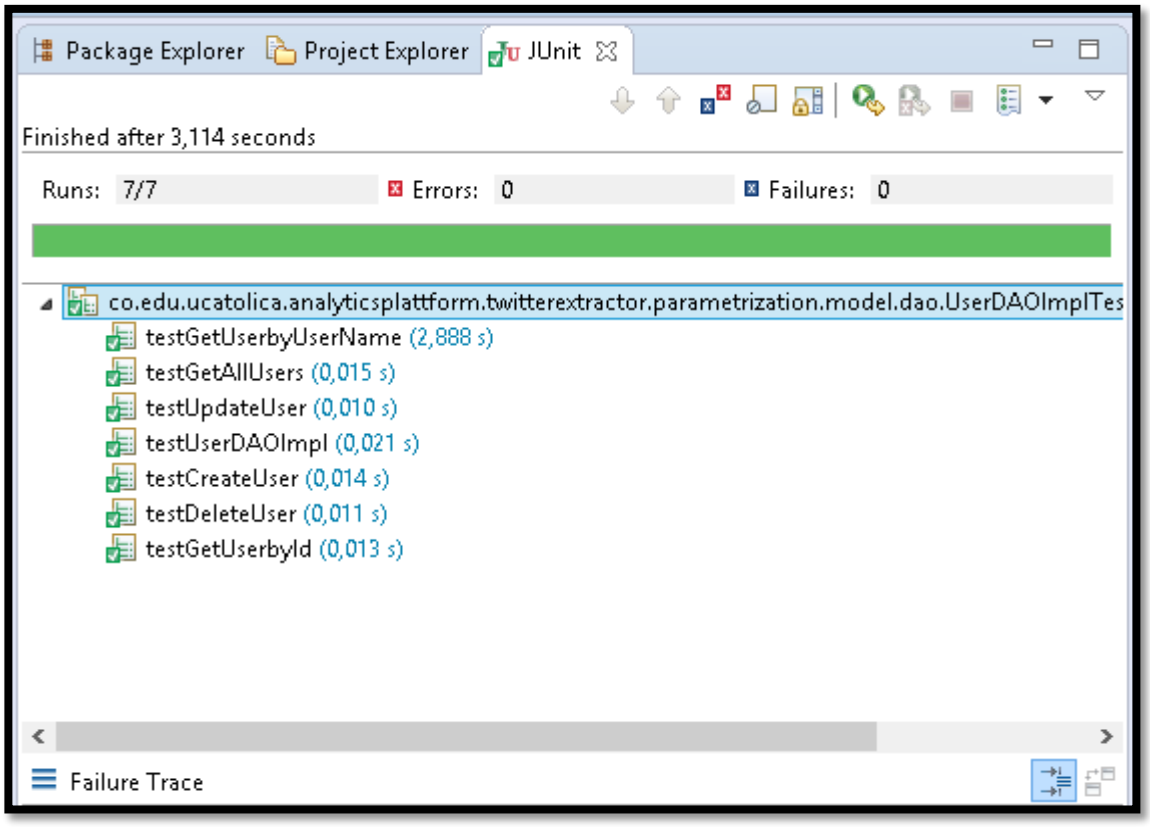
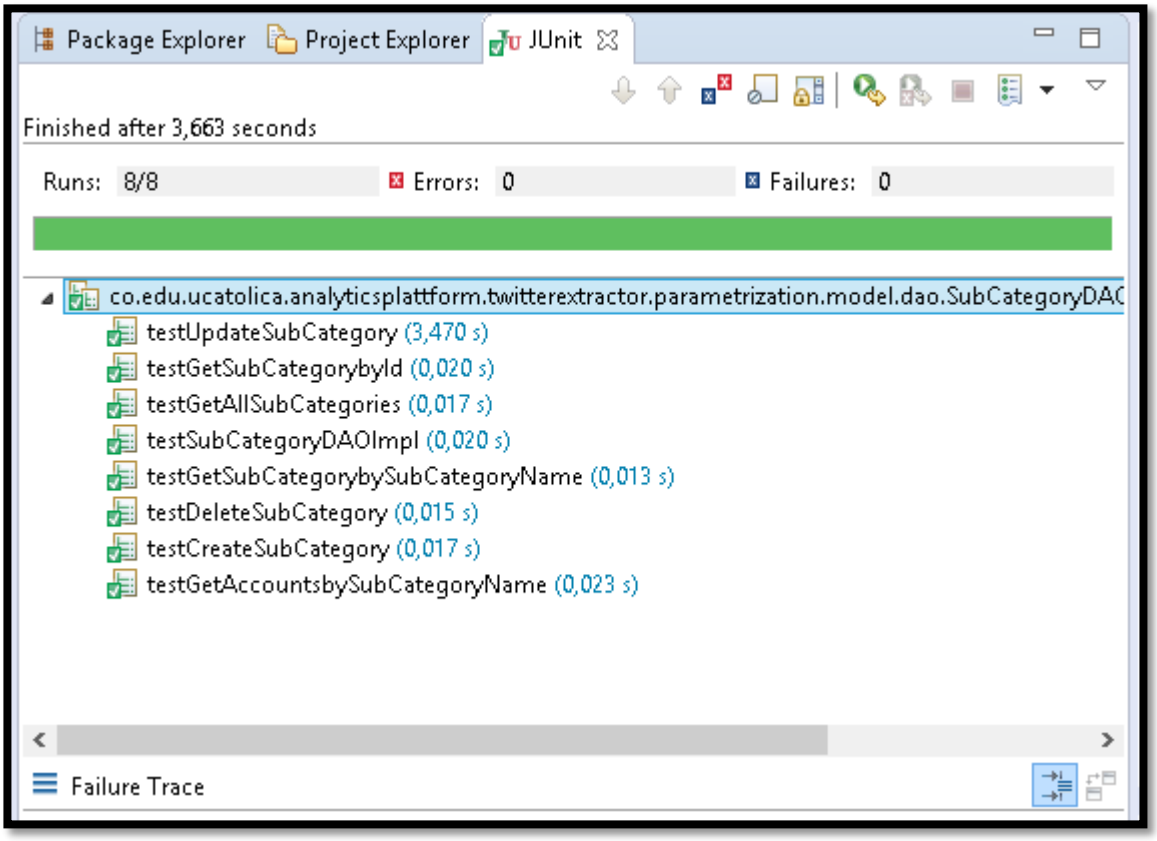


Ilustración 20 - Métodos SubCategoryDAO




Adicionalmente se generaron pruebas de cada DAO de una forma manual, que requirió la realización una clase llamada principal, la cual crea cada entidad pasando valores directamente a cada método (por parámetro) y en caso de fallo genera mensajes los mensajes de error correspondientes. Esta aunque es una forma manual de hacer pruebas permite junto con la aplicación de las pruebas unitarias una mayor efectividad a la hora de realizar el testeo general de la herramienta.
En un corto ejemplo podemos definir la creación de una cuenta implementando el DAO y pasando los atributos de forma manual según el tipo de dato requerido para cada uno de ellos. Estas pruebas tienen como resultado confirmación por parte de la base de datos a la cual se añaden los elementos y por parte de la herramienta al implementar correctamente cada método.

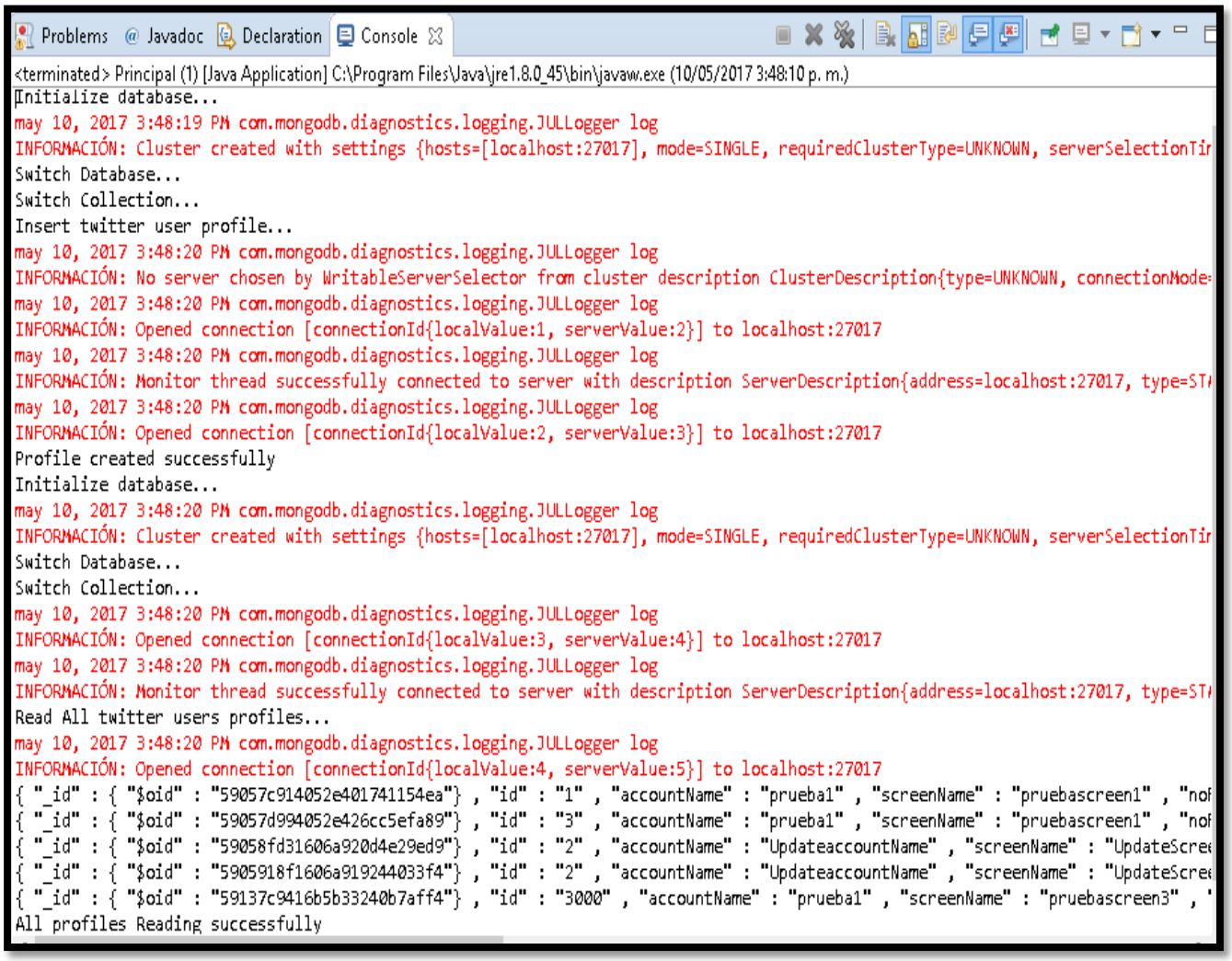
Luego de verificar la correcta funcionalidad del componente, procedemos a realizar el mismo proceso de testeo en el componente twitter-extractor-model, donde se realizara esta prueba en el DAO de perfiles y líneas de tiempo.

En este caso también se ha hecho adicionalmente a las pruebas unitarias un conjunto de pruebas manuales a través de una clase principal donde se encuentran las definiciones de parámetros y llamados a los diferentes métodos.

Específicamente para este modelo el cual almacena en una base de datos no relacional la información de perfiles y línea de tiempo los parámetros que se pasan de forma manual para este tipo de pruebas son de tipo entidad TwitterUserProfile en el caso de los perfiles de usuario, y tipo Status para el almacenamiento de timelines o líneas de tiempo, este último almacenamiento se realiza en un ciclo de extracción de estados (estatus) que se irán almacenando de forma continua en la base de datos.

Por su parte estas pruebas nos generan como resultado confirmaciones de la conexión con la base de datos e implementación correcta de los métodos. A continuación un ejemplo del resultado de creación de un perfil de usuario y la selección de todos los perfiles existentes en la base de datos no relacional:

Ilustración 21 - Prueba local de almacenamiento de perfiles



Los reportes de las pruebas unitarias de este componente, se realizan de la misma forma que en el componente anterior salvo que en este caso se representan las pruebas realizadas para el DAO de perfiles de usuario de twitter y para el DAO de línea de tiempo.

Ilustración 22 - Caso de prueba TwitterUserProfileDAO

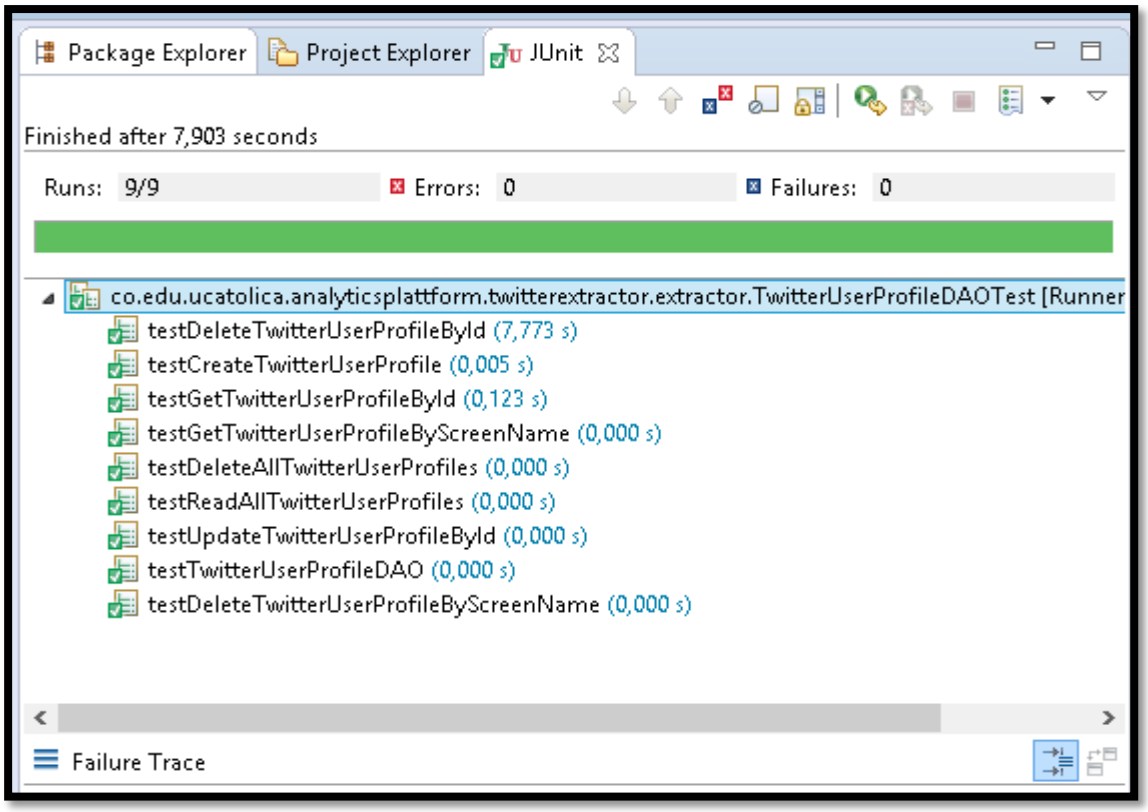
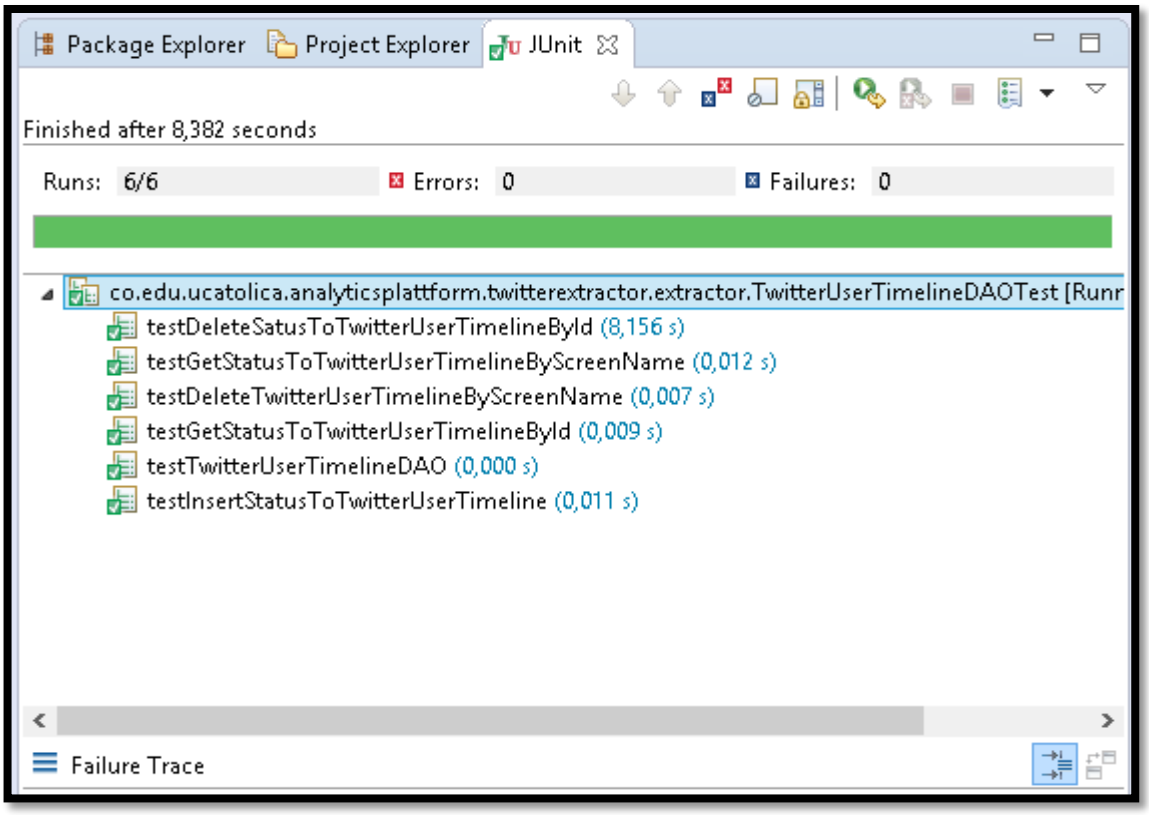


Ilustración 23 - Caso de prueba TwitterUserTimelineDAO




Donde cada uno de los métodos de ambas clases del modelo no relacional es testeado en los respectivos componentes o casos de prueba y las pruebas definidas se reportan como exitosas.

Se establece una variedad en el tiempo de ejecución albergada en el primer método de testeo incrementando su tiempo de ejecución notablemente en comparación con los demás métodos, esto es debido a que en el primer método de testeo se realiza la conexión a la base de datos NoSql y a la colección correspondiente, pero esta conexión se mantiene para los demás métodos, ya que los elementos de dicha colección son asignados a objetos de un tipo específico los cuales mantienen y hacen referencia a la conexión ya establecida.

Continuando con las pruebas, es el turno del componente de extracción, un componente que tiene un valor de prueba diferente a los demás, claramente aquí ya no estamos hablando de almacenamiento, es extracción de datos, conexión con el API externo, filtros y demás métodos de estructuración de datos.

Ilustración 24 - Caso de prueba TwitterExtractedDataByAccount.

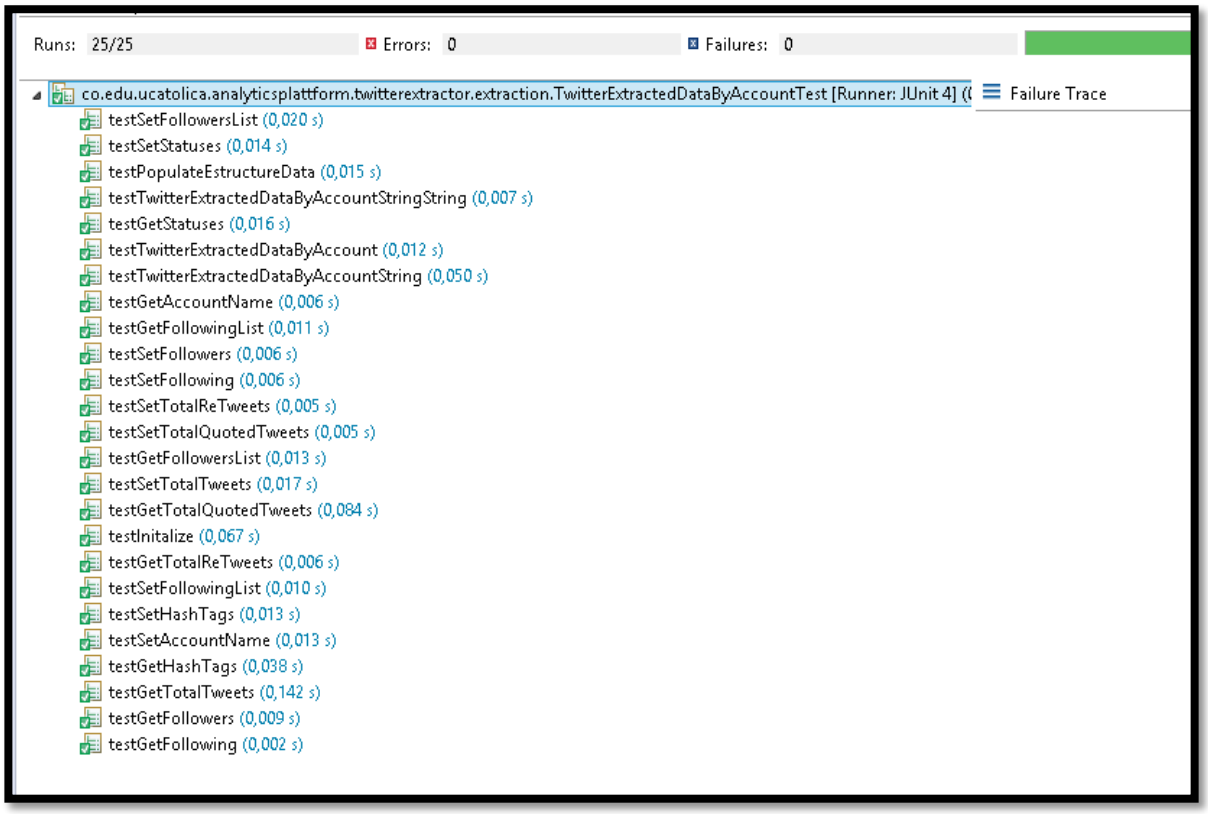
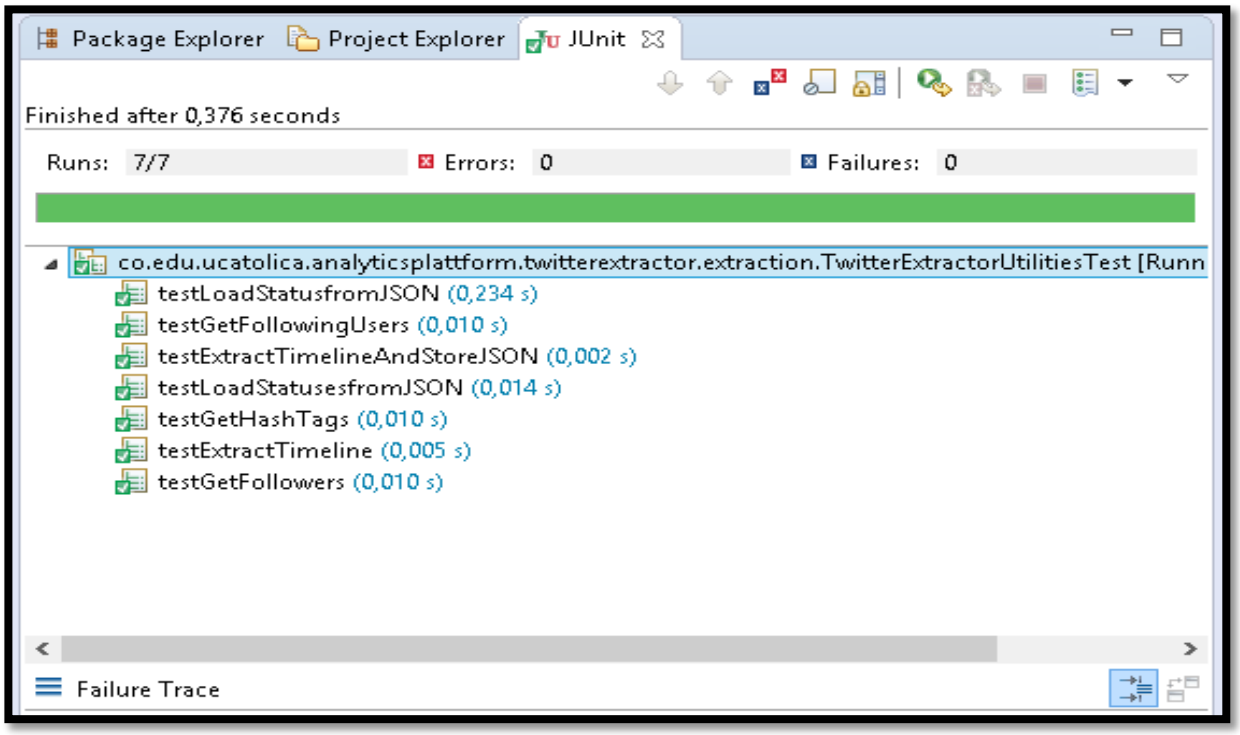



Ilustración 25 - Caso de prueba TwitterExtractorUtilities



De esta manera establecemos un testeo por cada uno de los componentes y al verificar el correcto funcionamiento de cada método perteneciente a ellos, garantizamos la calidad del software y su funcionalidad.

## 7.1 ANÁLISIS DE RESULTADOS

Adicionalmente a las pruebas realizadas para el componente de extracción, se establecieron pruebas de manera mecánica acomodando ciertos parámetros de información para su posterior extracción y verificación de almacenamiento, dando prioridad a la calidad de los datos extraídos. Un ejemplo de estas pruebas es la generación de un archivo de texto que contenga los tweets extraídos de cuentas específicas en notación JSON. Esta fue una prueba inicial y el resultado fue realmente positivo ya que permite identificar como se compone un tweet o una publicación en twitter y los diferentes conjuntos de información que contiene.

**Ejemplo de extracción de un tweet de @JuanManSantos y almacenamiento en un fichero .txt**

Ilustración 26 - Extracción y almacenamiento en archivo de texto

Ejemplo que al verlo desde una estructura JSON definida permite obtener más claridad de cada uno de sus componentes:

Ilustración 27 - Estructura de un tweet en formato JSON parte1

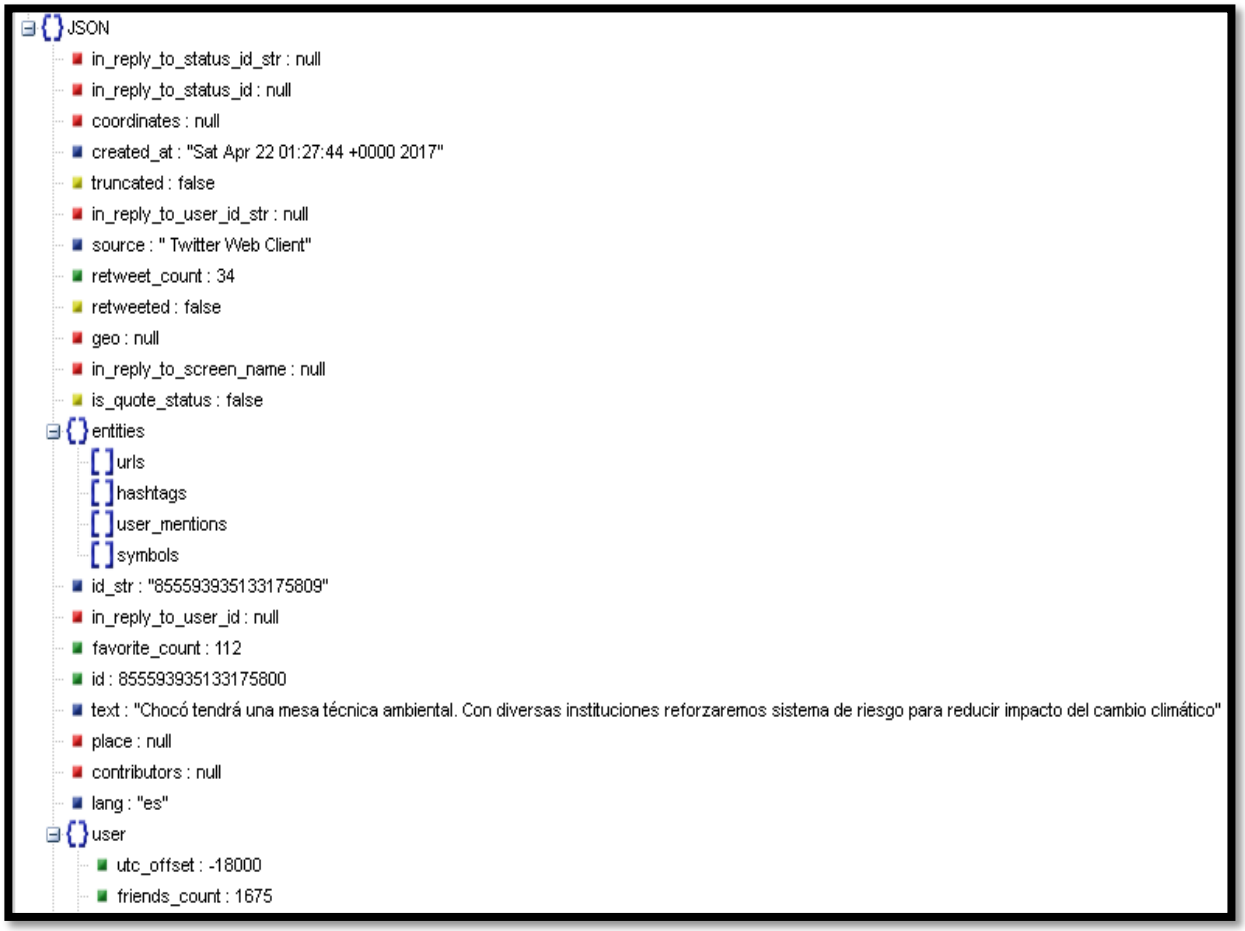


Ilustración 28 - Estructura de un tweet en formato JSON parte2

Ilustración 29 - Estructura de un tweet en formato JSON parte3


Estructura que es almacenada en una base de datos no relacional con colecciones definidas para cada tipo de dato o parametrización realizada en la codificación. En este caso como se definió en la sección de desarrollo, existe una clase para perfiles y otra para líneas de tiempo.

El almacenamiento de estas extracciones visualizado en la herramienta Robomongo se encuentra expuesto en las siguientes ilustraciones. Donde se presenta la colección general de perfiles extraídos y almacenados (Véase la ilustración 30 – Colección Profiles), acompañada de la colección creada para la cuenta JuanManSantos (cuenta del presidente actual del país) en la cual se almacenan los tweets extraídos (Ver las ilustraciones: Ilustración 31 - Colección JuanManSantos parte1 e Ilustración 32 - Colección JuanManSantos parte2)

Ilustración 30 - Colección Profiles.


Ilustración 31 - Colección JuanManSantos parte1

Ilustración 32- Colección JuanManSantos parte2


**(La herramienta definida construye una colección por cada cuenta y allí va insertando de forma cíclica cada Status de la cuenta, hasta llegar al límite por cuenta el cual es de 3200 tweets por extracción realizada).**

Al representar la información en esta estructura es mucho más claro notar la información que compone cada uno de los tweets que realiza un usuario en Twitter, en este caso representado en un tweet realizado por el Presidente de Colombia.

Allí es donde el límite de la imaginación se vuelve escaso al pensar los diferentes usos posibles para esta información. Muchas herramientas de extracción de datos de twitter (algunas expuestas en la sección de antecedentes) ofrecen servicios de análisis de información de twitter para potenciar las organizaciones o clientes que deseen hacer uso de sus herramientas. La mayoría de estas herramientas están enfocadas en el análisis del impacto de sus clientes en la parte social o comercial, el marketing, la influencia en el mercado y demás aspectos de impulso empresarial.

La herramienta construida para este proyecto es capaz de ofrecer una ventaja en cuanto a la información recolectada y el uso que se le pueda dar luego en una fase de análisis o desarrollo analítico externo. Esto debido a que independientemente del contexto político que tiene el proyecto en general, la herramienta ofrece y llegar a un nivel de especificación demasiado alto.

Un ejemplo claro es la extracción y almacenamiento de todos los metadatos de cada tweet, como su localización al momento de publicación, dispositivo desde el cual se realizó, y demás datos que abren la oportunidad de realizar análisis diferentes fuera de la parte de marketing y darle un contexto más científico, el cual pueda aportar a la sociedad con información que motive por ejemplo, el análisis predictivo en pro de la vida y bienestar global.

Cada resultado de extracción por parte de la herramienta es almacenado con un estándar de calidad y estructura definida, con el fin de proporcionar una fuente de datos organizada y que se ajuste a los requerimientos o necesidades del usuario.

El propósito planteado en cuanto a la especificación de objetivos, tanto el general como los específicos del proyecto ha sido cumplido implementando las herramientas necesarias para llevar a cabo la implementación de la mejor manera posible.

# CONCLUSIONES

Luego de la ejecución del proyecto y tomando como base los resultados obtenidos referentes a los procesos de extracción y almacenamiento de datos de la red social twitter, tanto en la fase de pruebas del proyecto, como en el resultado visual de los datos y teniendo en cuenta la calidad de estos, la cual es necesaria para poder establecer un almacenamiento sencillo y eficaz y así posteriormente con el apoyo de desarrollos o integraciones futuras, poder generar un análisis estadístico de varianza, con posibilidad de enfoque al a análisis de sentimientos y con un carácter predictivo. Se concluye de esta forma, que los objetivos planteados para el desarrollo del proyecto y enfocados en la descripción de la problemática a tratar fueron cumplidos, abarcando en sí cada aspecto planteado para el desarrollo de una forma adecuada que permitió con base al alcance y limitaciones del proyecto, brindar la solución esperada por los interesados tras este ciclo de implementación.

El desarrollo del proyecto permite definir a la extracción y almacenamiento de datos de redes sociales como una práctica o conjunto de procesos que implica la comprensión de una base teórica referente a arquitectura de software, programación y almacenamiento de datos, ya que sin esta base conceptual el entorno de desarrollo de una herramienta que permita establecer estas funcionalidades es invisible a los ojos del desarrollador.

Los tipos de datos recolectados en una extracción a un medio definido pueden variar, y esta estructura variable entre ellos implica el uso de diferentes técnicas de manejo de datos y sistemas de almacenamiento. El correcto tratamiento de estos datos puede definir en su totalidad los resultados obtenidos al culminar el proceso de desarrollo. En este caso modelos relacionales y no relacionales fueron integrados en el mismo sistema, con el fin de alcanzar el nivel de calidad y cumplir con los objetivos del proyecto, encontrando diferentes ventajas al definir correctamente la estructura general del desarrollo.

Definitivamente la extracción de datos de redes sociales, aplicando una metodología de desarrollo ágil como lo es el concepto de Personal Extreme Programming, seleccionado para la construcción de la herramienta y siguiendo un ciclo de investigación, planificación, diseño, desarrollo y pruebas, provee al implementador de una gran cantidad de información, que en una diferente fuente sería muy difícil de recuperar. El uso de herramientas como las APIs, en este caso generadas por comunidades de desarrolladores de las diferentes redes sociales, permite implementar de forma gratuita desarrollos de minería de datos, minería web e inclusive desarrollos que apuntan al concepto BigData en cuanto a la recolección masiva de datos para su posterior análisis y visualización. Lo que

permite que el concepto de datos abiertos se reproduzca y las soluciones a diferentes problemáticas se den de una manera mucho más sencilla.

La formación de un desarrollador frente a un proyecto con un concepto tan amplio, el cual implica el manejo de una gran variedad de herramientas y conocimientos de la ingeniería de sistemas y computación, crea una evolución en las habilidades de análisis y desarrollo de forma personal, y genera un tipo comprensión del flujo de información a escala global. Cambiando así el aspecto general del cómo se ven y relacionan las cosas en el entorno en el cual vivimos.

# RECOMENDACIONES

El presente proyecto de cierto modo fomenta e implica un proceso de desarrollo o un trabajo a futuro extenso, esto debido a que la herramienta desarrollada tiene la capacidad de generar un entorno al cual se le puede sacar provecho generando una fase específicamente enfocada en el análisis de los datos recolectados y también una fase posterior que permita una visualización de cada uno de los análisis planteados.

Se recomienda darle una continuidad al desarrollo integrado con otras herramientas, las cuales pueden desarrollarse enfocadas y adaptadas a la estructura general de este proyecto, por parte de los estudiantes de la universidad católica de Colombia, ya que sin duda alguna el proceso de desarrollo teniendo en cuenta el alcance y limitaciones del proyecto permite que se generen a través de su aplicación gran cantidad de propuestas con referencia a análisis descriptivos y por qué no, predictivos, visualizaciones e incluso mejoras de estructura general.

Por ahora la propuesta de la fase de análisis está siendo planteada para iniciar con el desarrollo en el próximo ciclo académico. Esta sería una fase que brinda a la herramienta funcionalidades que permitirán establecer soluciones a problemáticas diferentes, basadas en datos reales y concretos.

El uso de servicios web permite que este tipo de desarrollos se realicen de una forma ágil, por lo cual se recomienda la utilización de aquellos que se consideren adecuados y se adapten al contexto del proyecto a realizar.

Tecnologías como BigData e Inteligencia de negocios, deberían toman una mayor participación en cuanto a la construcción de soluciones con el fin de satisfacer necesidades durante el ámbito académico de una carrera como lo es la Ingeniería de sistemas. Esto implica un compromiso por parte de la entidad educativa y los docentes para poder aplicar los debidos conocimientos a los alumnos y así evolucionar de la misma forma que la tecnología lo hace.

# BIBLIOGRAFIA


(INTECO), I. N. de T. de la C. (2009). Estudio sobre la privacidad de los datos personales y la seguridad de la información en las redes sociales online. Retrieved from http://www.uv.es/limprot/boletin9/inteco.pdf

Agarwal, R., & Umphress, D. (2008). Extreme programming for a single person team. *Proceedings of the 46th Annual Southeast Regional Conference on XX - ACM-SE 46*, (March), 82. https://doi.org/10.1145/1593105.1593127

Bustamante, D., & Rodríguez, J. C. (2014). Metodología Actual Metodologia XP. Retrieved from http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf

Clases, D. De, Objetos, D. De, Estados, D. De, Secuencias, D. De, Actividades, D. De, Colaboraciones, D. De, & Componentes, D. De. (2001). Diagramas del UML, 1–23.

Eisenberg, T., Bigrigg, M. W., Kathleen, P., Kunkel, F., Chieffallo, D., & Diesner, J. (2010). AutoMap : *Office*.

Facchín, J. (2016). Las Redes Sociales más importantes del Mundo “Lista 2016.” Retrieved April 20, 2017, from http://josefacchin.com/2013/03/15/las-redes-sociales-mas-populares-del-planeta/

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

García García, P., & Rodríguez, C. A. (2017). Minería de Datos aplicada a las Redes Sociales. Retrieved from http://www.it.uc3m.es/jvillena/irc/practicas/08-09/08.pdf

Gartner Inc. (2007). “Dirty datos” es un problema de negocios, no un problema de TI, según Gartner. Retrieved April 3, 2017, from http://www.gartner.com/newsroom/id/501733

Gartner Inc. (2013). What Is Big Data? - Gartner IT Glossary - Big Data. Retrieved May 7, 2017, from http://www.gartner.com/it-glossary/big-data%5Cnhttp://www.gartner.com/it-glossary/big-data/

git. (2015). Una breve historia de Git. Retrieved May 23, 2017, from https://git-scm.com/book/es/v1/Empezando-Una-breve-historia-de-Git

Glez-Peña, D., Lourenzo, A., López-Fernández, H., Reboiro-Jato, M., & Fdez-Riverola, F. (2013). Web scraping technologies in an API world. *Briefings in Bioinformatics*, *15*. https://doi.org/10.1093/bib/bbt026

Hootsuite. (2017). Hootsuite (Home Page). Retrieved March 16, 2017, from https://hootsuite.com/es/

IA, D. C. de la C. e. (2014). Pruebasunitarias. Retrieved May 2, 2017, from http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.html

IBM Bluemix. (2015). Insights for Twitter. Retrieved May 28, 2017, from https://console.ng.bluemix.net/catalog/ibm-insights-for-twitter/

IETF. (1999). RFC 2616 - Hypertext Transfer Protocol -- HTTP_1. Retrieved May 15, 2017, from https://tools.ietf.org/html/rfc2616

Issi, G. (2003). Metodologías Ágiles en el Desarrollo de Software. Retrieved from

http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf

Liferay. (2017). Liferay Portal Feature Overview | Liferay. Retrieved April 4, 2017, from https://web.liferay.com/es/products/liferay-portal/features/portal

Logicalis. (2015). Redes sociales como fuentes de datos_ el caso de Twitter. Retrieved May 27, 2017, from https://www.marketingdirecto.com/digital-general/social-media-marketing/breve-historia-de-las-redes-sociales

Lotfy, A. E., Saleh, A. I., El-Ghareeb, H. A., & Ali, H. A. (2016). A middle layer solution to support ACID properties for NoSQL databases. *Journal of King Saud University - Computer and Information Sciences*, *28*, 133–145. https://doi.org/10.1016/j.jksuci.2015.05.003

Marketing Directo. (2011). Breve historia de las redes sociales - Marketing Directo. Retrieved May 27, 2017, from http://www.marketingdirecto.com/actualidad/social-media-marketing/breve-historia-de-las-redes-sociales/

Marset, R. N. (2007). REST vs Web Services. Retrieved from http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf

Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K., & Ishizuka, M. (2007). POLYPHONET: An advanced social network extraction system from the Web. *Web Semantics*. https://doi.org/10.1016/j.websem.2007.09.002

NTP-ISO/IEC 12207. (2006). isoiec12207[7]. Lima, Perú. Retrieved from http://www.senasa.gob.pe/senasa/wp-content/uploads/2014/11/Certificacion-citricos-a-mexico_26_mayo_2105_2.pdf

Olston, C., & Najork, M. (2010). Web Crawling. *Foundations and Trends R in Information Retrieval*, *4*(3), 175–246. https://doi.org/10.1561/1500000017

Rojas, D., & Platzi. (2016). ¿Cuál es la diferencia entre Big Data y Business Intelligence? Retrieved April 20, 2016, from https://platzi.com/blog/diferencia-big-data-business-intelligence/

Rubira, J. (2011). Twitter4j, integración de tu aplicación Java con Twitter. Retrieved March 8, 2017, from https://www.genbetadev.com/frameworks/twitter4j-integracion-de-tu-aplicacion-java-con-twitter

SAIMA Solutions. (2013). Business Intelligence y Big Data, ¿independencia o cooperación? | Blog de Saima Solutions. Retrieved from http://www.saimasolutions.com/blog/business-intelligence-big-data/

Schroeck, Michael; Shockley, Rebecca; Smart, J. (2012). Analytics: el uso de big data en el mundo real. *IBM. Informe Ejecutivo*, 22. https://doi.org/10.1007/978-1-84996-226-1

Srivastava, J., Desikan, P., & Kumar, V. (2006). Web Mining — Concepts, Applications, and Research Directions. Retrieved from http://dmr.cs.umn.edu/Papers/P2004_4.pdf

Techopedia. (2016). What is Data Retrieval? - Definition from Techopedia. Retrieved March 15, 2017, from https://www.techopedia.com/definition/26464/data-security

Vásquez Vélez, M. (2012). EL HABEAS DATA EN LAS REDES SOCIALES. Retrieved from http://bdigital.ces.edu.co:8080/repositorio/bitstream/10946/1281/2/Habeas data.pdf

Web, S. (2011). Guía Breve de Servicios Web. Retrieved March 16, 2017, from http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb

Weigend, A. (2014). Sabías que es un Data Scientist? Retrieved May 5, 2017, from http://sabiasqueestadistica.blogspot.com.co/2014/03/sabias-que-es-un-data-scientist_3.html

Wood, D. (2010). Linking enterprise data. *Linking Enterprise Data*, 1–291. https://doi.org/10.1007/978-1-4419-7665-9

Wood, R., Zheludev, I., & Treleaven, P. (2012). Mining Social Data with UCL's SocialSTORM Platform. London. Retrieved from http://weblidi.info.unlp.edu.ar/worldcomp2012-mirror/p2012/DMI9011.pdf

# ANEXOS

## ANEXO A

**Especificación de requisitos de software**

Herramienta web para la extracción y almacenamiento de datos de la red social twitter**.**

Marzo 2017

# Contenido

# 1 Introducción

Este documento es una Especificación de Requisitos Software (ERS) de la herramienta web para la extracción y almacenamientos de datos de la red social twitter. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

## 1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo de una herramienta web que permitirá extraer y almacenar datos de la red social twitter además de distintos procesos en cuanto a limpieza y parametrización de estos datos. Esta herramienta será utilizada por los usuarios interesados.

## 1.2. Alcance

Esta especificación de requisitos está dirigida al supervisor y usuario del sistema, para continuar con el desarrollo de herramientas web dinámicas y para profundizar en el desarrollo de ésta, la cual tiene por objetivo principal capturar y almacenar datos de la red social twitter, los cuales permitan en un futuro desarrollo, realizar una serie de análisis estadísticos con los diferentes tipos de datos obtenidos para un bien común.

## 1.3. Personal involucrado

| | |
|---|---|
| **Nombre** | José Camilo Barriga Mariño |
| **Rol** | Analista y programador |
| **Responsabilidad** | Análisis metodológico y desarrollo de la herramienta web. |
| **Información de contacto** | jcbarriga40@ucatolica.edu.co |

## 1.4. Definiciones, acrónimos y abreviaturas

| ***Nombre*** | ***Descripción*** |
|---|---|
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **Herramienta** | Herramienta web para la extracción y almacenamiento de datos de la red social twitter |
| **ERS** | Especificación de Requisitos Software |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |

## 1.5. Referencias

| **Título del Documento** | **Referencia** |
|---|---|

| | |
|---|---|
| Standard IEEE 830 - 1998 | IEEE |
| Formato IEEE 830 | FDI |

## 1.6. Resumen

Este documento tiene como fin, primero establecer una visión general de la herramienta que permita identificar los diferentes recursos del sistema, posteriormente se realiza una descripción general, la cual servirá como un factor de reconocimiento en cuanto a los principales puntos a los que debe ir enfocado el desarrollo, culminando así con una definición detallada de los requisitos identificados, los cuales debe satisfacer el sistema y definen el comportamiento funcional del mismo.

# 2. Descripción general

## 2.1. Perspectiva del producto

La herramienta será un producto diseñado para funcionar en un entorno web, permitirá su utilización de forma rápida, eficaz y con un fácil acceso. Se integrara conjuntamente con el portal gestor de contenidos Liferay.

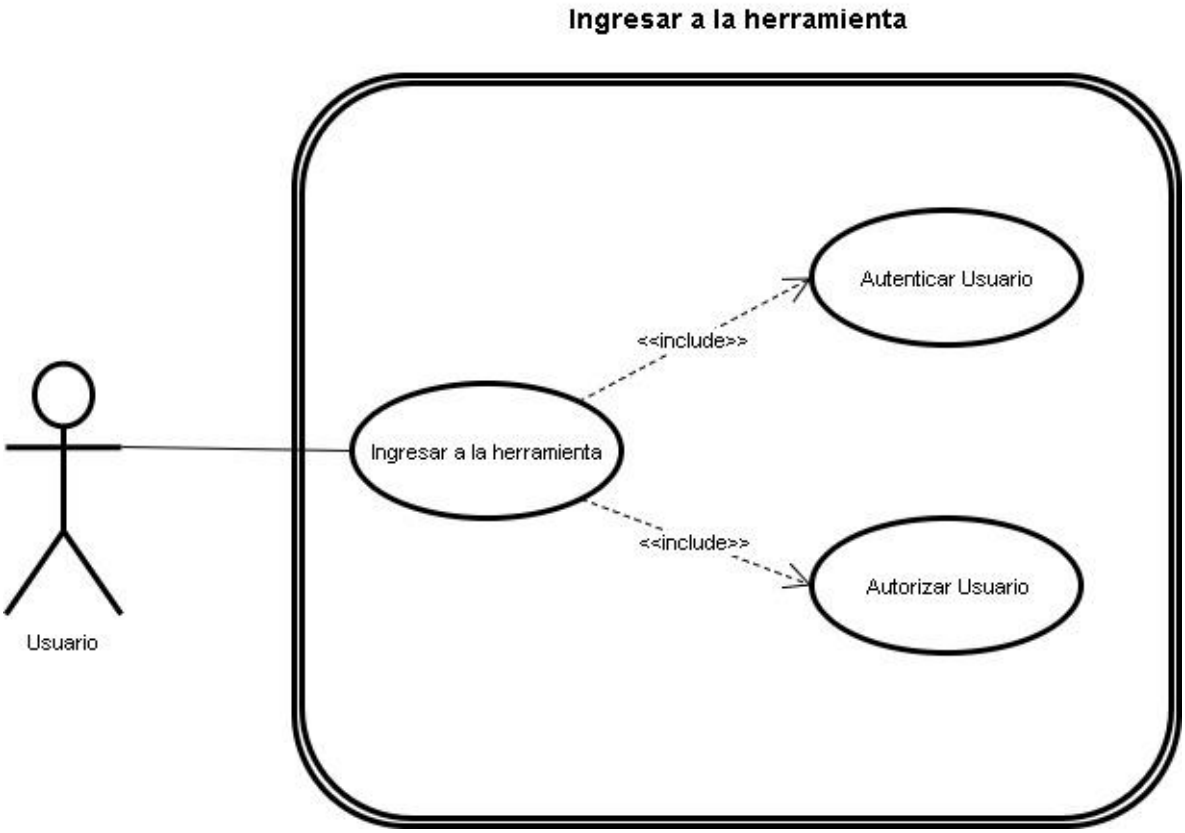
## 2.2. Metáfora del sistema

La herramienta elaborada consiste en un software web el cual permite extraer grandes cantidades de información de cuentas públicas de la red social twitter, información que luego organiza y posteriormente las almacena con el fin de generar una utilidad en caso de querer generarse un análisis o reporte estadístico.

## 2.3. Funcionalidad del producto

### Diagramas de Casos de Uso

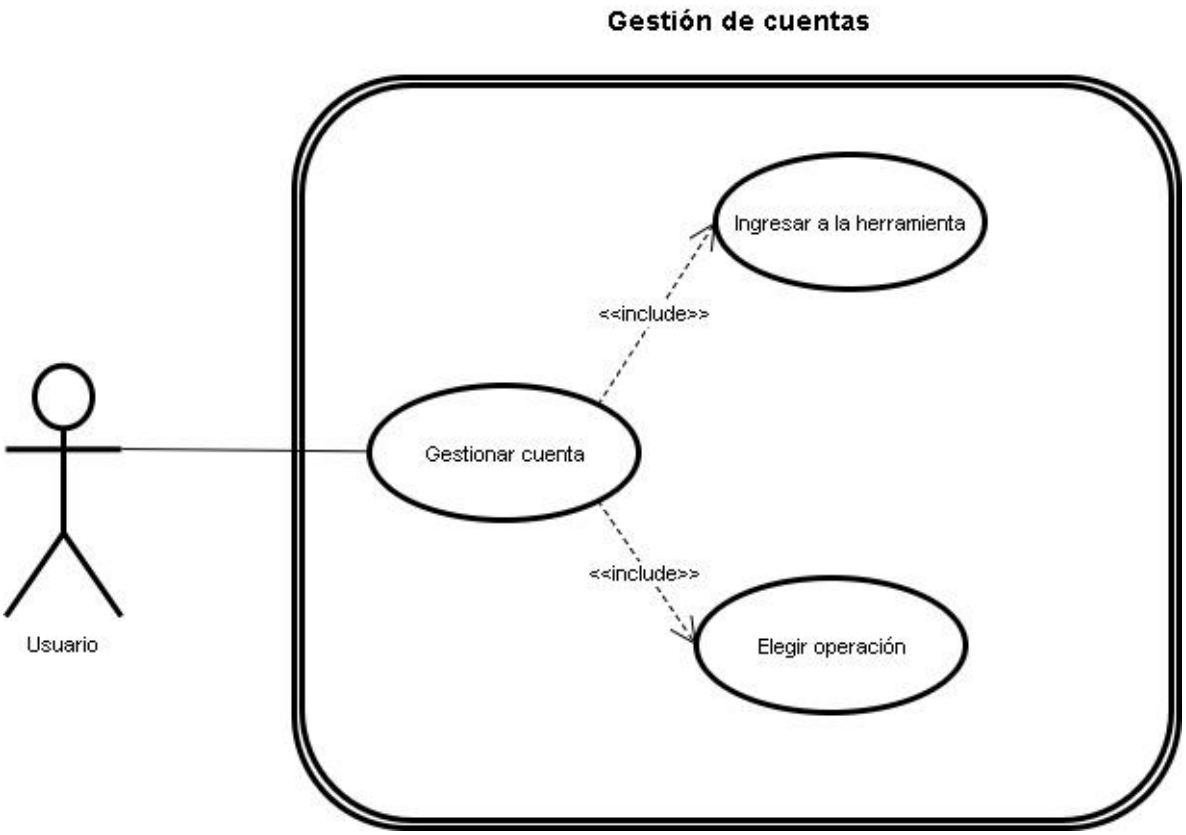
**Diagrama de casos de uso – Ingreso a la herramienta**




El usuario ingresará a la herramienta a través de una interfaz que deberá validar su identidad dependiendo de la autenticidad de los datos de acceso ingresados. Posteriormente el sistema al realizar una consulta en la base de datos debe permitir o no el acceso del usuario a la herramienta.

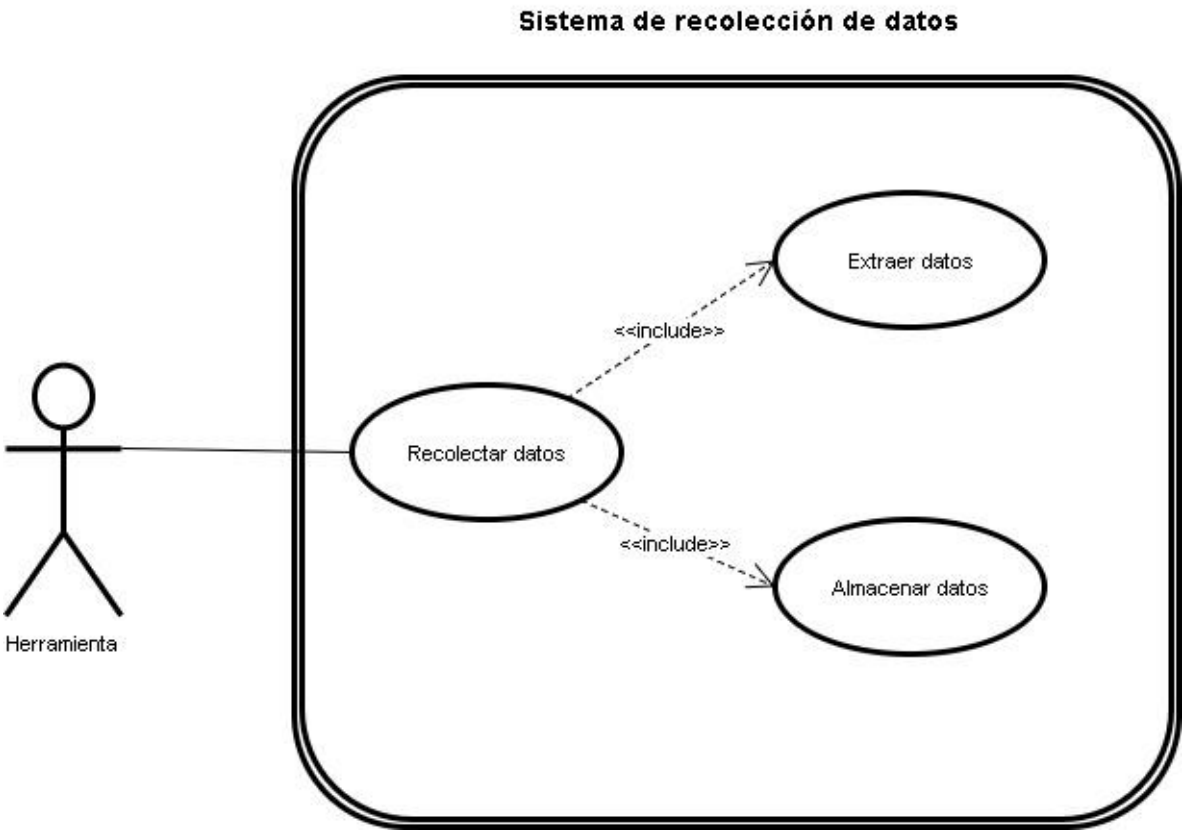
Por lo tanto debe existir un control y nivel de seguridad adecuados, que implican un almacenamiento de con la información del usuario que sea la base para realizar la verificación y posteriormente dar la autorización.

**Diagrama de casos de uso – Gestión de cuentas**




El usuario que desee extraer y almacenar los datos de una cuenta específica, deberá en primer lugar acceder a la herramienta e insertar la cuenta (identificador) a extraer. Este dato será la base para realizar las operaciones correspondientes con el fin de extraer la información necesaria de la cuenta, realizar curación de los datos y generar datos relevantes de la misma para posteriormente ingresarla en las bases de datos del sistema. De la misma manera el usuario puede gestionar cada una de las cuentas y clasificarlas en categorías y subcategorías, las cuales puede operar (crear, leer, actualizar y eliminar).

**Diagrama de casos de uso de Funcionalidad Generar – Recolección de información**

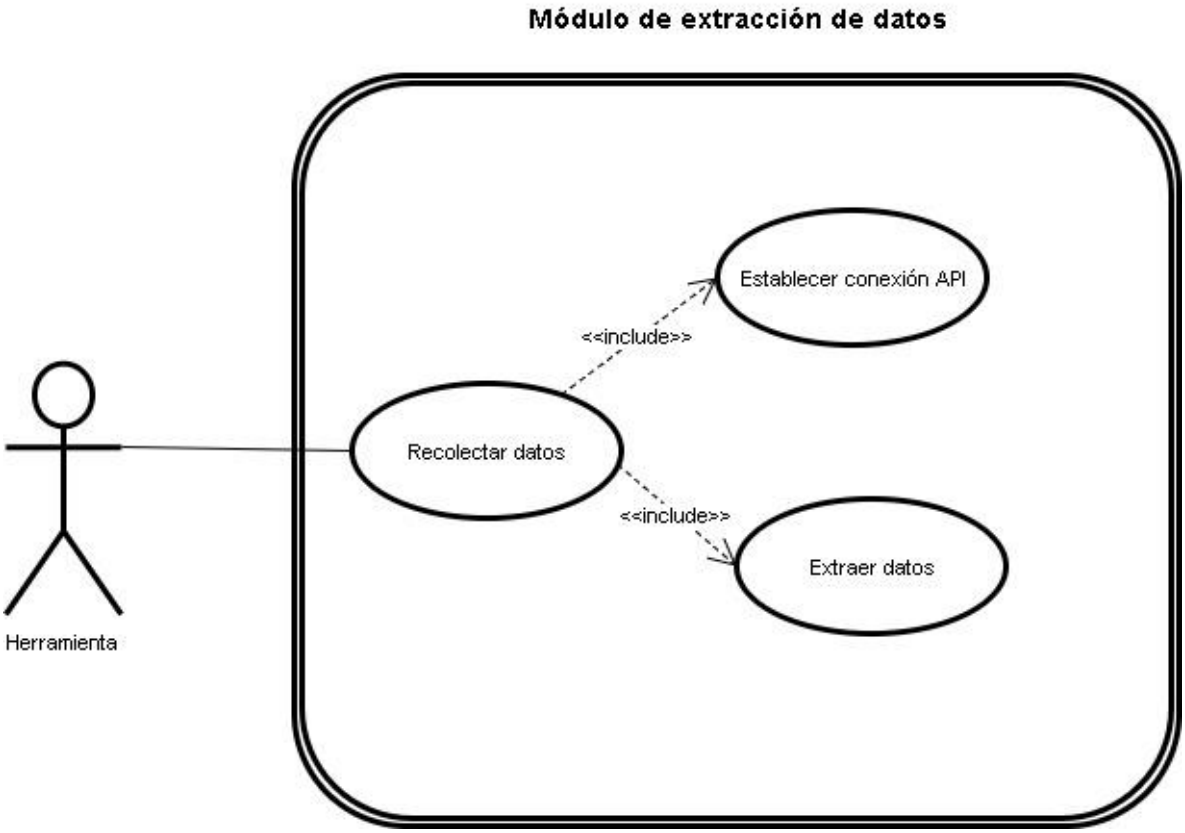



Representa el componente de extracción y almacenamiento de la herramienta, adicionalmente proporciona el contexto general de la funcionalidad de la herramienta.

**Este diagrama se compone de dos procesos vistos como requisitos principales, los cuales por sus componentes específicos se exponen a continuación de una forma más formal.**

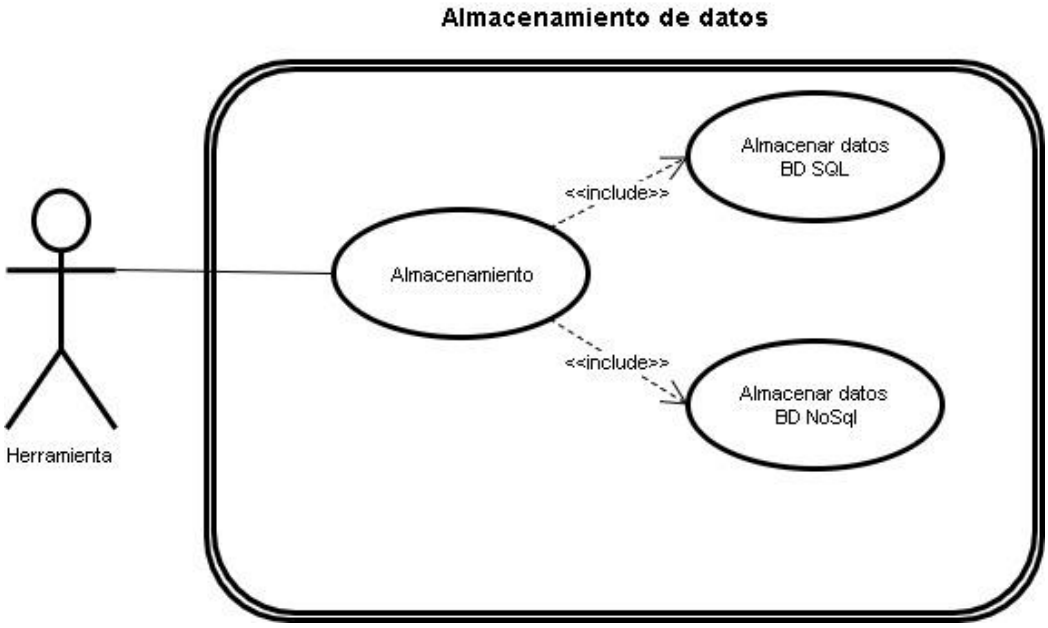
***Extracción de datos***

Modela el componente de extracción donde para poder realizar la recolección de los datos por cada cuenta, es necesario realizar la conexión con el API de la red social Twitter además de la correspondiente captura y parametrización de estos datos a través de la programación.




***Almacenamiento de datos***

Modela el proceso de almacenamiento de la herramienta, incluyendo los datos recibidos de la red social como los datos generados por la herramienta en su modelo parametrizable. Esto implica su almacenamiento de forma relacional y no relacional en diferentes gestores de bases de datos, donde los datos a almacenar en la base de datos no relacional deben primero pasar por un proceso en el cual se les aplica la notación necesaria (JSON) para posteriormente poder tratarlos dependiendo de la necesidad del usuario.




**Diagramas BMPN**

**Diagrama BPMN- Extracción y Almacenamiento de datos en base de datos No relacional.**

Este diagrama es generado para aclarar el proceso de almacenamiento en la base de datos no relacional, proceso de recolección empieza con la conexión establecida con la API de twitter la cual proveerá el acceso a los datos de las cuentas requeridas. Posteriormente se realiza la captura de los datos, a los cuales se les aplica la notación adecuada para su posterior operación (JSON) y se envían de forma organizada a la base de datos NoSQL.

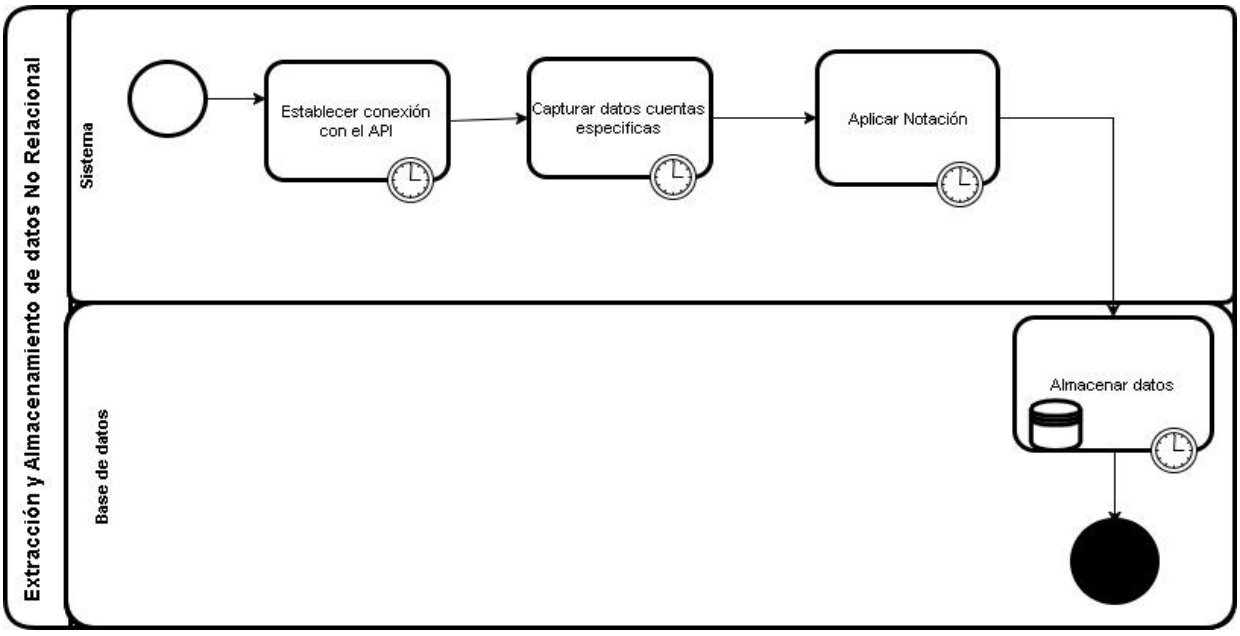
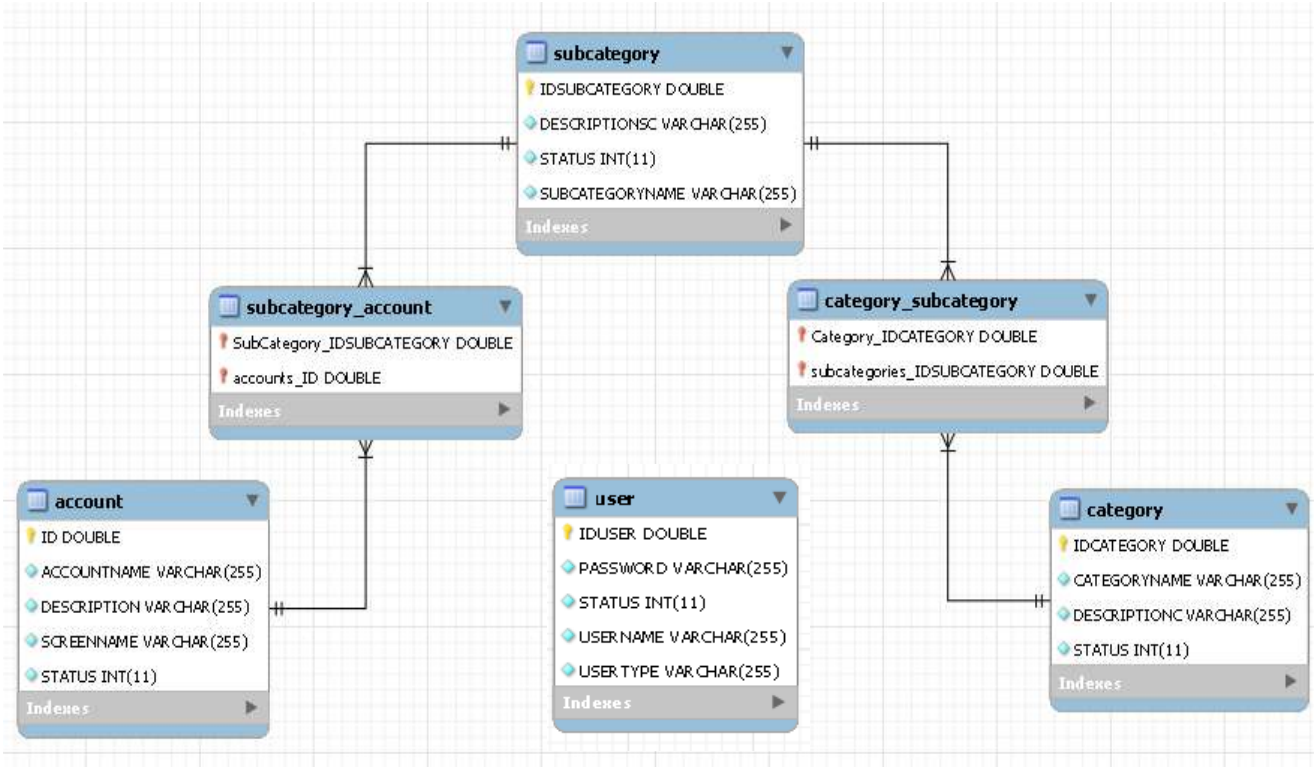


**Diagrama Entidad/Relación**




El diagrama Entidad relación podemos visualizar el orden, atributos y diferentes relaciones que posee cada entidad de nuestro sistema relacional, el cual está enfocado en el modelo de parametrización.

Específicamente en este diagrama podemos verlas entidades de cuenta, subcategoría y categoría.

Donde el objetivo es establecer una clasificación de cada cuenta, por lo tanto este fin es representado con la cardinalidad de cada relación.

Dos tablas intermedian representan los atributos que establecen la relación entre las entidades. En el caso de **account** y **subcategory** podemos identificar a través de la tabla de su relación "**subcategory_account**" que por cada subcategoría existirá una lista de cuentas pertenecientes a la misma. Así mismo, en el caso de la entidad **category** y la entidad **subcategory**, donde en la tabla de su relación "**category_subcategory**" podemos apreciar que por cada categoría existirá una lista o un grupo de subcategorías definidas.

Orden en el cual se establece una jerarquía y permite identificar la forma en que los datos que ingresaran al modelo se relacionarán entre ellos.

Adicionalmente se define una entidad **user,** la cual así no se relacione con las demás entidades, es importante ya que se trata de la información del usuario que manipulara los datos de todo el modelo.

## 2.4. Características de los usuarios

| | |
|---|---|
| **Tipo de usuario** | Administrador General |
| **Roles** | NA |
| **Actividades** | Tiene permisos en todos los módulos, en cuanto a la parametrización, clasificación de cuentas, extracción y almacenamiento de datos. |

| | |
|---|---|
| **Tipo de usuario** | Usuario general |
| **Roles** | Coordinador, Investigador |
| **Actividades** | Indagar información de cuentas extraídas y clasificación de cuentas según parametrización del modelo. |

## 2.5. Restricciones

- Interfaz para ser usada únicamente en la web.
- Lenguajes y tecnologías en uso: JAVA, JPA (gestión de la base de datos), MySQL, Liferay como gestor de contenidos.
- Se consume un servicio a través de un API que utiliza arquitectura REST.
- El modulo web utiliza el concepto de portlets.

## 2.6. Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben ciertos requisitos indicados en la sección anterior.

# 3. Requerimientos específicos

## 3.1. Requerimientos Funcionales

| | |
|---|---|
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Ingresar a la herramienta. |
| **Descripción del requerimiento:** | En este proceso de ingreso a través de una interfaz definida, se pedirá una autenticación obligatoria y autorización a los usuarios de la herramienta web para su ingreso. |
| **Acción** | El sistema debe imponer el ingreso al aplicativo por medio de la correcta comprobación de datos definidos, como usuario y contraseña. Esto aplicará a todo tipo de usuarios. |
| **Requerimiento NO funcional implicado:** | • RNF01<br>• RNF02<br>• RNF03 |

| | |
|---|---|
| | • RNF04<br>• RNF05 |
| **Pre-Condiciones** | Acceder a la interfaz de autenticación |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:**<br>Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Gestionar cuentas |
| **Descripción del requerimiento:** | El usuario podrá a través de la plataforma añadir cuentas, eliminarlas, listarlas y clasificarlas. Estas serán las cuentas de twitter a las cuales se realizará realizar el proceso de extracción y almacenamiento de datos. |
| **Acción** | Permitir al usuario en cualquier momento Ingresar a la interfaz de la herramienta y gestionar las cuentas que necesite. |
| **Requerimiento NO funcional:** | • RNF01<br>• RNF02<br>• RNF03<br>• RNF04<br>• RNF05 |
| **Pre-Condiciones** | RF01 |
| **Post-Condiciones** | RF03, RF04 |
| **Prioridad del requerimiento:**<br>Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Extraer Datos. |
| **Descripción del requerimiento:** | La herramienta debe ser capaz de extraer datos de la red social Twitter de las cuentas que el usuario seleccione. |
| **Acción** | Extraer datos de las diferentes cuentas públicas de la red social seleccionadas por el usuario para su posterior uso, estableciendo las conexiones y la parametrizaciones necesarias. |
| **Requerimiento NO funcional:** | • RNF02<br>• RNF04<br>• RNF05 |
| **Pre-Condiciones** | RF01, RF02 |

| | |
|---|---|
| **Post-Condiciones** | RF04 |
| **Prioridad del requerimiento:**<br>Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RF05 |
| **Nombre del Requerimiento:** | Almacenamiento. |
| **Descripción del requerimiento:** | Almacenar datos generados y extraídos por la herramienta. |
| **Acción** | Almacenar los datos extraídos e ingresados tanto de la red social como de la información de autenticación y la clasificación del modelo de parametrización con un formato adecuado que promueva la calidad de esta información. |
| **Requerimiento NO funcional:** | • RNF02<br>• RNF04<br>• RNF05 |
| **Pre-Condiciones** | RF01, RF02, RF03 |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:**<br>Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RF06 |
| **Nombre del Requerimiento:** | Clasificar cuentas |
| **Descripción del requerimiento:** | El usuario podrá clasificar las cuentas en subcategorías y categorías. |
| **Acción** | Permitir al usuario generar una clasificación de cuentas creando diferentes subcategorías y categorías a las cuales pertenecen. |
| **Requerimiento NO funcional:** | • RNF01<br>• RNF05 |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:**<br>Alta | |

## 3.2. Requerimientos No Funcionales

| | |
|---|---|
| **Identificación del** | RNF01 |

| | |
|---|---|
| **requerimiento:** | |
| **Nombre del Requerimiento:** | Interfaz del sistema. |
| **Descripción del requerimiento:** | La herramienta debe tener una interfaz de uso intuitivo y debe ser sencilla. |
| **Acción** | Presentar una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema. |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Interacción con el usuario |
| **Prioridad del requerimiento:** Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RNF02 |
| **Nombre del Requerimiento:** | Desempeño |
| **Descripción del requerimiento:** | La herramienta debe ser ágil y eficaz en su respuesta, en base a la extracción, almacenamiento y consulta de datos. |
| **Acción** | Garantizar la interacción con la herramienta de una forma óptima en cuanto a las diferentes operaciones con los datos implicados. |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:** Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RNF03 |
| **Nombre del Requerimiento:** | Facilidad de uso |
| **Descripción del requerimiento:** | Debe ser un aplicativo Intuitivo, Sencillo pero eficiente |
| **Acción** | Garantizar fácil acceso a sus módulos y debe ser entendible para el usuario. |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:** | |

| |
|---|
| Alta |

| | |
|---|---|
| **Identificación del requerimiento:** | RNF04 |
| **Nombre del Requerimiento:** | Disponibilidad del sistema. |
| **Descripción del requerimiento:** | La herramienta tendrá que estar disponible para su uso, las ocasiones y en el tiempo que el usuario lo necesite. |
| **Acción** | Garantizar una disponibilidad de la herramienta ligada a la necesidad del usuario. |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:** Alta | |

| | |
|---|---|
| **Identificación del requerimiento:** | RNF06 |
| **Nombre del Requerimiento:** | Seguridad en información |
| **Descripción del requerimiento:** | El sistema garantizara una seguridad en cuanto a la información que se procede en el sistema. Por ejemplo datos de autenticación. |
| **Acción** | Garantizar la integridad de los datos manipulados por la herramienta, estos pueden ser datos internos generados por el usuario o el sistema o también se hace referencia a la seguridad de los datos extraídos de la red social, los cuales tienen un contenido único que no debe ser modificado con otro fin que el de generar estadísticas o añadir parámetros para sacar conclusiones en base a estadísticas o reportes de conteos. |
| **Pre-Condiciones** | Ninguna |
| **Post-Condiciones** | Ninguna |
| **Prioridad del requerimiento:** Alta | |

## 3.3. Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y/o campos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

### 3.3.1. Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Procesador de 1.66GHz o superior.
- Memoria mínima de 256Mb.
- Mouse.
- Teclado.
- Acceso a internet.

### 3.3.2. Interfaces de software

- Sistema Operativo: Windows XP o superior, Linux.
- Explorador: Mozilla o Chrome (Recomendación).

### 3.3.3. Interfaces de comunicación

Los servidores, interfaces, clientes y la aplicación se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible.

## 3.4. Historias de usuario

### 3.4.1. Historia de usuario 1

**Nombre de la historia:** Ingresar a la herramienta**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera poder ingresar a la herramienta con usuario y contraseña, que el sistema valide y verifique que los datos ingresados en la interfaz de autenticación sean correctos.

**Observaciones del desarrollador:** En este proceso de ingreso a través de una interfaz definida, se pedirá una autenticación obligatoria y autorización a los usuarios de la herramienta web para su ingreso.

### 3.4.2. Historia de usuario 2

**Nombre de la historia:** Gestionar cuentas**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera poder Gestionar las cuentas a las cuales, se les realizará la extracción y el almacenamiento definidos, siguiendo el modelo y diseño de almacenamiento.

**Observaciones del desarrollador:** Este modelo comprende una clasificación a tener en cuenta, la cual divide las cuentas en categorías y subcategorías, generando así un orden y control diferente y más específico de los datos. Va de la mano con esta clasificación de cada cuenta establecida en el modelo de parametrización, ya que en él se encuentra la información del cuentas registradas oficialmente en el sistema así en la base de datos exista información de otras cuentas no registradas.

### 3.4.3. Historia de usuario 3

**Nombre de la historia:** Recolectar información de cuentas de twitter**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera poder recolectar información de las cuentas públicas de twitter que necesite tanto de su perfil como su actividad y almacenarla.

**Observaciones del desarrollador:** Va de la mano con el requisito de extracción el cual puede implicar tareas de conexión (con la red social) y algún tipo de parametrización u orden estructural adicional que deba realizarse a los datos capturados para su posterior operación.

También incluye al requisito de almacenamiento donde la herramienta debe almacenar los datos internos y los datos extraídos de la red social en los diferentes formatos necesarios para su operación. Esto implica el uso adecuado de gestores de bases de datos definiendo el tipo de datos y su tipo de relación para promover del funcionamiento correcto de la herramienta y aplicando los conceptos de procesamiento y tratamiento de datos masivos.

El almacenamiento implicara técnicas de programación para establecer una correlación entre una base de datos relacional y un sistema orientado a objetos.

### 3.4.4. Historia de usuario 4

**Nombre de la historia:** Clasificar cuentas**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera poder clasificar las cuentas a las cuales se les extraerá la información en categorías y subcategorías políticas.

**Observaciones del desarrollador:** Debe crearse la funcionalidad correspondiente que involucre la interfaz gráfica para que el usuario tenga la oportunidad de realizar la clasificación de las diferentes cuentas.

### 3.4.5. Historia de usuario 5

**Nombre de la historia:** Interfaz**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera poder manejar la herramienta a través de una interfaz web.

**Observaciones del desarrollador:** Interfaz accesible e intuitiva, básica pero consistente con la funcionalidad del aplicativo que permita su uso desde diferentes navegadores. Teniendo en cuenta que la mayor parte de funcionalidad (extracción y almacenamiento desde la red social) no necesitarán mayor participación de esta interfaz.

### 3.4.6. Historia de usuario 6

**Nombre de la historia:** Desempeño óptimo.

**Prioridad:** Media**.**

**Descripción:** Como usuario quisiera que la herramienta tuviera un tiempo de respuesta bajo sin afectar las funcionalidades.

**Observaciones del desarrollador:** Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red. El desempeño debe ser proporcional al alcance y limitaciones de los recursos donde se lleve a cabo el aspecto operacional de cada funcionalidad.

### 3.4.7. Historia de usuario 7

**Nombre de la historia:** Fácil uso de la herramienta**.**

**Prioridad:** Media**.**

**Descripción:** Como usuario quisiera que la herramienta fuera sencilla de operar al igual que su interfaz.

**Observaciones del desarrollador:** La herramienta debe tener un diseño e interfaz intuitivo y sencillo, este diseño debe ajustarse a las características requeridas (según funcionalidad), dentro de las cuales se encuentran el sistema de ingreso de datos y visualización de respuesta.

### 3.4.8. Historia de usuario 8

**Nombre de la historia:** Disponibilidad a necesidad**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quisiera que la herramienta estuviera disponible para su uso siempre que lo necesite.

**Observaciones del desarrollador:** La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios dependiente de la necesidad del usuario y el tiempo de uso que le dé, garantizando un esquema adecuado que permita satisfacer las necesidades de una forma segura y la facilidad de acceder cuando lo requieran los usuarios.

### 3.4.9. Historia de usuario 9

**Nombre de la historia:** Seguridad de la información**.**

**Prioridad:** Alta**.**

**Descripción:** Como usuario quiero que la herramienta tenga una seguridad confiable tanto en el acceso como en el manejo de datos.

**Observaciones del desarrollador:** Garantizar la integridad de los datos manipulados por la herramienta, estos pueden ser datos internos generados por el usuario o el sistema o también se hace referencia a la seguridad de los datos extraídos de la red social, los cuales tienen un contenido único que no debe ser modificado con otro fin que el de generar estadísticas o añadir parámetros para sacar conclusiones en base a estadísticas o reportes de conteos.

## GLOSARIO

**ANÁLISIS DE VARIANZA**: es una colección de modelos estadísticos y sus procedimientos asociados, en el cual la varianza está particionada en ciertos componentes debidos a diferentes variables explicativas., 19

**API**: Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción., 8, 20, 32, 47, 48, 49, 50, 54, 71, 83, 93, 95, 97

**CRUD**: Crear, Leer, Actualizar y Borrar", que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software., 55, 56

**CURACIÓN DE DATOS**: Gestión de datos a través de su ciclo de vida, incluyendo su recuperación y mantenimiento y facilitando su reutilización, 11

**DAO**: Objeto de acceso a datos es componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo., 54, 55, 57, 58, 60, 61, 62, 65, 66, 68, 70

**DISEÑO FACTORIAL**: es un experimento cuyo diseño consta de dos o más factores, cada uno de los cuales con distintos valores o niveles, cuyas unidades experimentales cubren todas las posibles combinaciones de esos niveles en todo los factores., 19

**IDE**: Entorno de desarrollo integrado., 49, 54, 64

**JSON**: JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos., 8, 31, 52, 73, 74, 75, 76, 94, 95

**LIBRERÍA**: En la programación se define como un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca., 49, 50, 64

**MINDFULNESS**: Prestar atención, momento a momento, a pensamientos, emociones, sensaciones corporales y al ambiente circundante, aceptándolos, es decir, sin juzgar si son correctos o no., 12

**PARAMETRIZACIÓN**: organización y estandarización de la información que se ingresa en un sistema, 58

**PLN**: El procesamiento del lenguaje natural (PLN) es un campo que combina las tecnologías de la ciencia computacional (como la inteligencia artificial, el aprendizaje automático o la inferencia estadística) con la lingüística aplicada, con el objetivo de hacer posible la comprensión y el procesamiento asistidos por ordenador de información expresada en lenguaje humano para determinadas tareas, 36

**PRUEBAS UNITARIAS**: Casos de prueba cada función no trivial o método en el módulo, de forma que cada caso sea independiente del resto., 11

**PXP**: Personal Extreme Programming (ágil metodología de desarrollo de software), 11

**REPOSITORIO REMOTO**: son versiones de un proyecto que se encuentran alojados en Internet o en algún punto de la red., 29

**SDLC**: Ciclo de vida del desarrollo de software, 28

**TIMELINE**: es la página principal de Twitter en la cual aparecen los mensajes de todos los usuarios de Twitter a los cual el usuario sigue., 51, 52, 53, 54, 57, 58