UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

CENTRO DE IDEOINFORMATICA

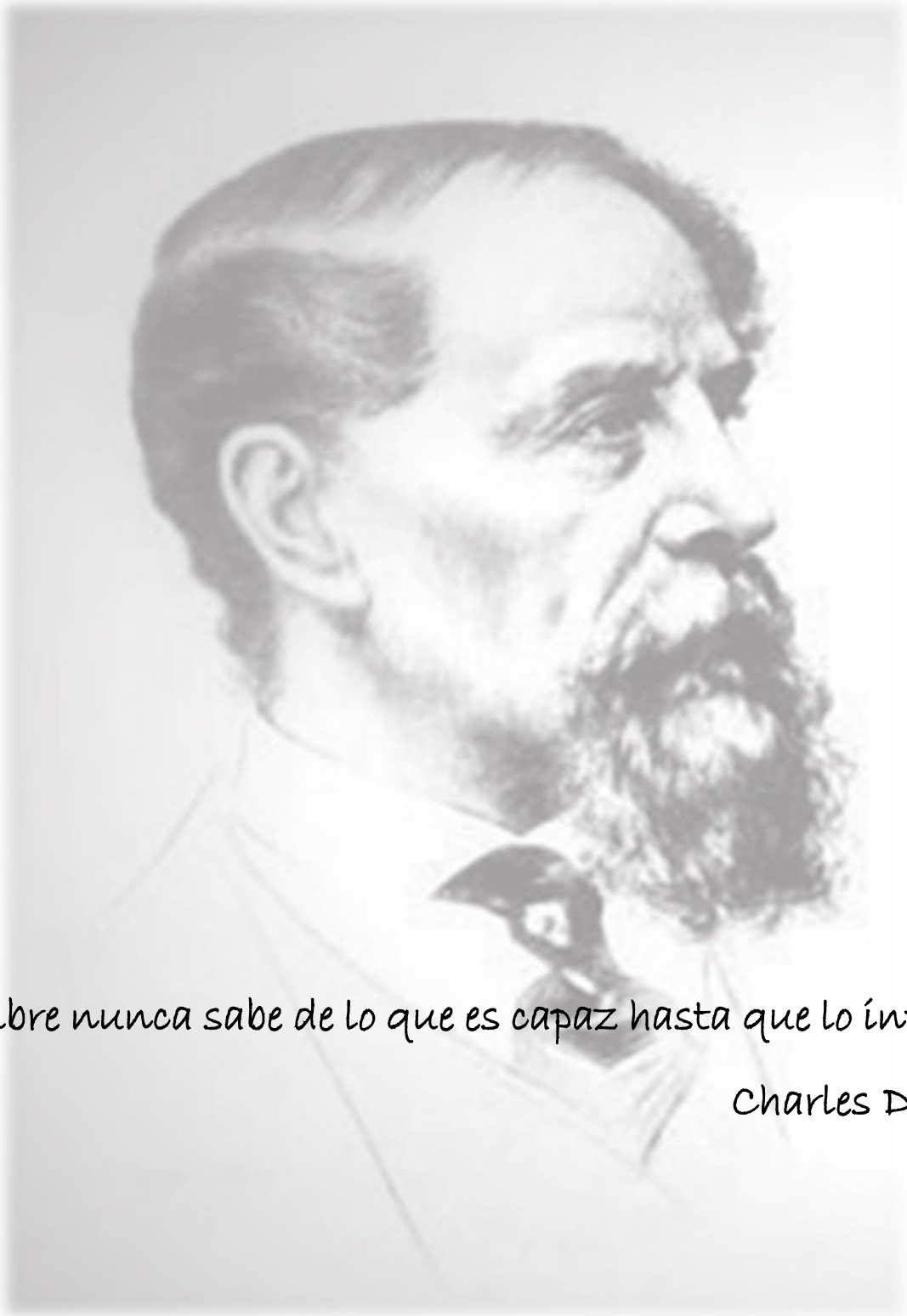
TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Amsor: Aplicación informática para la evaluación de impactos de los Movimientos Sociales en Twitter

Autor: Adrian Carballo Guilarte

Tutores: Msc. José Gabriel Espinosa Ramírez
Msc. Waldo Barrera Martínez

La Habana
2019



"El hombre nunca sabe de lo que es capaz hasta que lo intenta".

Charles Dickens

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que tiene por título: Amsor: Aplicación informática para la evaluación de impactos de los Movimientos Sociales en Twitter y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor:

_____

Adrian Carballo Guilarte

Tutores:

_____ _____

Msc. José Gabriel Espinosa Ramírez Msc. Waldo Barrera Martínez

Dedicatoria

Resumen

La presente investigación tiene como objetivo desarrollar una aplicación para la evaluación del impacto de los movimientos sociales en Internet, específicamente desde el sitio de redes sociales Twitter. Para esto se hizo uso del análisis de redes sociales, del tipo global de la red, y de la centralidad de grado para las relaciones de retuit y favoritos. El desarrollo de la solución estuvo guiado por la metodología AUP-UCI, en su cuarto escenario. Para la implementación de la aplicación se hizo uso de los lenguajes de programación PHP y Java, y se seleccionó como mecanismo de integración al sistema de colas de mensajería RabbitMQ. Para el procesamiento asíncrono de los tuits se hizo uso del patrón llamada a procedimiento remoto. Como marco de trabajo para la implementación de aplicaciones Web en PHP se utilizó Symfony. La estrategia de prueba aplicada permitió verificar el cumplimiento de los objetivos trazados y evaluar la calidad del sistema. Para la validación del resultado de la investigación se realizaron entrevistas a expertos.

Palabras claves: análisis de redes sociales, centralidad de grado, tuit, impacto, movimientos sociales.

Índice

Introducción.

Como parte del desarrollo de Internet, la aparición de los primeros sitios de redes sociales, se enmarcan a partir de 1995, con el nacimiento de *theglobe.com.* Este sitio le ofrecía a su comunidad de usuarios, registrados en todo el mundo, la posibilidad de personalizar su experiencia *online*, publicar contenidos propios e interactuar con otros usuarios de intereses similares. Durante el año 1995, cuando Internet contaba ya un millón de páginas web, también en EE.UU., surge *classmates.com*, permitiendo encontrar a antiguos compañeros de guardería, colegio, instituto, universidad e incluso del ejército estadounidense (Barrera, 2017).

Particular importancia en esta etapa inicial reviste también *SixDegrees.com*, proto-red nacida en 1997, considerada por algunos estudiosos como la primera en reflejar mejor la definición de RSI (Ros, 2009). Entre sus principales características se encontraban la posibilidad de crear perfiles, listas de amigos, envío de mensajes, y a partir de 1998, la facilidad de navegar en las listas de amigos de terceros. Fiel al nombre y a la connotación de la teoría de los seis grados de separación, se promovió como herramienta útil para ayudar a la gente a conectarse.

Con el paso de unos pocos años y el perfeccionamiento de las tecnologías de la denominada Web 2.0, pero principalmente durante la primera década del actual siglo, se produce una verdadera explosión en la aparición de plataformas sociales en todo el mundo. Friendster (2002), MySpace (2003), Facebook (2004), Youtube (2005), Twitter (2006) e Instagram (2010), se encuentran entre las más notables.

El impacto meteorítico de las RSI en prácticamente todos los ámbitos de la sociedad, se manifiesta también con mucha fuerza en el terreno de las relaciones internacionales. Los nuevos movimientos sociales, surgidos durante la segunda mitad del siglo XX, a partir del ciclo de protestas iniciado en EE.UU., con nuevos actores, formas de organización y rasgos distintivos que los diferencian de los movimientos tradicionales, como los obrero-campesinos, nacionalistas o comunistas (Perera, 2015), han ampliado de manera extraordinaria sus capacidades a partir de la incorporación de las plataformas sociales a los repertorios de actuación.

En las conclusiones de un estudio sobre los movimientos sociales publicado hace 30 años se vislumbran los indicios de lo que hoy acontece, y se asume que los movimientos sociales han ampliado sus capacidades

de movilización (Ibarra, 2000). Como hemos podido comprobar a través de los años, en consonancia con el desarrollo tecnológico de la Internet, este potencial se ha incrementado considerablemente y es evidente que ha trascendido los alcances que podían suponerse.

En la actualidad con la irrupción de la Web 2.0 y las herramientas digitales asociadas a ella, el fácil acceso, la inmediatez y universalidad del alcance, han permitido la ruptura del tradicional modelo de comunicación basado en la jerarquía y unidireccionalidad. También, la creación de espacios de articulación civil, donde consolidan movilizaciones instantáneas de ciudadanos en torno a determinados asuntos, convirtiendo a Internet en la más poderosa herramienta de transmisión de ideas de todos los tiempos (Barrera, 2017).

Entre las ventajas que ofrece Internet a los movimientos sociales en la red, en lo adelante MSR, están: la creación de vida social, el incremento de los canales de información y el volumen de esta; mayor cantidad de datos y posibilidades de reflexión y decisión sobre gran diversidad de temas, con independencia de la ubicación de los usuarios, al suprimir las barreras geográficas para la participación (Carcar, 2015). Permiten, asimismo, la proliferación de los foros de debate público, la adopción de mejores soluciones, más representativas y realizar estudios de opinión de forma inmediata, entre otras.

Fischer y Reuber afirman que la utilización de medios sociales propicia nuevas modalidades de interacción social, así como facilitan las cogniciones y la conducta de quienes emplean estos recursos. En este sentido los elementos simbólicos que emplean los movimientos sociales resultarían fácilmente interiorizados y motivarían hacia la acción social, etiquetas, mensajes cortos o mensajes vía *WhatsApp*, a pesar de su aparente simplicidad tendrían un impacto importante en los usuarios (Fischer y Reuber, 2011).

En este ámbito adquiere particular importancia la red social Twitter, desde hace varios años ha llegado a convertirse en importante instrumento para el desarrollo de la labor política, con una participación protagónica de movimientos sociales de la más variada orientación. Y es que, en esta, un solo tuit puede llegar a tornarse en tema de agenda mediática y alcanzar niveles de repercusión global insospechados.

Surgida el 21 de marzo de 2006, fruto del ingenio de Jack Dorsey y Biz Stone, esta excelente plataforma tiene entre sus características principales facilitar el desarrollo en Internet de una marca o identidad, ya sea personal, empresarial o institucional. Con sus distintas versiones para la Web, dispositivos móviles y clientes

oficiales para diversos teléfonos inteligentes y las miles de aplicaciones desarrolladas por terceros, proveen un ecosistema altamente aprovechable por personas y empresas (Barrera, 2017).

Basada en la filosofía del *microblogging*, los usuarios se comunican a través de pequeños fragmentos de información (tuits), con un máximo de 280 caracteres. Según Antonio Cambronero, uno de los más destacados blogueros de habla hispana, *Twitter* "se ha convertido en el sistema nervioso central de la sociedad conectada y maravilla por su concepto fuerte..., por su alcance global –acuerdos de acceso SMS con más de 90 países–, por su versatilidad –accesible desde móviles incluso sin servicio de Internet, teléfonos inteligentes, ordenadores de sobremesa, portátiles y tabletas–, por su instantaneidad –factor crucial en movilizaciones ciudadanas y catástrofes naturales– y, sobre todo, porque es divertido y social" (Cambronero, 2016).

Entre los aportes de este sitio de redes sociales se encuentra el uso de los populares *"hashtags"*, o etiquetas "#", los que fueron propuestos por el usuario Chris Messina, el 23 de agosto de 2007. En la actualidad Twitter está disponible en 33 idiomas, se dice que su base tecnológica es capaz de manejar 18 quintillones de seguidores -10^{30}; esto es, un millón de cuatrillones (Cambronero,2016). El 25 de enero de este año, contaba 326 millones de usuarios activos (Hootsuite, 2019).

En particular, el uso de las etiquetas tiene considerable trascendencia; de ahí que la información sobre el propósito del movimiento social en cuestión, debe darse a partir de la decisión de la propia etiqueta elegida. Para Chang, desde la teoría de la difusión de innovaciones, la posibilidad de adopción de las etiquetas depende de la medida en que el usuario ha expuesto su información a través de Twitter con alguna aplicación instalada en su teléfono móvil (Chang, 2011). Pero allí no queda: el empleo de las etiquetas representa un factor de importancia para la identidad del movimiento social, aún para quienes no realizan una presencia activa en las acciones colectivas a este respecto. Autores como Dahlberg, Grundberg y Lindgren señalan que en tanto los seguidores de un movimiento usen las etiquetas se involucrarán en mayor medida, sin necesidad de estar presentes en los lugares de protesta (Grundberg, y otros, 2014).

En Cuba, desde hace varios años, se han desarrollado múltiples estudios aplicando el análisis de redes sociales. El Centro de Ideoinformática (CIDI), adjunto a la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), haciendo uso de la plataforma Twitter, enfrenta el reto de mantenerse informado sobre los principales MSR y su impacto en la red de redes. Por tal motivo, lo referente a este tema en el centro

cuenta con el Departamento de Operaciones Web y Análisis de Información (DOWAI), que tiene entre sus funciones el análisis del impacto de los MSR. Esta tarea se realiza de forma manual, repercutiendo de forma negativa sobre la ecuación «Resultados» / «Tiempo». En opinión de los especialistas del centro eso se traduce en: mayor esfuerzo, menor efectividad y afectación sobre otras tareas que podrían ser ejecutadas con mayor prontitud existiendo una herramienta que optimice este proceso.

A la hora de evaluar el impacto de los MSR, a partir de las funciones de las RSI, la tendencia hasta hoy ha sido valorarlos como un todo y plantear determinados niveles de influencia en los acontecimientos sin un basamento científico debidamente fundamentado. En tal sentido, el MSc. Waldo Barrera, ha propuesto realizar el análisis a partir de tres dimensiones, debido a que su repercusión en cada una de ellas pudiera ser diferente en lo relativo a impacto:

- "Como recursos de información: constituyen fuentes de conocimiento sobre los movimientos y sus agendas, para dar a conocer las acciones desarrolladas por estos en los ámbitos tanto nacional como internacional. Permiten romper eventuales barreras impuestas por los gobiernos y la maquinaria mediática al servicio de estos, logrando en no pocas ocasiones superar las acciones de bloqueo aplicadas por las fuerzas en el poder. No solo en calles y plazas; sus reivindicaciones y posiciones políticas circulan también a nivel mundial y logran visibilidad en los medios de comunicación de masas.
- "Como recursos de comunicación y coordinación: Facilitan la participación en los debates públicos y la realización de intercambios y coordinación de acciones entre los diferentes actores. Incluye la coordinación informativa –para reforzar el conocimiento y la participación–; coordinación propositiva –para promover la acción–y la coordinación reactiva –de carácter defensivo. Los MSR no necesitan del apoyo de los grandes medios para coordinar y actuar; se comunican a través de la telefonía móvil o de Internet –RSI, foros y sitios webs. La difusión generalizada del mensaje entre amigos y conocidos, posibilita la creación de una cadena de comunicación capaz de movilizar a millones de personas.
- "Como recursos para captar adeptos y generar solidaridad en los espacios inmediatos, nacionales, e incluso a nivel global. Las RSI, han permitido sensibilizar y promover grandes manifestaciones de apoyo en todo el orbe; por ejemplo, las protagonizadas en contra de los organismos y entidades rectoras de las políticas neoliberales mundiales" (Barrera, 2017).

De estas, seleccionamos para informatizar en la primera versión del sistema informático AMSOR, el análisis del impacto en la dimensión Recursos de información, llegando a la evaluación del impacto que alcanza la etiqueta distintiva del MSR en cuestión.

Dada la situación problemática expuesta anteriormente se identifica como **problema de investigación**:
¿Cómo elevar la capacidad de análisis del impacto de los MSR en la dimensión Recursos de información en Twitter por parte de DOWAI?

Se define como **objeto de estudio:** Análisis de redes sociales, tomando como **campo de acción:** El estudio del impacto de los MSR en la dimensión recursos de información en Twitter.

Para dar respuesta al problema antes mencionado se traza como **objetivo general:** Desarrollar una aplicación que permita determinar el impacto de los MSR en la dimensión recursos de información en Twitter.

En correspondencia con el problema de investigación, el objeto de estudio, el objetivo general y el campo de acción se determinó la siguiente **hipótesis:**
Con la implementación de un sistema para determinar el impacto de los MSR en la dimensión recursos de información en la red social Twitter, se incrementará la capacidad de análisis del desempeño en Internet de los movimientos sociales por parte del departamento DOWAI.

Se proponen los siguientes **objetivos específicos:**

- Elaborar una fundamentación teórica.
- Diseñar un sistema para el análisis del impacto de los MSR en la dimensión “recursos de información en Twitter”.
- Implementar un sistema para la medición del impacto MSR en Twitter.
- Validar el resultado obtenido en la investigación mediante la ejecución de pruebas.

Durante el desarrollo de esta investigación se hizo necesario profundizar en el estudio de los temas abordados, por lo que se utilizaron varios métodos científicos de investigación:

Métodos teóricos:

- **Histórico-Lógico:** Permite entender el surgimiento, las características actuales, conceptos y términos y, en específico de la red social Twitter.
- **Analítico-Sintético:** Se emplea con el fin de hacer un correcto, amplio y sintetizado análisis de la bibliografía existente posibilitando extraer las ideas esenciales y relevantes relacionadas con el objeto de estudio. A través de su empleo se definen los conceptos y principios relacionados con las RSI y los MSR.

- **Modelación:** Se utiliza para modelar el sistema web y una interfaz web de administración para la medición del impacto de una etiqueta en la red, a través de la representación gráfica de los contenidos tales como: diagramas de clases y diagramas de secuencia, utilizando las herramientas necesarias que permiten el establecimiento de las relaciones entre los elementos del problema y el objeto de estudio.

- **Análisis documental:** Para la consulta de la literatura especializada y la extracción de los referentes teóricos para la presente investigación.

- **Sistémico estructural:** Para la descripción de la arquitectura del sistema informático desarrollado, los protocolos y tecnologías implicadas en el funcionamiento de la misma, así como la relación entre los componentes de esta y los servicios ofrecidos por la plataforma Twitter.

Métodos empíricos:

- **Entrevista:** Se desarrolla una entrevista al personal del área en cuestión, con el objetivo de obtener toda la información necesaria, así como sus necesidades y posibles resultados esperados (Ver Anexo 1).

Para un mejor entendimiento, tanto del problema como de la solución propuesta, el trabajo de diploma se ha estructurado de la siguiente forma: Resumen, Introducción, tres Capítulos, Conclusiones, Recomendaciones, Glosario de términos, Referencias bibliográficas, Bibliografía y Anexos. A continuación, se desglosa brevemente el contenido abordado en cada capítulo.

Capítulo 1: Fundamentación teórica

En este capítulo se abordan los conceptos fundamentales relacionados con el problema a resolver. Se realiza un estudio de la metodología de desarrollo de software, se exponen los lenguajes de programación, el lenguaje de modelado, las herramientas y tecnologías fundamentales a utilizar.

Capítulo 2: Propuesta de solución

En este capítulo se elabora una presentación de la propuesta de solución a la problemática. Se determinan las funcionalidades y características del sistema. Se determinan los patrones de diseño a utilizar y elabora la descripción de los requisitos.

Capítulo 3: Implementación y pruebas

En este capítulo se elabora la propuesta de solución, la página web que da acceso a la inclusión de las etiquetas de los MSR y los órganos de prensa. Se muestra el tuit de la etiqueta seleccionada y su impacto en la red basado en la fórmula matemática. Se realizan las pruebas pertinentes al software para verificar su correcto funcionamiento.

Capítulo 1: Fundamentación teórica.

En este capítulo se exponen los conceptos fundamentales relacionados con las redes sociales. Se realiza una descripción de los antecedentes del análisis de redes sociales, tales como el análisis de redes sociales, y las métricas. Se realiza una descripción, tuit entre otros. Se realiza un análisis sobre las herramientas, metodología de desarrollo de software, modelado y lenguajes de programación a emplear.

1.1 Antecedentes del análisis de redes sociales.

Los predecesores del análisis de las redes sociales en el siglo XVIII, incluyen a Émile Durkheim y a Ferdinand Tönnies (Figueiras, 2014). Este último planteó que los grupos sociales pueden existir como relaciones sociales formales e instrumentales. Émile Durkheim dio una explicación no individual de los fenómenos sociales, ya que estos nacen cuando las personas que interactúan forman parte de una realidad que ya no puede explicarse, en base a las características de los actores individuales (Cronin, y otros, 2016).

En 1929, el escritor húngaro Frigyes Karinthy, en un relato corto titulado Cadenas, propuso la Teoría de los seis grados de separación, desarrollada por Stanley Milgram años más tarde. Esta plantea que todas las personas del planeta están conectadas a través de no más de seis enlaces. Las redes sociales parten de dicha teoría, bajo el presupuesto de que el número de conocidos de una persona crece exponencialmente con el número de enlaces de la cadena, y solo un pequeño número de enlaces son necesarios para que el conjunto de conocidos se convierta en la población humana entera (Figueiras, 2014).

En los años 30 del siglo XX, Jacob L. Moreno fue pionero en el registro sistemático y en el análisis de la interacción social de pequeños grupos, especialmente dentro de la rama de la sociometría, su trabajo más relevante empezó una iniciativa basada en la improvisación teatral, lo que daría paso a una propuesta psicoterapéutica a la que llamó psicodrama. Por otro lado, autores pertenecientes a la universidad de Harvard, liderados por W. Lloyd Warner y Elton Mayo, examinaron las relaciones interpersonales en el trabajo. Llegada la década de los 40 del mismo siglo, los antropólogos británicos A.R. Radcliffe-Brown instaron al estudio sistemático de las redes, pero no fue hasta después de 15 años que sus saberes se siguieron de forma sistemática (Figueiras, 2014).

En 1950, en Inglaterra, Elizabeth Bott llevó a cabo estudios de parentesco y en la Universidad de Manchester

un grupo de antropólogos realizaron investigaciones de urbanización, con lo que se desarrolló el análisis de redes sociales (Figueiras, 2014). El concepto de red social fue introducido por John Arundel Barnes en 1954, quien al estudiar la organización social de una parroquia en Noruega. planteó el modelado de estas relaciones como una red en la que las personas o grupos de estas se representan como puntos unidos por líneas, que indican la interacción entre un par de personas o grupos determinados (Barnes, 1954).

Varios fueron los académicos que investigaron entre los años 1960 y 1970 la combinación de diferentes temas y tradiciones. Entre ellos destacan Harrison White, estudiantes del Departamento de Relaciones Sociales de la Universidad de Harvard y Charles Tilly, quien se centró en la sociología política y los movimientos sociales (Figueiras, 2014).

1.2 Marco conceptual.

1.2.1 Análisis de Redes sociales.

El análisis de redes sociales comprende un conjunto de técnicas para la cuantificación e interpretación de las relaciones entre las entidades sociales. Estas entidades pueden representar a individuos o grupos de individuos tales como: organizaciones, comunidades, equipos, o alianzas. El enfoque basado en las entidades sociales es el reflejo del hecho de que el análisis de redes sociales tiene su origen en el campo de la sociología y la antropología, para el estudio de las interacciones sociales de una forma práctica. Una de las características del análisis de redes sociales es la visualización de las relaciones de forma simultánea tanto a nivel individual como global (Cronin, y otros, 2016) (Espinosa, 2018).

Al alcanzarse un mejor entendimiento sobre las dinámicas de las relaciones entre los actores de la red, es posible cuantificar, monitorear y evaluar el flujo de la información. Lo que puede ser empleado para mejorar el desempeño de las organizaciones dentro de la red social. Según Serrat (Serrat, 2017) el resultado del análisis de redes sociales puede ser aplicado, entre otros objetivos, para:

- Identificar los individuos, grupos y unidades que juegan un rol relevante dentro del funcionamiento de la red.
- Descubrir cuellos de botella, segmentos aislados, y otras barreras para el flujo de la información.
- Identificar oportunidades para acelerar la difusión de la información.
- Fortalecer la eficiencia y efectividad de los canales para la comunicación.

Según Wasserman y Faust, en la base del análisis de redes sociales se encuentran los conceptos del actor, el enlace, la pareja, el subgrupo, el grupo, la relación y la red. El actor se define como un individuo discreto, o un colectivo de unidades sociales. La característica que define a un enlace es el establecimiento de una relación entre un par de actores. Los enlaces pueden representar diferentes tipos de relaciones, ser unidireccionales o bidireccionales, y solo existen entre un par de actores específicos (Wasserman, y otros, 1994) (Espinosa, 2018).

La pareja de actores se establece al concretarse, al menos, un enlace relacional entre estos, siendo este una característica inherente de la pareja. Una misma pareja de actores puede mantener más de un enlace al mismo tiempo. El subgrupo se conforma por un subconjunto de actores y todos los enlaces entre estos. Mientras que un grupo consiste en un conjunto finito de actores, sobre los que se realiza el estudio de la red; la relación se define como la colección de enlaces de un mismo tipo entre todos los miembros de un grupo (Wasserman, y otros, 1994) (Espinosa, 2018).
Siguiendo las anteriores definiciones, y adaptando las al sitio de redes sociales Twitter, se asume para la investigación, que:

• Los usuarios se corresponden con la definición del actor.
• Los enlaces que se pueden establecer entre los actores representan las relaciones de seguimiento, mención, retuit, replica, cita y contribución.
• Todos los enlaces se caracterizan por ser unidireccionales.

El análisis de redes sociales se clasifica en dos tipos principales, el análisis global de la red y el análisis de redes centrado en el actor. El primer tipo se basa en las interacciones entre los actores dentro de un entorno delimitado. Esta delimitación, geográfica o social, condiciona la naturaleza del análisis a realizar. El objetivo de este tipo de estudio es recolectar información sobre cada miembro del grupo y de las interacciones entre estos (McCarty, y otros, 2015) (Espinosa, 2018). El análisis de redes centrado en el actor, no está geográfica o socialmente delimitado. El propósito del análisis centrado en el actor es operacionalizar el contexto social del individuo en un conjunto de variables que puedan ser utilizadas para obtener información sobre el actor. A diferencia del análisis global de la red, en el cual los actores son conocidos con antelación, en los estudios centrados en el actor, estos son los que indican cuáles son sus contactos, expandiéndose la red con estos nuevos actores (McCarty, y otros, 2015) (Espinosa, 2018).

Siguiendo lo planteado McCarty y Molina la presente investigación se encuentra enfocada hacia el análisis global de la red, debido a que se conocen con antelación el conjunto de usuarios que participarán en el estudio.

1.2.2 Red Social.

El concepto de red social ha evolucionado desde que fuera enunciado por Arundel en 1954, la presente investigación se acoge al concepto de red social presentado por Richard Luciano García y C. Rafael Isequilla Lima, quienes expresan que "Una red social es una estructura social hecha por individuos u organizaciones llamados nodos". Los nodos (que a menudo representan a los individuos) están conectados por uno o más tipos específicos de relaciones. "Una red social es un conjunto de actores vinculados entre sí" (García, y otros, 2014). Los actores pueden ser personas o grupos de estas: empresas, comunidades, organizaciones, países, ciudades, etc. Los vínculos son cualquier cosa que relacione a los actores, ejemplo (amistad, parentesco, trabajo) y que permiten el contacto entre estos, de manera que se puedan comunicar e intercambiar información (García, y otros, 2014).

1.2.3 Red social en internet (RSI).

Existen diversas definiciones para los sitios de redes sociales en Internet, entre ellas sobresalen las definiciones de (García, y otros, 2014) y (Boyd, y otros, 2007).

Según Boyd y Elison un sitio de redes sociales constituye un "servicio sobre la Web que permite a las personas construir un perfil público o semipúblico dentro de un sistema cerrado, articular una lista de otros usuarios con los que se comparte una conexión, observar y navegar a través de sus listas de conexiones, así como por aquellas hechas por otros dentro del sistema" (Boyd, y otros, 2007).

Mientras que García y Lima en el año 2014 plantearon que los RSI son "Páginas que permiten a las personas conectar con sus conocidos, familiares y amigos, incluso realizar nuevas amistades, a fin de compartir contenidos, interactuar, crear comunidades sobre intereses similares: trabajo, lecturas, juegos, amistad, relaciones interpersonales" (García, y otros, 2014). Un conjunto de personas que se relacionen entre sí de manera física y sin la intervención de tecnologías es también una red social.

Para la presente investigación, tomando como punto de partida lo expresado en ambas definiciones se asumen los sitios de redes sociales en Internet como: Un servicio sobre la Web en el cual los usuarios administran un perfil, público o privado, desde el cual se comparten diversos contenidos y se interactúa con otros usuarios, con los que se ha establecido algún tipo de relación, basada en intereses compartidos, dentro del marco de un sistema cerrado. Estas relaciones pueden ser unidireccionales o bidireccionales, en dependencia de las características de la propia red social en Internet.

1.2.4 Movimiento Social (MS).

Según Reichmann & Fernández, constituyen el «agente colectivo que interviene en el proceso de transformación social, promoviendo cambios, u oponiéndose a ellos» (Reichman, y otros, 1994). Los estudios sitúan el origen de los MS a mediados del siglo XIX, considerando al movimiento obrero como el primero de ellos. En los años 60 del pasado siglo, describen los rasgos que diferencian a los nuevos movimientos sociales (NMS) de los viejos movimientos sociales de la modernidad (Barrera, 2017).

1.2.5 Movimiento Social en la Red (MSR).

Rocío Ortiz, los define como actores colectivos, estructurados en forma de red distribuida, que de manera intencional y con cierta continuidad, utilizan las oportunidades comunicativas de la era de Internet y sus redes sociales para conseguir afectar al cambio social, a través del impulso de sus acciones colectivas, las cuales pueden desarrollarse tanto en los espacios físicos como en Internet–o en ambos al propio tiempo–, con el objetivo de sensibilizar a la opinión pública sobre un conflicto social y unos objetivos públicos que se reivindican desde una identidad colectiva establecida (Ortiz, 2016).

1.2.6 Tweet.

Un tuit es una publicación o actualización de estado realizada en Twitter. Como tal, un tuit es un mensaje cuyo límite de extensión son 280 caracteres. La palabra *tweet* podría traducirse al español como trino, pío o gorjeo, en alusión al sonido que hacen los pájaros. De allí que el icono de la marca Twitter sea un pajarito. En español es correcta la adaptación fonética tuit. También puede emplearse la palabra trino.

Puede contener letras, números, signos y enlaces. Los tuits, además, pueden contener hashtags o etiquetas, que permiten establecer el tema o enfoque que se le pretende dar a la publicación, o relacionarlo con un tema de conversación que se encuentra en las tendencias del momento.

Términos relacionados con el mundo de Twitter son tuitero (usuario), el verbo tuitear o trinar, para designar la acción de hacer una publicación en Twitter; tuiteo, para referirse a la acción y efecto de tuitear (Significados, 2019).

1.2.7 Retweet.

Se denomina retuit a un tuit compartido de manera pública con los seguidores. Es una excelente forma de difundir noticias y descubrimientos interesantes en Twitter. Existe la opción de agregar los propios comentarios a un *tweet* antes de retuitearlo. Cuando se usa el ícono retuitiar de Twitter, el *retweet* o *retweet* con comentario, harán referencia al tuit compartido. Cuando alguien responde al *retweet* con comentario, el autor del tuit original no se agregará automáticamente a la conversación. Si se desea incluir el autor del tuit original, es preciso mencionar su nombre de usuario.

Esta función es muy útil si se desea volver a publicar uno de los tweets propios antiguos porque ha vuelto a cobrar relevancia, o para retuitiar las respuestas a otras personas si precisa asegurarse de que todos los seguidores las vean (Twitter, 2018).

1.2.8 Favorito.

Hacer clic en favorito o me gusta en una publicación de alguna red social como Facebook, Twitter e Instagram es una forma de decirle a las personas que es del agrado de quien lo hace sin dejar un comentario. Tal como ocurre con los comentarios, cualquiera que pueda ver la publicación podrá percatarse que han indicado que les gusta.

Por ejemplo, si se hace clic en favorito debajo del vídeo de un amigo:

- Las personas que puedan ver el vídeo también verán que se ha indicado que gusta.
- La persona que publicó el vídeo recibirá una notificación informándole de que se ha indicado que gusta (Twitter, 2018).

1.2.9 Teoría de Grafos aplicadas a redes sociales.

Para Wasserman, Faust y Espinosa, la teoría de grafos constituye una herramienta para la modelación de una red social, mediante la representación de los actores como nodos y los enlaces entre estos como aristas. La representación matricial del grafo facilita las labores de cálculo. Para esto se genera una matriz en la que el valor de una celda representa la existencia, o no, de un enlace entre los actores correspondientes a la fila y la columna (Wasserman, y otros, 1994)(Espinosa, 2018).

Un segmento de una red social puede ser modelado como un grafo G, el cual se compone por un conjunto de vértices V y un conjunto de enlaces E entre los distintos pares de vértices v, tales que $v \in V$. Un camino se constituye por una secuencia alterna $(v_0, e_1, v_1, \dots, e_n, v_n)$ de vértices v y enlaces $e \in E$ de tal forma que el enlace e_i conecta al vértice en la posición $i-1$ con el vértice en la posición i. La longitud del camino está determinada por la cantidad de enlaces que lo componen. El conjunto de vecinos para un vértice v_0 se define como $\partial\{v_0\} = v_n \in V | \{v_0, v_n\} \in E$ (Espinosa, 2018)

1.2.10 Métrica.

“Es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias” (Figueiras, 2014).

1.2.11 Métricas de centralidad.

La centralidad de los nodos, o la identificación de cuáles nodos son más centrales que otros es un elemento clave del análisis de redes sociales. Esto es siguiendo el principio de que el nodo con una mayor centralidad cuenta con una ventaja sobre el resto en lo referente a la capacidad de influir sobre la dinámica de la red. Si se toma como base a una red ideal, este nodo tiene una mayor cantidad de enlaces o forma parte de enlaces relevantes en el grafo, por lo que partiendo de este se puede alcanzar al resto de los nodos con mayor facilidad, controlando el flujo de la red (Opsahl, y otros, 2010).

En el marco de la teoría de grafos y el análisis de redes sociales se cuenta con varias métricas de centralidad para un nodo perteneciente a un grafo conexo, que aportan información sobre la importancia del nodo con relación al grafo. Existen varias métricas de centralidad que son ampliamente utilizadas en el análisis de redes sociales: centralidad de grado, intermediación, cercanía, excentricidad, media armónica, PageRank, HITS y eigenvector (Wasserman, y otros, 1994) (Kleinberg, 1999) (Opsahl, y otros, 2010) (Hautz, y otros, 2016) (Huang, y otros, 2014) (Espinosa, 2018).

En la presente investigación se aborda el impacto de las publicaciones relacionadas con las etiquetas de los movimientos sociales en Internet. Las métricas de PageRank, Hits cercanía, media armónica intermediación y eigenvector suelen emplearse para el estudio de la relevancia de los nodos o actores dentro de la estructura de la red. En el caso de la presente investigación se analiza el impacto de las etiquetas a

partir de las relaciones de favoritos y retuits, para publicaciones que estén relacionadas con las etiquetas estudiadas. Desde la perspectiva de la presente investigación, en todos los casos se construyen grafos limitados a la vecindad inmediata del nodo que emite la publicación. Es por esta razón que solo se debe tomar en cuenta la métrica de centralidad de grado, en el cálculo del índice de relevancia.

1.2.12 Centralidad de grado.

La centralidad de grado se calcula como el total de nodos que son directamente adyacentes al nodo estudiado (Wasserman, y otros, 1994). En el caso de que las relaciones sean direccionadas esta métrica puede subdividirse en dos: grado de entrada y grado de salida. El grado de entrada representa la cantidad de nodos que mantienen un enlace cuyo destino es el nodo estudiado. Mediante esta métrica se puede estimar la "popularidad" del actor dentro de la red social. En el caso del grado de salida se refiere al total de nodos que son el destino de los enlaces que tienen como origen al nodo estudiado (Espinosa, 2018).

El grado de salida representa la "sociabilidad" del actor analizado (Passmore, 2011). Esta métrica se distingue por ofrecer una valoración sobre cuan involucrado se encuentra un nodo en la red. Su simplicidad radica en el hecho de que solo es necesario conocer las estructuras locales en las cercanías inmediatas al nodo estudiado. Sin embargo, tiene la limitación de no tomar en cuenta la estructura global de la red. Aunque un nodo puede presentar una alta centralidad de grado, estando conectado directamente con muchos otros nodos de la red, puede que no se encuentre en una posición desde la cual se pueda acceder rápidamente a otros nodos (Opsahl, y otros, 2010).

Para un grafo dado $G := (V, E)$ donde el conjunto V de vértices representa a los actores y el conjunto de aristas E a los enlaces entre los actores. Siendo n el cardinal del conjunto de vértices, la centralidad de grado de salida $COut(v)$ para un vértice v se define en la ecuación 2 (Berge, 1985).

$$COut(v) = |\{y \in V \mid \{v, y\} \in E\}|$$

Ecuación 1. Grado de salida.

Siguiendo el mismo principio, la centralidad de grado de entrada $CIn(v)$ se define en la ecuación 3 como la media del grado de entrada del vértice para la cantidad de nodos del grafo (Berge, 1985).

$$CIn(v) = |\{y \in V \mid \{y, v\} \in E\}|$$

Ecuación 2. Grado de entrada.

1.2.13 Impacto de las publicaciones en los sitios de redes sociales.

Los sitios de redes sociales en Internet, se han convertido en un canal de comunicación directa con el individuo. Por esta razón se han transformado en objeto de interés para la realización de estudios sobre la efectividad de las publicaciones. Se han realizado investigaciones centradas en el usuario para determinar la posición de estos en el marco de la estructura de las campañas comunicacionales de las marcas corporativas (Langrado, y otros, 2018).

También se han desarrollado métodos complejos para evaluar el impacto de las publicaciones en las redes sociales. Estos métodos se basan en el análisis de los grafos que se generan a partir de las relaciones de intercambio de contenidos, y sirven como una medida de la influencia de los usuarios estudiados (Brunn, y otros, 2018).

Tanto para el método desarrollado por Langrado y el método descrito por Brunn, es necesario construir un grafo de nivel dos o superior. En el caso de la presente investigación al abordarse el estudio del impacto de los movimientos sociales, mediante las relaciones de favoritos y retuits, haciendo uso del API de búsqueda de Twitter no es posible construir un grafo de nivel superior a uno. Por lo que es necesario evaluar el impacto haciendo uso de la métrica de centralidad de grado, que brinda una descripción de las relaciones del nodo estudiado con su vecindad inmediata.

1.3 Estudio de herramientas homologas a la investigación.

Para la investigación se toman las siguientes herramientas como referencias a evaluar ellas son:

1.3.1 Twitter search.

Twitter Search es la base de los buscadores desarrollados para Twitter. Su sistema de búsqueda avanzada permite personalizar búsquedas, y además señala los temas más mencionados en el momento, así como las tendencias.

Ofrece en los resultados de búsqueda la posibilidad de filtrarlos por el tipo de contenido, orden cronológico, y la calidad de las publicaciones o cuentas entre otros. Se caracteriza por ser de uso gratuito (Twitter, 2018).

En el caso de la aplicación Twitter Search, la valoración del impacto de una etiqueta no es el único factor que interviene en el orden de los resultados.

1.3.2 Hashtracking.

La herramienta Hashtracking, es una aplicación web que ofrece el servicio de análisis de etiquetas en sitios de redes sociales. Permite la generación de reportes sobre la actividad de los usuarios, así como la detección de usuarios influyentes para un tópico determinado. Evalúa el alcance de las publicaciones de acuerdo a las características estructurales de estas y el comportamiento de los usuarios.

Ofrece informes sobre los tuits y retuits, alcance o impresiones para las etiquetas seleccionadas. Para su utilización es necesario suscribirse a alguno de los planes que ofrece, los cuales van desde 50.00 USD hasta 1500.00 USD mensuales (Hashtracking, 2010).

1.3.3 Hashtags.org.

El sitio web Hastags.org facilita el análisis de las etiquetas más empleadas en el marco de las últimas 24 horas. Permite conocer las etiquetas más populares, así como clasifica a los usuarios que hacen uso de estas. Esta aplicación cuenta con un diccionario propio de etiquetas, en el que estas se relacionan con usuarios puntuales. Permite la clasificación de las etiquetas populares de acuerdo a temáticas específicas. Para hacer uso de esta aplicación es necesario adquirir alguno de los paquetes de pago, los cuales van desde 49.00 USD hasta los 349.00 USD al mes (Hashtags.org, 2006).

1.3.4 HASHTAGIFY.ME.

Es una aplicación web que permite explorar la popularidad, o la tendencia de una etiqueta en Twitter, además de su relación con otros. Permite la exploración del uso de las etiquetas por usuarios influyentes en Twitter. Cuenta con las facilidades necesarias para monitorear las etiquetas analizar el alcance de estas y su popularidad.

Además, permite la comparación del rendimiento entre cuentas de Twitter, así como la construcción de reportes y métricas personalizadas. Esta aplicación está disponible mediante el pago de cuotas mensuales o anuales (Hashtagify.me, 2011).

1.3.5 Metricool.

Es una herramienta que te permite analizar, gestionar y medir el éxito de los contenidos digitales en diversas redes sociales. Ofrece herramientas para la planificación de la estrategia social en un único sitio, desde analítica web y de contenidos para blogs, métricas y planificación de redes sociales, tales como Facebook, Instagram, Twitter y LinkedIn, hasta la realización de estudios de otras cuentas (Metricool, 2019).
Esta aplicación mide el impacto y alcance de las etiquetas en Twitter, para esto es necesario contar con una cuenta Premium.

1.3.6 Selección de herramienta homologa.

Cada una de ellas aporta una perspectiva diferente en la evaluación del uso de las etiquetas. En el caso de las aplicaciones Hashtags.org, Hashtracking, el análisis no abarca las relaciones de favoritos, limitándose a la descripción de los usuarios y la clasificación de los tuits en mensajes, tuits originales o retuits. Las aplicaciones Hashtagfy.me y Metricool se centran en el análisis de los contenidos generados por una cuenta de Twitter asociada y las reacciones de los seguidores de este.

Para el caso del estudio de los movimientos sociales en Internet, el análisis se centra en las etiquetas relacionadas con estos, las cuales pueden ser empleadas por diversos usuarios, en distintos momentos. Por lo que las aplicaciones anteriormente descritas no se ajustan a las necesidades de la investigación. La aplicación Twitter Search, es la base para el funcionamiento del API de Búsqueda de Twitter, la cual será empleada para la recuperación de los tuits sobre los que se realizará el análisis.

1.4 Selección de herramientas y tecnologías.

1.4.1 Lenguajes de programación.

PHP 7.2

Es un lenguaje de programación web de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor (Achour, y otros, 2006). Este lenguaje de programación nace en la década de los 90 y actualmente es uno de los lenguajes de programación más importantes y populares. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

A grandes rasgos la lógica de PHP es la siguiente: cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Este procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo, obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Este lenguaje además permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite (Achour, y otros, 2006).

Java Scripts

Es lenguaje de programación que se utiliza esencialmente en el desarrollo de páginas web. Es interpretado y que no requiere compilación. Está basado en prototipos, es imperativo, débilmente tipado y dinámico. Los navegadores web en sus versiones más recientes interpretan código JavaScript. Permite añadir propiedades interactivas a las páginas web de manera de forma sencilla. Se basa en la utilización de guiones que son integrados directamente al código HTML por lo que el código es transferido al cliente para que este lo interprete al cargar la página de esta forma si no llena correctamente un formulario no tiene que esperar a que el servidor muestre de nuevo el formulario indicando los errores existentes.

Java

Java es un lenguaje de programación y la primera plataforma informática creada por *Sun Microsystems* en 1995. Es un lenguaje moderno, de alto nivel, que recoge los elementos de programación que típicamente se encuentran en todos los lenguajes de programación, permitiendo la realización de programas profesionales (Hommel, y otros, 1999). En este trabajo se utiliza por la rapidez, seguridad y fiabilidad que presenta, aparte de que contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones.

Python

Es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación identada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan

elementos como las llaves o las palabras clave *begin* y *end*. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar (Alvarez, 2003).

C++

Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

El nombre **C++** fue propuesto por Rick Mascitt en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C (ICTA, 2019).

Selección de lenguajes de desarrollo

Para la implementación de los componentes de procesamiento se seleccionó al lenguaje de programación Java en su versión 8. Este lenguaje aporta rapidez en tiempo de ejecución y se cuentan con varias librerías que facilitan su integración con los servicios de Twitter.

Para el desarrollo de la aplicación web que cumple la funcionalidad de interactuar con el usuario se seleccionó el lenguaje de programación PHP en su versión 7.2. Este lenguaje al ser débilmente tipado e interpretado aporta agilidad en el proceso de desarrollo. Al desplazarse el procesamiento y la interacción con los servicios de Twitter a los componentes implementados en Java, se simplifica las tareas a ejecutarse por el servidor Web, ajustándose PHP a estos requerimientos. Como leguajes secundarios se seleccionan JavaScript, HTML y CSS.

1.4.2 Marcos de trabajo para aplicaciones web.

Symfony 3.4

Es un marco de trabajo para PHP empleado en el desarrollo de aplicaciones y sitios web rápidos y seguros. Su código, y el de todos los componentes y librerías que incluye, se publican bajo licencia MIT de software libre. Cuenta con un sitio online en el cual se puede encontrar una gran cantidad de documentación, también de forma gratuita. Symfony 4 es su versión más reciente, se anunció por primera vez en el año 2014 y propone un cambio radical tanto en la arquitectura como en la filosofía de trabajo de versiones anteriores. Está diseñado para explotar las potencialidades de PHP 7.2, lo cual supone mejoras en el rendimiento del framework. Su arquitectura está totalmente desacoplada, esto permite eliminar aquellos componentes no necesarios para el desarrollo de nuestro proyecto (Symfony, 2019).

Bootstrap 4.0

Bootstrap es un *framework* de *software* libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de *JavaScript* adicionales. Desde la versión 2.0 soporta diseños sensibles. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados, así como clases *mixin*. Desde la versión 2.0, la configuración de Bootstrap también tiene una opción especial de "Personalizar" en la documentación. Los componentes de JavaScript para Bootstrap están basados en la librería jQuery de JavaScript. Los *plugins* se encuentran en la herramienta de *plugin* de jQuery. Proveen elementos adicionales de interfaz de usuario como diálogos, *tooltips* y carruseles.

Laravel

Es un nuevo y poderoso *Framework* PHP desarrollado por Taylor Otwell, que promete llevar al lenguaje PHP a un nuevo nivel. Laravel, propone una forma de desarrollar aplicaciones web de un modo mucho más ágil. Por ejemplo, en Laravel opcionalmente podemos usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde al igual que otros *fameworks* PHP, el controlador es programado como una clase.

Laravel 'entrega la opción' de seguir usando la metodología tradicional MVC. Sin embargo, el *framework* propone una vía más rápida en PHP, la cual consiste en programar la interacción HTTP directamente como una función anónima asociada a una Ruta. Esto tiene la ventaja de reducir la cantidad de código, especialmente cuando sólo necesitamos incluir una funcionalidad.

Presenta grandes ventajas en el desarrollo web en PHP, apoyado en los avances de las nuevas versiones de PHP han ofrecido a la comunidad. Este hecho, implica que para usar Laravel necesitamos disponer de versiones modernas de PHP.

La actual versión de Laravel 3.2 requiere PHP 5.3, lo cual significa en la práctica que si tenemos código heredado que fueron programados para usar versiones antiguas del intérprete PHP, este no funcionará correctamente si lo ejecutamos en PHP 5.3 en el mismo servidor web (Rees, 2012).

Selección framework de desarrollo:

Se seleccionan los *Frameworks* de desarrollo Symfony y Bootstrap ya que permiten desarrollar rápidamente aplicaciones web, son compatible con la mayoría de los gestores de bases de datos, son multiplataforma, en el caso de Symfony utiliza programación orientada a objetos y permite una fácil integración con las herramientas a utilizar.

1.4.3 Sistema de mensajería.

RabbitMQ

Es un sistema de mensajería empresarial completo y altamente confiable basado en el estándar AMQP. Está licenciado bajo la licencia de código abierto Mozilla *Public License* y cuenta con una distribución independiente de plataformas. Debido a esto es de los más extendidos, incluso existe un módulo de *puppetlabs* que permite instalarlo, configurarlo y administrarlo. Sus características principales son:

- Garantía de entrega
- Enrutamiento flexible
- Clusterización
- Federación
- Alta disponibilidad
- Tolerancia a fallos

En concreto permite instalar un servidor (*broker*) federado compuesto por dos nodos con RabbitMQ para disponer de alta disponibilidad, administrar cada uno de ellos, mecanismos de autenticación y autorización, configurándolos de modo seguro mediante *Two-way* SSL *Authentication* (Ramos, 2014).

Redis

Es un almacén de estructura de datos en memoria de código abierto (con licencia BSD), que se utiliza como base de datos, caché y agente de mensajes. Es compatible con estructuras de datos como *strings*, *hashes*, listas, conjuntos, conjuntos ordenados con consultas de rango, *bitmaps*, *hyperloglogs*, índices geoespaciales con consultas de radio y *streams*. *Redis* tiene replicación incorporada, *Lua scripting*, *LRU eviction*, transacciones y diferentes niveles de persistencia en disco, y proporciona alta disponibilidad a través de *Redis Sentinel* y partición automática con *Redis Cluster*.

Puede ejecutar operaciones atómicas en estos tipos, como agregar a una cadena; incrementando el valor en un *hash*; empujando un elemento a una lista; conjunto de computación intersección, unión y diferencia; o conseguir el miembro con mayor rango en un conjunto ordenado (Redis, 2019).

Apache Kafka

Es una plataforma de transmisión distribuida. Esto significa que una plataforma de transmisión tiene tres capacidades clave:

- Publica y suscribe flujos de registros, similar a una cola de mensajes o un sistema de mensajería empresarial.
- Almacena flujos de registros de forma duradera y tolerante a fallos.
- Procesa flujos de registros a medida que se producen.

Kafka se usa generalmente para dos amplias clases de aplicaciones:

- Construyendo flujos de datos en tiempo real para obtener datos de manera confiable entre sistemas o aplicaciones.
- Creación de aplicaciones de transmisión en tiempo real que transforman o reaccionan a los flujos de datos.

En Kafka, la comunicación entre los clientes y los servidores se realiza con un protocolo TCP simple, de alto rendimiento y sin lenguaje. Este protocolo está versionado y mantiene la compatibilidad con versiones anteriores. Proporcionamos un cliente Java para Kafka, pero los clientes están disponibles en muchos idiomas (Kafka, 2019).

Apache ActiveMQ

Es el servidor de mensajería de código abierto, multiprotocolo y basado en Java más popular. Admite protocolos estándar de la industria para que los usuarios obtengan los beneficios de las opciones de los clientes en una amplia gama de idiomas y plataformas. La conectividad de C, C ++, *Python*, *.Net* y más está disponible. Integre sus aplicaciones multiplataforma utilizando el omnipresente protocolo AMQP. Intercambie mensajes entre sus aplicaciones web utilizando STOMP a través de *websockets*. Administra sus dispositivos *IoT* utilizando MQTT. Apoya su infraestructura JMS existente y más allá. ActiveMQ ofrece la potencia y la flexibilidad para admitir cualquier caso de uso de mensajería.

Actualmente hay dos "versiones" de *ActiveMQ* disponibles: el *broker* "clásico" 5.x y el *broker* "siguiente generación" de Artemis. Una vez que Artemis alcance un nivel suficiente de paridad de características con la base de código 5.x, se convertirá en *ActiveMQ 6* (ActiveMQ, 2019).

Fundamentación sistema de mensajería seleccionado:
De los sistemas de mensajería selecciona RabbitMQ ya que es multiplataforma, permite por medio del *plugin* amqlib la integración con Symfony, presenta flexibilidad, seguridad y su simple manipulación en cuanto al trabajo con sus colas.

1.4.4 API de Twitter.

Las API son la forma en que los programas informáticos "hablan" entre sí para solicitarse y enviarse información. Para esto, se le permite a la aplicación del software que llame a lo que se denomina punto de conexión: una dirección que corresponde a un tipo específico de información que se proporciona (generalmente, los puntos de conexión son únicos, como los números telefónicos). Twitter permite acceder a diversos servicios mediante las API para facilitarle a los usuarios crear aplicaciones que se integren con la red social (Twitter, 2018).

Los datos de Twitter son únicos y reflejan la información que los usuarios deciden compartir de forma pública. La API de Twitter ofrece un amplio acceso a los datos de la red social que los usuarios han decidido compartir con el mundo. Twitter también es compatible con una API que permiten a los usuarios administrar su propia información que no es pública (como los Mensajes directos) y brindarla a los desarrolladores que ellos han autorizado a administrarla (Twitter, 2018).

1.4.5 Librerías para integración con Twitter.

Hbc

Twitter Hosebird Client (hbc) es una biblioteca de código de cliente HTTP de Java que consume la API de transmisión de Twitter. Originalmente construido alrededor del Twitter (público) *Streaming* API, admite de forma nativa la autenticación OAuth (Moffitt, 2016).

El cliente de *hosebird* utiliza la noción de un "procesador" que procesa el flujo y coloca mensajes individuales en el *Blocking Queue* proporcionado. Proporcionamos una clase *String Delimited Processor* que debe usarse junto con los *Streaming Endpoints* proporcionados. El procesador toma como parámetro un *Blocking Queue*, en el que el cliente colocará los mensajes de cadena a medida que los transmite (Steven, y otros, 2013).

Twitter4J.

Es una biblioteca Java de código abierto, que proporciona una API conveniente para acceder a la API de Twitter. En pocas palabras, aquí es donde se puede interactuar con la API de Twitter; Es posible: publicar un tweet; obtener la línea de tiempo de un usuario, con una lista de los últimos tweets; envíar y recibir mensajes directos; buscar tweets y mucho más. Esta biblioteca garantiza realizar estas operaciones fácilmente, y también la seguridad y privacidad de un usuario, para lo cual, naturalmente, es necesario disponer de las credenciales de OAuth configuradas en la aplicación (Baeldung, 2018).

Selección de librerías de integración con Twitter.

En la presente investigación la interacción con los servicios de Twitter tiene lugar mediante el API de búsqueda, por lo que se seleccionó a Twitter4j, teniendo en cuenta que Hbc, no cuenta con esta funcionalidad.

1.4.6 IDES.

NetBeans 8.0

Es un entorno de desarrollo integrado que permite escribir, compilar, ejecutar y corregir errores a programas. Desarrollado en Java también tiene soporte para otros lenguajes de programación mediante el soporte nativo (C/C++, Ruby y PHP) y el uso de bibliotecas. Es un producto libre y gratuito que no tiene restricciones de utilización. Permite el desarrollo de aplicaciones web, de escritorio, móviles empleando las plataformas

de Java. El proyecto NetBeans es apoyado por una variada comunidad de desarrolladores y ofrece una amplia documentación y una variada selección de componentes.

Eclipse

Es un entorno de desarrollo integrado que es multiplataforma y open source. A diferencia de otros entornos monolíticos que proporcionan todas las funcionalidades las requiera el usuario o no, Eclipse emplea *plugins* lo que permite una plataforma ligera para componentes de software. Puede extenderse a otros lenguajes de programación como son C/C++ y Python (García, y otros, 2014).
Eclipse presenta diversas características entre ellas están:

Este dispone de un Editor de texto con resaltado de sintaxis, presenta la funcionalidad de compilación en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con *An*t, asistentes (*wizards*) para creación de proyectos, clases, *tests*, etc., y refactorización. A través de "plugins" libremente disponibles es posible añadir control de versiones con *Subversion* e integración con *Hibernate*.

PHP Storm

Es un IDE de PHP que realmente "obtiene" su código. Es compatible con PHP 5.3 / 5.4 / 5.5 / 5.6 / 7.0 / 7.1 / 7.2, ofrece prevención de errores sobre la marcha, mejor autocompletado y refactorización de código, depuración de configuración cero y un editor extendido de HTML, CSS y JavaScript.

- Editor de código PHP inteligente

El IDE proporciona código inteligente, resaltado de sintaxis, configuración de formato de código extendido, verificación de errores sobre la marcha, plegado de código, soporta mezclas de idiomas y más. Refactorizaciones automatizadas que tratan su código con cuidado, ayudando a realizar configuraciones globales de proyectos de forma fácil y segura.

- Análisis de Calidad de Código

  Cientos de inspecciones de código verifican su código a medida que escribe e inspeccionan todo el proyecto para detectar posibles errores u olores de código. Las soluciones rápidas para la mayoría de las inspecciones facilitan la reparación o mejora instantánea del código. Alt + *Enter* muestra las opciones apropiadas para cada inspección.

- Fácil navegación y búsqueda de código

PhpStorm lo ayuda a sortear su código de manera más eficiente y a ahorrar tiempo cuando trabaja con proyectos grandes. Salte a un método, función o definición de variable en un solo clic, o busque sus usos (JetBrains, 2019).

Selección IDE de desarrollo:
Como ide de desarrollo se selecciona NetBeans 8.0 ya que es multiplataforma, permite la integración con los lenguajes de desarrollo Java y PHP, compilación en tiempo real, depurador y autocompletado.

1.4.7 Sistema Gestor de Base de Datos.

Una de las tareas más comunes de las aplicaciones web dinámicas es la persistencia y lectura de la información en las bases de datos. *Symfony* se integra a *Doctrine*, biblioteca que se utiliza para simplificar las operaciones que se realizan en las bases de datos, posee controladores que permiten la compatibilidad con servidores de código abierto como es el caso de *MySQL* (Pacheco, 2013) (Naranjo, 2018).

PostgreSQL
En su versión 9.3 es un SGBD objeto-relacional, funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema, está distribuido bajo licencia BSD y su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto; el sistema continuará funcionando.

Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, MacOSX, Solaris, Tru64) (PostgreSQL, 2010). Todas estas características lo convierten en el candidato ideal para cualquier tipo de aplicación, ajustable a las necesidades del software y del entorno de desarrollo a utilizar.

MySQL
Es un sistema de gestión de bases de datos relacional. Su diseño multihilos le permite soportar una gran carga de información de forma eficiente. Está diseñado para entornos de producción críticos, con alta carga de trabajo, así como para integrarse en aplicaciones para ser distribuido. *MySQL* es una marca registrada de *MySQL AB*, aunque tiene una doble licencia. Los usuarios pueden elegir entre usar el producto *MySQL*

como un sistema *Open Source* bajo los términos de la licencia *GNU* (del inglés *General Public License*) o pueden adquirir una licencia comercial estándar de *MySQL AB*. “Se encuentra programado en *C* y en *C++,* probado con un amplio rango de compiladores diferentes, multiplataforma, proporciona sistemas de almacenamiento transaccionales y no transaccionales. El servidor está disponible como un programa separado para usar en un entorno de red *cliente/servidor*. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible. Los clientes pueden conectar con el servidor *MySQL* usando *sockets TCP/IP* en cualquier plataforma” (ORACLE, 2012).

Selección de gestor de base de datos.

El sistema gestor de base de datos seleccionado es MySQL*,* principalmente por ser multiplataforma lo cual permitirá ser utilizado tanto en *Windows* como en *Linux*. Otro aspecto importante es que la aplicación no requiere de consultas complejas y se pueden editar rápidamente y va muy bien de la mano de PHP el cual es uno de los lenguajes utilizados. Por estas razones MySQL es la mejor opción.

1.4.8 Servidor Web.

Nginx

NGINX, pronunciado en inglés como “engine-ex”, es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP.

Los servidores web tradicionales crean un solo hilo para cada solicitud, pero NGINX no funciona de esa manera. Como mencionamos antes, NGINX trabaja con una arquitectura asíncrona y controlada por eventos. Esto significa que los hilos similares se administran bajo un proceso de trabajo, y cada proceso de trabajo contiene unidades más pequeñas llamadas conexiones de trabajo. Toda esta unidad es la responsable de manejar los hilos de las solicitudes. Las conexiones de trabajo entregan las solicitudes a un proceso de trabajo, que también lo enviará a su turno al proceso maestro. Finalmente, el proceso maestro proporciona el resultado de esas solicitudes (Nginx, 2019).

Apache

Es un servidor web que implementa el protocolo HTTP, es software libre y se puede instalar en los sistemas operativos *Windows* y *Linux*. Es de fácil configuración, pues su estructuración en módulos permite al usuario utilizar los servicios y funcionalidades que ofrece. Se define Apache como software para el servidor de aplicaciones en su versión 2.4.34, por su flexibilidad, rapidez, porque es gratuito, multiplataforma e interpreta varios lenguajes como PHP. Estas características promueven la eficiencia y rendimiento del sistema que debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño, teniendo en cuenta la concurrencia que pueda existir; debe prestar servicios sin que se amplíen los rangos de tiempo de repuesta (APACHE, 2017).

1.4.9 Herramientas de modelado.

Visual Paradigm v 8.0

Visual Paradigm for UML es una herramienta de Ingeniería de *Software* Asistida por Computadora (CASE: *Computer Aided Software Engineering*), diseñada para la ayuda al desarrollo de software. Soporta estándares de la industria clave, tales como Lenguaje de Modelado Unificado (UML). Ofrece un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad, disponibilidad de múltiples versiones y puede integrarse con los principales *IDEs*. Es extensible mediante desarrollo de nuevos *plugins* (López, 2017). Se ha actualizado rápidamente en sintonía con el nuevo desarrollo de técnicas de modelado UML 2.1 con el propósito de generar un entorno de modelados visuales en el que se reúnen hoy todas las necesidades, tanto de software y tecnología, como de las necesidades de comunicación.

Oracle Jdeveloper

Es un entorno de desarrollo integrado desarrollado por *Oracle Corporation* para los lenguajes Java, HTML, XML, SQL, PL/SQL, Javascript, PHP, Oracle ADF, UML y otros.

Es un software propietario pero gratuito desde 2005.

Las primeras versiones de 1998 estaban basadas en el entorno *JBuilder* de *Borland*, pero desde la versión 9i de 2001 está basado en Java, no estando ya relacionado con el código anterior de *JBuilder* (Oracle, 2019)

Las últimas versiones estables son:

- Para JDK 6: 11.1.1.2.0 (noviembre de 2009)

- Para JDK 5: 10.1.3.5 (agosto de 2009).

Jude UML

JUDE (Java y UML *Developers 'Environment*) es una herramienta de modelado UML única que admite el diseño de software orientado a objetos en Java (TM) combinado con *Mind Map*. Permite convertir el mapa mental a modelos UML y viceversa.

JUDE / Professional es una versión de producto de JUDE con características especiales. Es adecuado para uso comercial y modelos de gran tamaño. Mind Map también ayuda a los desarrolladores a visualizar sus objetivos, acelerar los procesos de pensamiento y conducir a una nueva inspiración y resultados (Daida, 2005).

Características básicas de JUDE:

- Clase, Caso de uso, Secuencia, Colaboración, Cuadro de estado, Actividad, Componente, Diagramas de implementación.
- Generación de plantillas de archivos fuente Java.
- Importación de archivos fuente Java.
- Generación automática de diagramas de clase con información del modelo.

Selección herramienta de modelado:

Como herramienta de modelado se decide emplear Visual Paradigm ya que esta soporta varios de los estándares necesarios para el modelado, ofrece herramientas para la generación de reportes, genera documentos, fácil modelado de base de datos y es compatible con Linux y Windows.

1.4.10 Metodología de desarrollo de software.

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, por tal motivo es importante aplicar buenas prácticas a la hora de utilizarlas. La metodología de desarrollo de software a utilizar es la AUP-UCI, debido a que es la definida por la universidad para emplearse en los proyectos productivos. Dicha metodología es una variación de la metodología “Proceso Unificado Ágil” (AUP por sus siglas en inglés) en unión con el modelo CMMI-DEV v 1.3 (*Capability Maturity Model Integration*) (Sanchez, 2014).

La metodología AUP-UCI tiene 3 fases ellas son inicio, ejecución y cierre. La segunda fase trabaja con 7 disciplinas para el ciclo de vida de los proyectos las cuales son: modelo de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación, pruebas de aceptación. Además de esto son utilizados 4 escenarios para modelar el sistema en los proyectos.

- Escenario no.1: Proyectos que modelen el negocio con Caso de Uso del Negocio (CUN) solo pueden modelar el sistema con Caso de Uso del Sistema (CUS).
- Escenario no.2: Proyectos que modelen el negocio con Modelo Conceptual solo pueden modelar el sistema con CUS.
- Escenario no.3: Proyectos que modelen el negocio con Descripción de Proceso de Negocio (DPN) solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).
- Escenario no.4: Proyectos que no modelen negocio solo pueden modelar el sistema con Historias de Usuario (HU).

Debido a que la investigación se va a enfocar en el análisis de redes sociales y sus usuarios, se empleará el escenario 4.

1.5 Conclusiones parciales.

En la presente investigación se aborda el análisis de redes sociales del tipo global la red. Para el cálculo de la importancia de las etiquetas estudiadas solo se tendrá en cuenta la vecindad inmediata de los nodos, debido a la propia naturaleza de los datos ofrecidos por Twitter. Para esto se emplea la métrica de centralidad de grado, aplicada a las relaciones de favoritos y retuits.

Para la recolección de los tuits se seleccionó el API search de Twitter. Como lenguajes de desarrollo se adoptaron PHP y Java. Para el procesamiento asíncrono de los tuits es necesario hacer uso de las colas de mensajería, para lo cual fue seleccionado RabbitMQ.

La Fundamentación de la metodología de desarrollo permitió definir AuP-UCI como la más indicada para el desarrollo de todos los procesos y análisis del producto. El estudio de los lenguajes y herramientas propuestas, permitió seleccionar Symfony como marco de trabajo.

Capítulo 2: Propuesta de solución.

En este capítulo se aborda la solución a la problemática planteada. Se describen los procesos asociados al negocio a partir de un modelo de dominio y los principales conceptos que estos engloban, logrando una mejor comprensión, tanto del sistema como del contexto donde se desarrolla el mismo. Se describen los requisitos funcionales y no funcionales, para dar cumplimiento a los objetivos planteados. A partir de la derivación de los requisitos funcionales, se representa un diagrama de casos de uso, su descripción y la interacción con los actores, define el estilo y los patrones arquitectónicos a utilizar, precisan los patrones de diseño y realizan los diagramas de clases y de secuencia.

2.1 Descripción de la propuesta de solución.

Se implementará un sistema capaz de gestionar la información asociada a la red social Twitter, así como medir el impacto de las diferentes etiquetas que se deseen buscar. También se podrán analizar estos resultados mediante la visualización de los resultados de los impactos, lo que constituirá un apoyo a la toma de decisiones en la Universidad de las Ciencias Informáticas.

Se fundamenta la evaluación del impacto por medio de la siguiente ecuación matemática:

$$\frac{(\sum_{i=1}^{n} u_i * p) * \left(\sum_{j=1}^{m} (v_j + r_j)\right)}{c}$$

Ecuación 3. Métrica de impacto

Donde el impacto de un tuit es una medida numérica que depende de las siguientes variables:

- u : Cantidad de usuarios.
- p : Cantidad de publicaciones que utilizan la etiqueta.

La sumatoria de la cantidad de favoritos de una etiqueta está definida por:

- v : La suma de todos los favoritos obtenidos por esa etiqueta.

La sumatoria de la cantidad de retuits de la etiqueta.

- r : La suma de todos los retuits obtenidos por esa etiqueta.

- c : El impacto de referencia se toma de una etiqueta definida por un grupo de expertos para darle medida a los rangos.

Para determinar el rango de gran impacto se tomó como base uno de los tuits más retuiteado en esta red social referente a MSR. Producido por el usuario Alyssa Milano el 15 de octubre del 2017 que se popularizó en Twitter, cuando escribió a sus seguidores: “Si has sido abusada o acosada sexualmente, escribe #MeToo en respuesta a este tweet”. La actriz recibió más de 60 000 respuestas y desató un movimiento internacional en el que las mujeres comenzaron a compartir sus experiencias utilizando el famoso hashtag. Por ello, como rango de gran impacto se establece desde 1.0 en adelante (sumando los favoritos y retuits); el rango medio, desde los 0.5 hasta 1 y el rango bajo, de 0 a 0.5.

Figura 1: Unos de los tuits más divulgado en las redes. (Elaboración propia)

Colector de tuits: Es la aplicación encargada de interactuar directamente con los servicios de Twitter para la recuperación de los tuits, de acuerdo con los criterios de los estudios de detección de roles que se encuentren activos.

Procesador de tuits: Se encarga del procesamiento asíncrono de los tuits que han sido recuperados por el recolector. Tiene entre sus funciones la extracción de los usuarios contenidos en el tuit y de las relaciones que se establecen entre estos. Así como, la persistencia del tuit y del modelo de relaciones obtenido. Este ejecuta el cálculo del impacto que ha de tener la etiqueta del tuit en cuestión por medio de un algoritmo matemático.

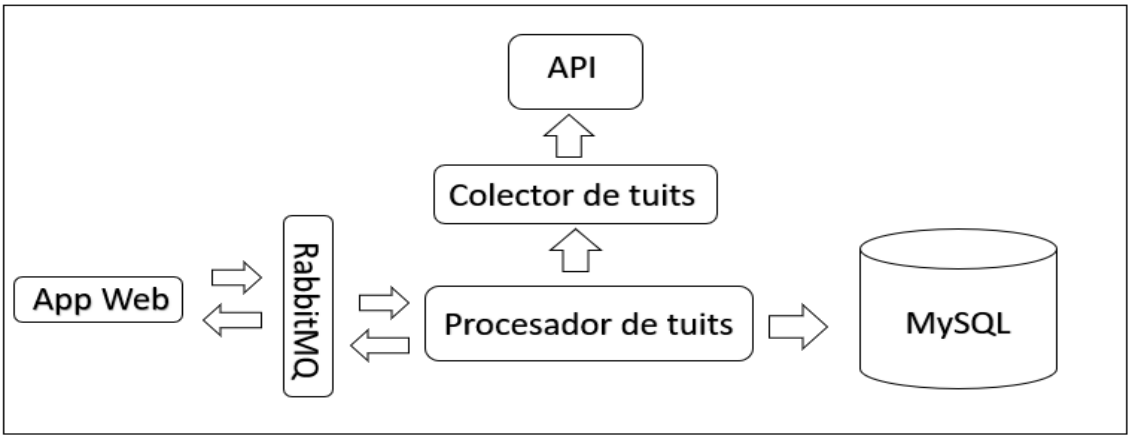


Figura 2. Vista general del sistema. (Elaboración propia)

Además de las aplicaciones anteriores, el sistema hace uso del sistema de gestión de bases de datos, MySQL. De esta manera se obtiene un bajo acoplamiento entre la aplicación encargada de procesar las métricas y el sistema gestor de bases de datos, a la vez que se obtiene un punto de acceso para la integración con otras aplicaciones externas al sistema.

2.2 Levantamiento de Requisitos.

Los requerimientos de software son las características y cualidades que el sistema debe tener. El proceso de captura de requisitos da inicio a la interacción con el cliente, trazando como meta fundamental satisfacer sus necesidades, las cuales son analizadas e investigadas, derivando en un estudio exhaustivo de los referentes teóricos-metodológicos. A partir de este punto, describen las condiciones que se necesitan para dar cumplimiento a los objetivos planteados. Una vez definidos e identificados los mismos, se tiene la visión general de lo que se requiere hacer en el sistema (Figueiras, 2014).

2.2.1 Requisitos funcionales.

Los requisitos funcionales se definen como las condiciones o capacidades que el sistema debe cumplir. Responden a ¿Qué debe hacer el sistema? (Pressman, 2009.).

Luego del estudio realizado, se identificaron los requisitos funcionales (RF):

Tabla 1. Requisitos funcionales. (Elaboración propia)

Requisitos	Descripción	Prioridad
RF1: Autenticar.	El sistema debe permitir al usuario registrarse en el sistema.	Media
RF2: Gestionar usuario de sistema.	El sistema debe permitir gestionar los usuarios de Twitter a emplear.	Media
RF3: Establecer conexión con Twitter.	El sistema debe permitir una conexión con Twitter.	Alta
RF4: Gestionar hashtag de Twitter.	El sistema debe permitir la gestión de hashtag.	Media
RF5: Gestionar tuits.	El sistema debe permitir la gestión de los tuits.	Alta
RF6: Recuperar tweet de forma asíncrona.	El sistema debe permitir la recuperación de tweets de forma asíncrona.	Alta
RF7: Calcular impacto.	El sistema debe permitir calcular el impacto de una etiqueta en un usuario.	Alta
RF8: Mostrar impacto.	El sistema debe mostrar el impacto de la etiqueta en el usuario.	Media
RF9: Generar reporte.	El sistema debe generar un reporte del impacto sobre la etiqueta.	Alta
RF10: Generar boletines por correo electrónico.	El sistema debe generar un boletín por correo electrónico donde muestre el informe del estudio.	Media

2.2.2 Requisitos no funcionales.

Describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los requisitos funcionales. Los requisitos no funcionales incluyen restricciones cuantitativas, como el

tiempo de respuesta o precisión, tipo de plataforma, lenguajes de programación o sistemas operativos, entre otros (Pressman, 2009.).

Interfaz

- RnF1. Interfaz sencilla, intuitiva y amigable para sus usuarios.
- RnF2. Implementar la ejecución de acciones de una manera rápida, minimizando los pasos a ejecutar en cada proceso.
- RnF3. La interfaz deberá garantizar la distinción visual entre los elementos.

Requerimientos de hardware

- RnF4. Para que el servidor realice todas funcionalidades especificadas se requiere:

- 4GB de memoria RAM como mínimo.
- Procesador de la familia INTEL o AMD Ryzen con una velocidad mínima de 2.6 GHz.

Software

- RnF5.Debe ser multiplataforma para ganar en compatibilidad.

Accesibilidad

- RnF6. Debe ser accesible desde los navegadores web Mozilla Firefox (versión 60 o superior) y Google Chrome (versión 33.0 o superior).

Integración

- RnF7. Modulo con bajo acoplamiento y alta cohesión que facilite la integración.

2.3 Descripción de historias de usuario.

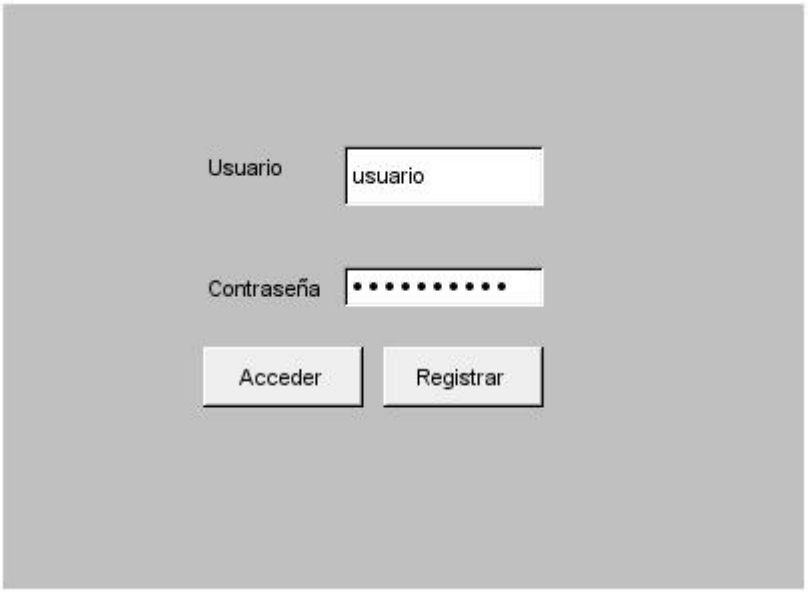
Las siguientes tablas presentan la descripción de los requisitos funcionales mediante las historias de usuario, según la metodología seleccionada.

Tabla 2: Historia de usuario: Autenticar. (Elaboración propia)

HISTORIA DE USUARIOS	
Número: RF_1	**Nombre del requisito:** Autenticar.
Programador: Adrian Carballo Guilarte	**Iteración Asignada:**1

Prioridad: Media	**Tiempo Estimado:** 1 semana
Riesgo en Desarrollo:	**Tiempo Real:** 4 días
Descripción: El sistema debe mostrar una interfaz de *login* para la autenticación del usuario de la aplicación.	
Observaciones:	

Prototipo elemental de interfaz gráfica de usuario:



2.4 Validación de requisitos.

La validación de los requisitos es el proceso de verificar que los requisitos realmente definen el sistema que el cliente realmente quiere. Es importante porque los errores en un documento de requisitos pueden conducir a costos de trabajos extensos, cuando estos problemas se descubren durante el desarrollo o después de que el sistema está en servicio (Sommerville, 2011).

Existen varias técnicas de validación de requerimientos, entre ellas:

- **Revisión de requerimientos**

Estas son un proceso manual en la que intervienen el cliente y el personal involucrado en el desarrollo del sistema, ésta puede ser formal o informal, y se efectúa con el fin de verificar que el documento de requerimientos no presente anomalías ni omisiones.

- **Construcción de prototipos**

Consiste en mostrar un modelo ejecutable del sistema a los usuarios finales y a los clientes, así éstos pueden experimentar con el modelo para ver si cumple con sus necesidades reales.

- **Generación de casos de prueba**

La generación de los casos de prueba consiste en probar los requisitos. Si una prueba es difícil o imposible de diseñar, normalmente significa que los requerimientos serán difíciles de implantar y deberían ser considerados nuevamente.

De las técnicas de validación de requisitos enunciadas anteriormente se aplicó la revisión técnica formal. Esta permitió verificar que el documento de requerimientos no presenta anomalías ni omisiones.

2.5 Patrón de arquitectura.

Se relacionan con el diseño a gran escala y de grano grueso, que se aplican típicamente durante las primeras iteraciones (la fase de elaboración) cuando se establecen las estructuras y conexiones más importantes (Pressman, 2009.).

Modelo Vista Controlador

“La arquitectura Modelo Vista Controlador surgió como patrón arquitectónico para el desarrollo de interfaces gráficas de usuario en entornos *Smalltalk*. Su concepto se basa en la necesidad de reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, mediante la división de la aplicación en tres partes fundamentales” (García, y otros, 2014):

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre el modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

"Estos tres elementos juntos, la lógica de acceso a la base de datos, la lógica de negocios, y la lógica de presentación comprenden un concepto, que a veces es llamado, el patrón de arquitectura de software Modelo-Vista-Controlador (MVC). En este patrón, el modelo hace referencia al acceso a la capa de datos, la vista se refiere a la parte del sistema que selecciona qué mostrar y cómo mostrarlo, y el controlador implica la parte del sistema que decide qué vista usar, dependiendo de la entrada del usuario, accediendo al modelo si es necesario" (García, y otros, 2014).

En el desarrollo de la solución se utilizará esta arquitectura, para una mejor comprensión se ejemplifica esta arquitectura en la siguiente imagen:

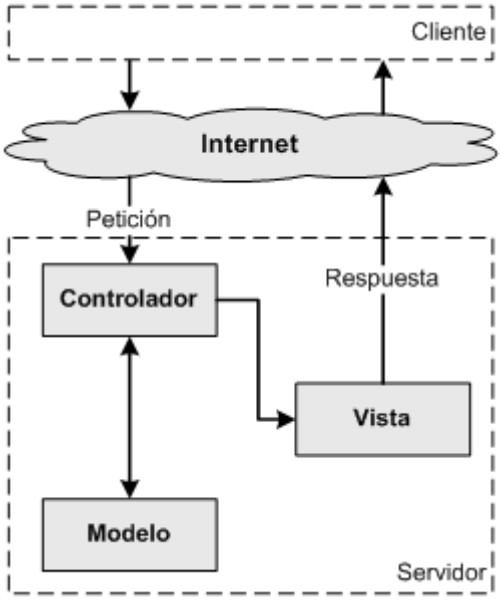


Figura 3. Funcionamiento de arquitectura MVC. (Elaboración propia)

Para la aplicación que nos ocupa, la separación seria:

- Capa de datos (Modelo): la BD de la aplicación y los modelos a partir de los cuales se mapea esta.
- Capa de interfaces graficas (Vista): son las vistas que devuelven los controladores, concretamente archivos HTML.
- Capa de lógica (Controlador): Los controladores y servicios que hacen de enlace entre la capa de datos y las interfaces.

Llamada a procedimiento remoto

Capítulo 2: Propuesta de solución.

La llamada a procedimiento remoto (RPC), como su nombre lo indica, es un mecanismo que permite al programador invocar a una funcionalidad de forma remota, ya sea desde una máquina diferente u otro proceso al realizar una llamada a un procedimiento local. Este enfoque permite modelar una comunicación como una función de llamada directa mientras oculta los detalles sobre la red real. Eventualmente es una abstracción, un concepto de interacción dentro de la red.

RPC puede utilizar diferentes protocolos para su funcionamiento, tales como, HTTP y HTTPS. Por lo tanto, cuando se hace uso del RPC, el identificador del procedimiento y los parámetros se serializan en el mensaje de solicitud que se envía al proceso remoto que atiende la llamada. En el proceso remoto, el mensaje se vuelve a deserializar, para extraer la información asociada, que es empleada en la ejecución y los resultados obtenidos son retornados de forma similar, mediante colas de mensajería (Slutsker, 2017).

De forma general el funcionamiento de RPC sigue los siguientes pasos:

1. La máquina cliente realiza una llamada RPC a una máquina servidor que se registró para atender esta llamada específica,
2. Parámetros de procedimiento transferidos a través de la red a la máquina del servidor correspondiente
3. Servidor máquina ejecuta el procedimiento.
4. Cuando finaliza el procedimiento y se producen los resultados, se transfieren al entorno de llamada.
5. La ejecución en la máquina cliente se reanuda como si regresara de una llamada de procedimiento regular (local).

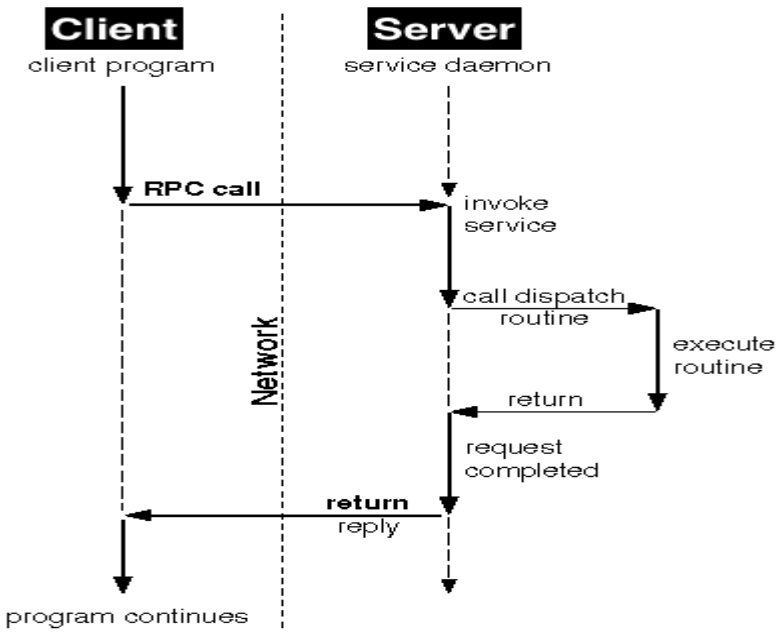


Figura 4: Funcionamiento de un RPC. (Slutsker, 2017)

2.6 Patrones de diseño.

Un patrón de diseño constituye un esquema para refinar subsistemas o componentes. Es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades, además de que ayuda a construir clases y a estructurar sistemas de clases. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces (Gamma & otros, 2002). En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otro contexto; en teoría, indica la manera de utilizarlo en circunstancias diversas. Los patrones de diseño son soluciones simples y elegantes basadas en la experiencia a problemas específicos y comunes del diseño orientado a objetos. Con el uso de patrones los diseños serán mucho más flexibles, modulares y reutilizables (Larman, 2004).

Resumiendo, un patrón de diseño es (Gamma & otros, 2002)

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un proyecto o estructura de implementación que logra una finalidad determinada.
- Un lenguaje de programación de alto nivel.
- Una manera más práctica de describir ciertos aspectos de la organización de un programa.
- Conexiones entre componentes de programas.
- La forma de un diagrama de objeto o de un modelo de objeto

Su uso, brinda indudables ventajas en muchos sentidos (Gamma & otros, 2002):

- Proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Están basados en la recopilación del conocimiento de los expertos en desarrollo de software.
- Es una experiencia real, probada y que funciona.
- Facilitan la localización de los objetos que formarán el sistema, la determinación de la granularidad adecuada, el aprendizaje y la comunicación entre programadores.
- Especifican interfaces para las clases e implementaciones al menos parciales.

Para el desarrollo, se siguieron los patrones de diseño conocidos como Patrones Generales de *Software* para Asignar Responsabilidades (*General Responsibility Assignment Software Patterns*, GRASP por sus siglas en inglés).

2.7 Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2004). Existen diversos patrones GRASP utilizados para guiar el proceso de desarrollo y cumplir con los requerimientos de una manera limpia y simplificada.
El **Bajo Acoplamiento** estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento y soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades (Larman, 2004).

El acoplamiento tal vez no sea tan importante, si no se busca la reutilización, para dado que el sistema está previsto para su posterior integración con las otras esferas por confeccionar, es un aspecto a tener en cuenta durante el desarrollo.

Como el patrón Bajo Acoplamiento, también **Alta Cohesión** es un principio que debemos tener presente en todas las decisiones de diseño: es la meta principal que hade buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño.

La cohesión es una medida de cuán relacionadas están las responsabilidades de una clase. Una alta cohesión permite a las clases que están muy relacionadas no realizar un enorme trabajo. Las clases que presentan baja cohesión son difíciles de comprender, reutilizar y conservar (Larman, 2004).

Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen.

Además, se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto contribuye a un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento, alentando con ello definiciones de clases más sencillas y cohesivas que son más fáciles de comprender y de mantener (Larman, 2004).

Siguiendo lo planteado por los patrones experto, bajo acoplamiento, y alta cohesión, las clases implementadas se centran en funcionalidades específicas, para las cuales tienen toda la información necesaria. Por ejemplo, las clases entidades, solo se encargan de contener la información, delegando la interacción con la base de datos a la clase manejadora, provista por Doctrine. Otro ejemplo lo encontramos en la clase RPCServer, la cual se encarga de la integración con el RabbitMQ para el procesamiento de los tuits.

Otra forma de garantizar el bajo acoplamiento es mediante la creación de servicios y formularios en Symfony, de esta manera se libera a las clases controladoras de la responsabilidad de la lógica del negocio y otras tareas complejas. En el código implementado esto se puede apreciar en las clases AnalisisForm y PrensaForm.

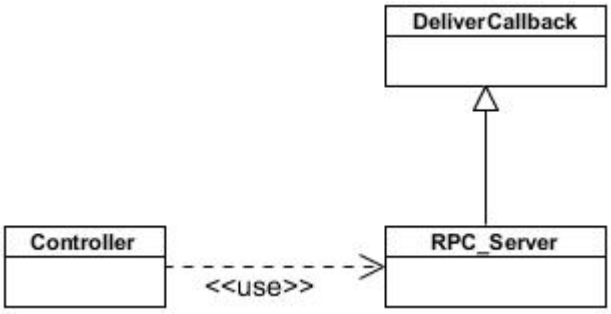


Figura 4. Ejemplo patrón experto. (Elaboración propia)

El patrón **Creador** guía la asignación de responsabilidades relacionadas con la creación de objetos. Una clase B tiene la responsabilidad de crear un objeto de una clase A cuando la clase está contenida en la clase B, la clase B en una agregación o composición de la clase A, la clase B almacena a la clase A, la clase B inicializa los datos de la clase A o la clase B usa la clase A. Entre sus beneficios está que brinda soporte al bajo acoplamiento (Larman, 2004).

Este patrón se aplica en las clases controladoras, para la instanciación de los formularios y servicios de Symfony. En los componentes encargados del procesamiento asíncrono de los Tweets este patrón se manifiesta en las clases RPCServer y TweetProcessor.

Al hacer uso del marco de trabajo para PHP, se emplean buenas prácticas entre las cuales se encuentran los patrones *Singlenton* y No Hables con Extraños. Estos patrones se manifiestan en la relación existente entre las clases controladoras y los servicios.

2.8 Diagrama de clases del diseño.

Según Jacobson, "Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases en sí, muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los mismos se utilizan para reflejar la vista de diseño estática de un sistema. Son la base para los posteriores diagramas de componentes y los de despliegue. Su importancia radica en que permitirá construir sistemas ejecutables aplicando ingeniería directa e inversa" (Jacobson, 2004).

El propósito principal es adquirir una comprensión con profundidad de los aspectos relacionados con los requisitos funcionales, no funcionales y restricciones asociadas a la programación, sistemas operativos y tecnologías de interfaz de usuario. UML posee una extensión para el modelado de aplicaciones *web*, esa extensión es usada para el diseño de las clases. Los estereotipos que usa esta extensión son:

- ✓ <<*Client Page*>> Una instancia de Página Cliente es una página *web*, con formato HTML. Cada página cliente solo puede ser construida por una página servidor.
- ✓ <<*Form*>> Grupo de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario.
- ✓ <<*Server page*>> Representa la página web que contiene código que se ejecuta en el servidor.
- ✓ *Link*: Representa un apuntador desde una "client page" hacia una "*client page*" o "*serverpage*". Corresponde directamente con una etiqueta <a> (ancla) de HTML.
- ✓ *Submit*: Esta relación siempre se da entre una "form" y una *"server page",* por supuesto, la "*server page*" procesa los datos que la "*form*" le envía (*submits*).
- ✓ *Build*: Sirve para identificar cuales "*server page*" son responsables de la creación de una "*client page*".
- ✓ *Redirect*: Esta es también una relación unidireccional que indica que una página *Web* redirige hacia otra. Las *client page* las cuales son las páginas que se muestran en la interfaz del sistema y son las que interactúan con el usuario además construyen un formulario en el cual el usuario llena los datos

necesarios para adicionar una nueva red, modificar una red seleccionada o eliminar la red que se seleccione.

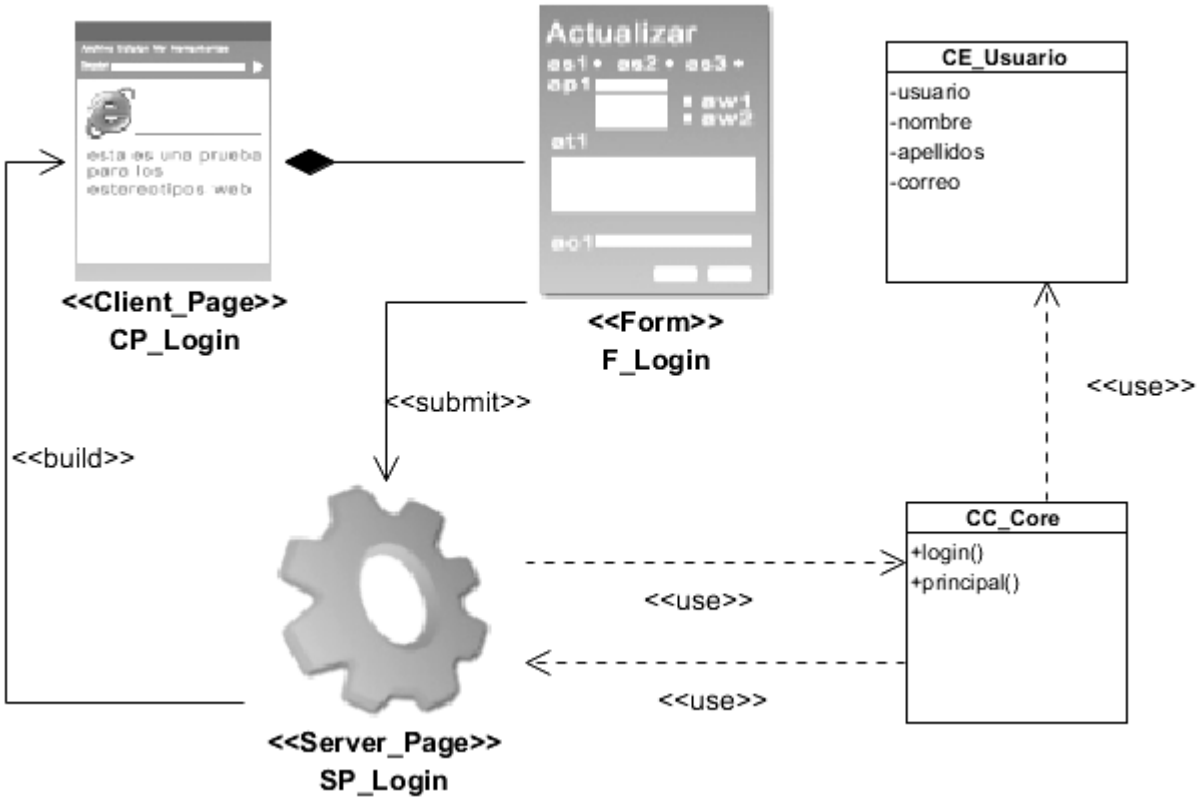


Figura 5. Diagrama de clases de diseño autenticar. (Elaboración propia)

2.9 Diagrama de secuencia.

Para Richard Luciano García, “Es el diagrama que muestra las interacciones entre los objetos organizados en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados. Dentro del conjunto de mensajes representados dispuestos en una secuencia temporal, cada rol en la secuencia se muestra como una línea de vida, es decir, una línea vertical que representa el rol durante cierto plazo de tiempo, con la interacción completa. Los mensajes se muestran como flechas entre líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción. Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso de uso” (García, 2014).

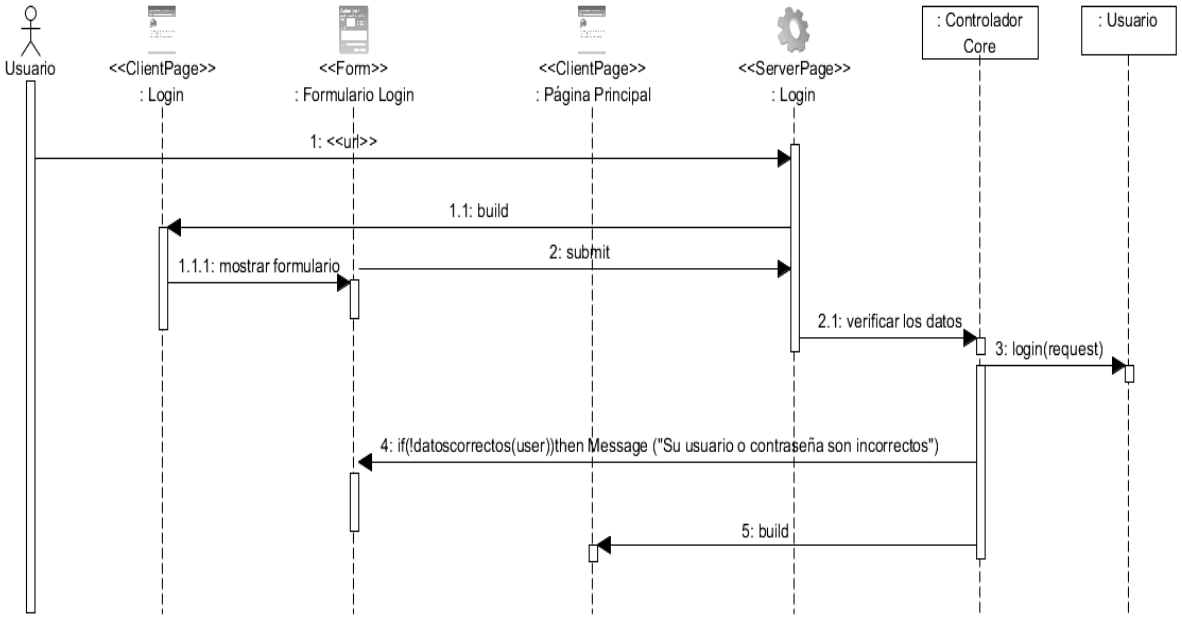


Figura 6. Diagrama de secuencia autenticar usuario. (Elaboración propia)

Este diagrama de secuencia autenticar usuario se dispone a autenticarse en la web, para esto hace la petición a la *serverpage* mediante una url, seguidamente esta muestra en la *clientpage* con su formulario, donde el usuario escribe su usuario y contraseña, estos los envía a la página servidora, la cual verifica los datos con el controlador y efectúa la autenticación solo si el usuario no se equivocó al poner sus credenciales, de esta forma al ser los datos correctos se construye la página principal.

2.10 Modelo de datos.

Un modelo de datos: “Es un conjunto de conceptos que permite describir los datos, las relaciones que existe entre ellos, la semántica y las restricciones de consistencia” (González, 1988). Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. El modelo de datos físico plantea cómo se maneja la información en la base de datos en el cual existen varias tablas relacionadas con los órganos de prensa, los usuarios y el impacto que guardan toda la información necesaria para una posterior gestión de ella.

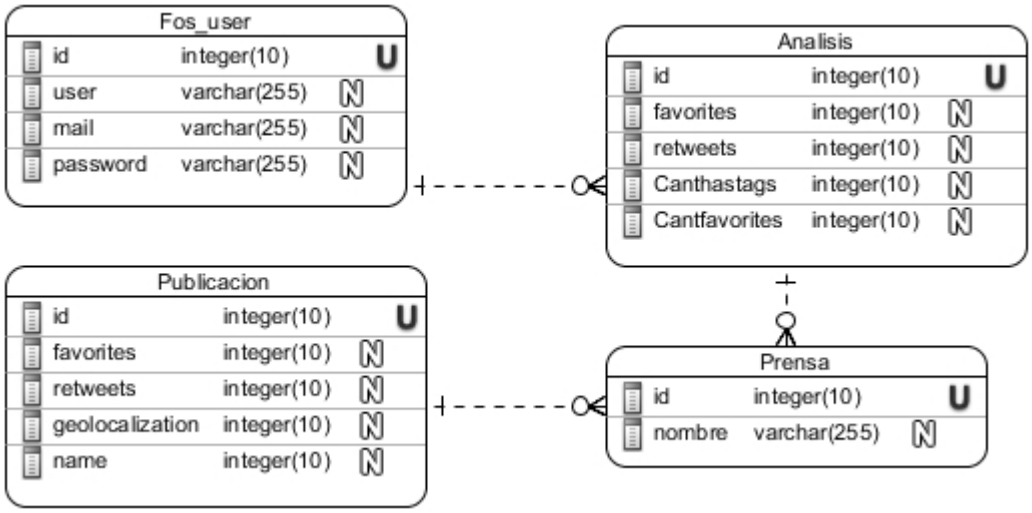


Figura 7. Modelo de datos. (Elaboración propia)

2.11 Conclusiones parciales.

La realización de la propuesta de solución permitió definir los servicios web a brindar por la aplicación, la identificación de los requerimientos funcionales y la modelación de los diagramas de clases del diseño. Los diagramas de secuencia permitieron tener una mejor visión de la interacción entre el usuario y el sistema. La realización del modelo físico de datos proporcionó un buen entendimiento de como almacenar y gestionar la información proporcionada para el análisis del impacto de los MSR. Por otra parte, la definición de la arquitectura Modelo-Vista-Controlador dejó sentadas las bases para la confección e implementación del sistema.

Capítulo 3: Implementación y pruebas.

En este capítulo se elabora la propuesta de solución, la página web que da acceso a la búsqueda e inclusión de las etiquetas de los MS. Se muestra el tuit de la etiqueta seleccionada en tiempo real y su impacto en la red. Se realizan las pruebas pertinentes al software para verificar su buen funcionamiento.

3.1 Estilos de código.

Los estilos son normas usadas para escribir código y que incluye una gran gama de aspectos dentro del proceso de codificación. Un buen estilo de programación debe aportar a la eficiencia del proceso de desarrollo, logrando que los programas sean robustos y comprensibles (Figueiras, 2014) (Muñoz Caro, Niño, y Vizcaíno, 2002).

Uno de los aspectos a tener en cuenta para lograr mayor calidad en los productos de software es codificar de una manera clara y legible.

El mejor estilo es el que más aporte a la eficiencia del proceso de desarrollo y el que logre los programas más robustos y fáciles de usar. Todo lo que vaya en contra de esto es incorrecto, sobre todo porque para cumplir con este principio no se entra en contradicción con la propia lógica del problema (Figueiras, 2014) (Muñoz Caro, Niño, & Vizcaíno, 2002)

Independientemente de que las plataformas de desarrollo influyan en cierto modo a la formación de estilos de programación, es responsabilidad de cada equipo determinar el suyo. A la hora de confeccionar un estilo de código, existen aspectos fundamentales que se deben tener en cuenta; notación, comentarios, identación, espacios y líneas en blanco, longitud máxima de las líneas de caracteres, declaración de variables, declaración de constantes y parámetros en los métodos (Figueiras, 2014) (Muñoz Caro, Niño, y Vizcaíno, 2002).

Para el desarrollo, se usa el definido en la guía de estilo del código Java y los que propone el lenguaje PHP para el marco de trabajo *Symfony*. (Hommel & Otros, 1999) (PHP-Fig, 2018).

Tabla 3. Estándares de codificación. (Elaboración propia)

Tipo de Estándar	**Descripción**

Organización del código.	1. La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración. 2. Las aperturas de llaves en las estructuras de control se colocan en la misma línea 3. El cierre de las llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código. 4. La identación se realiza con tabulación y no con 4 espacios en blanco.
Líneas.	5. El tamaño máximo de una línea no excederá de ochenta (80) caracteres. 6. Se deberá agregar una línea en blanco delante de la sentencia *return*. 7. Se puede agregar líneas en blanco para mejorar la legibilidad e indicar bloques de códigos relacionados. 8. Debe haber una declaración por línea.
Codificación.	9. Los archivos PHP deben utilizar solo las etiquetas de apertura <*?php* omitiendo las etiquetas de cierre ?>. 10. Utilizar la codificación UTF-8.
Espacios en blanco en expresiones y sentencias.	11. Se debe utilizar un espacio en blanco después de la palabra clave en las estructuras de control. 12. No deben existir espacios en la apertura y cierre de paréntesis. 13. Debe usarse un espacio en blanco entre los operadores lógicos, aritméticos, de comparación y asignación.

Convenciones de Nombramientos.	14. Se debe utilizar el estilo de escritura *camelCase* para la declaración de variables y no guiones bajos para separar palabras. 15. Utilizar letras mayúsculas sostenida para declarar contantes y separar las palabras con guiones bajos. 16. Se debe usar el estilo de escritura *StudlyCaps* para los nombres de clases y funciones.

3.2 Diagrama de Componente.

"Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. El diagrama de componentes muestra la vista física del software en términos de componentes ejecutables y librerías de clases con sus relaciones y sus dependencias" (Jacobson, 2004). Estos representan los elementos físicos del sistema y las relaciones entre ellos. Los componentes representan todos los elementos de *software* que intervienen en la fabricación de aplicaciones informáticas.

Un diagrama de componentes representa las dependencias entre componentes de *software*, incluyendo componentes de código fuente, códigos binarios y ejecutables, hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación.

Para consultar los demás diagramas de componentes consultar los Anexos.

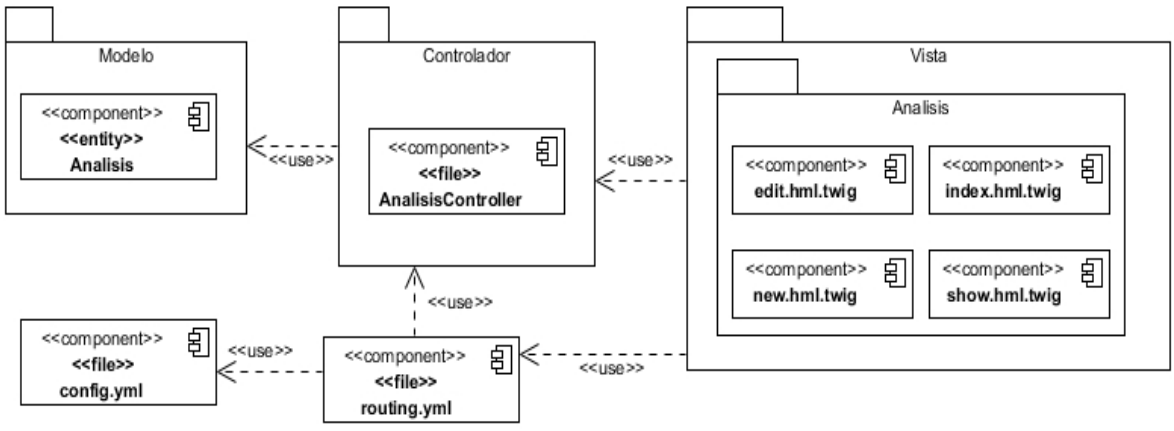


Figura 8. Diagrama de componentes Análisis. (Elaboración propia)

3.3 Modelo de despliegue.

El diagrama de despliegue permite indicar la situación física de los componentes lógicos desarrollados. Este modelo es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre dichos elementos. Cada Hardware es representado como un nodo; un nodo es un elemento donde se ejecutan los componentes, entre ellos existen relaciones que son representados como medios de comunicación tales como HTTPS o TCP/IP.

En el modelo de despliegue del sistema (ver Diagrama 3.2), se puede observar de forma gráfica los elementos que intervienen (Cliente, Servidor Web, Servidor de Base de Datos), el Cliente realiza la petición al Servidor Web mediante el protocolo HTTPS, este solicita los datos al Servidor de Base de Datos mediante el protocolo TCP/IP, luego los procesa y muestra al Cliente.

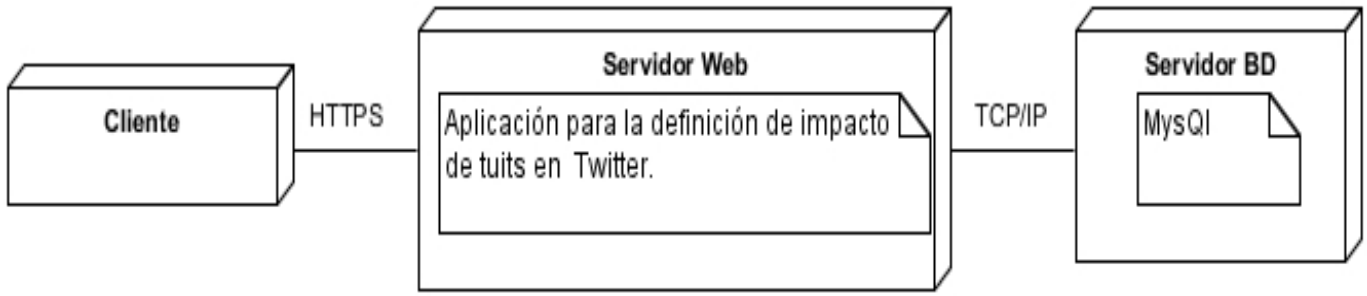


Figura 9. Diagrama de despliegue. (Elaboración propia)

3.4 Comprobación de la hipótesis.

Para la comprobación de la hipótesis se hizo uso de la técnica de Iadov, y la entrevista. La técnica de Iadov se empleó para evaluar la aceptación de la aplicación desarrollada y la entrevista para indagar sobre el impacto en la capacidad de análisis de los movimientos sociales por parte del departamento DOWAI. La selección de los participantes se realizó mediante la determinación del coeficiente de competencia. Para esto se siguió la metodología para la elaboración de pronósticos científicos – técnicos, aprobada por el Comité Estatal para la Ciencia y la Técnica de Rusia en 1971, que plantea la siguiente fórmula:

$$K = \frac{1}{2}(K_c + K_a)$$

Ecuación 4. Coeficiente de competencia.

En la que K representa el coeficiente de competencia, K_c el coeficiente de conocimiento, el cual se calcula mediante la autovaloración del propio experto en una escala del 0 al 10 y multiplicado por 0.1. El coeficiente de argumentación es representado por K_a, para su cálculo se hace uso de la suma de los puntos alcanzados a partir de la tabla número cuatro, por cada experto (Sariol, y otros, 2018) (Mederos, 2017) (Flores, y otros, 2017).

Tabla 4. Tabla modelo.

Fuentes de argumentación	Alto	Medio	Bajo
Investigaciones teóricas y/o experimentales relacionadas con el tema, realizadas por usted.	0,3	0,2	0.1
La experiencia obtenida en su actividad profesional (docencia de pregrado y posgrado recibida y/o impartida).	0,5	0,4	0,2
Análisis de la literatura especializada y publicaciones de autores nacionales.	0,05	0,05	0,05
Análisis de la literatura especializada y publicaciones de autores extranjeros.	0,05	0,05	0,05
Su propio conocimiento del estado actual de la problemática en el país y en el extranjero.	0,05	0,05	0,05
Su intuición.	0,05	0,05	0,05
Total	1	0,8	0,5

Para la interpretación de los resultados se utilizan tres escalas, como se muestra en la siguiente tabla.

Tabla 5. Escalas de coeficiente de argumentación.

K	Coeficiente de competencia
0 a 0,5	Bajo
0,5 a 0,8	Medio
0,8 a 1	Alto

La muestra seleccionada estaba compuesta por 4 expertos y los cuales obtuvieron coeficientes mostrados en la tabla:

Tabla 6. Resultados del procesamiento de la autovaloración de expertos.

Experto	Kc	Ka	K	Valoración
1	0.8	0.95	0.88	Alto
2	0.9	1	0.95	Alto
3	0.9	1	0.95	Alto
4	0.8	0.95	0.88	Alto

3.4.1 Entrevista.

A los expertos seleccionados se les realizó una entrevista cerrada, siguiendo un cuestionario en el cual se indagaba su criterio sobre la aplicación desarrollada, el resultado ofrecido por esta y el impacto que tiene en la capacidad de análisis del desempeño en Internet de los movimientos sociales por parte del departamento.

Con antelación a la entrevista los expertos interactuaron con el sistema, realizando análisis para varias etiquetas que constituían tendencias en ese momento. Durante las entrevistas se valoró de forma positiva el resultado obtenido, y se manifestó que mediante esta aplicación es posible incrementar la capacidad de análisis de los movimientos sociales en el sitio de Twitter.

3.5 Pruebas de software.

"Las pruebas del *software* son un elemento fundamental para la garantía de calidad del sistema. Estas pruebas representan una revisión final de las especificaciones del diseño y de la codificación, es decir, las pruebas verifican que el *software* funcione como se diseñó y que los requerimientos son satisfechos, además de brindar soporte para encontrar y documentar defectos del sistema" (Pressman, 2009.). El objetivo de la prueba de *software* es descubrir errores.

Para la realización de las pruebas al sistema se decidió utilizar las Pruebas de Rendimiento por la necesidad de probar el rendimiento que tendrá el sistema ante un número de conexiones concurrentes de usuarios o peticiones de los mismos. Además, se utilizarán las pruebas de seguridad para comprobar que se cumplan

los niveles de acceso según los roles establecidos y restringir el acceso a la información. Se emplearán Pruebas unitarias para verificar que se cumplan los requisitos del cliente.

3.5.1 Pruebas unitarias.

Comienzan con la prueba de cada módulo. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de estos módulos funcione correctamente por separado. El objetivo que persiguen es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el fragmento de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas: fomentan el cambio, simplifican la integración, documentan el código, separan la interfaz del código y hacen que los errores estén más acotados y sean fáciles de localizar.

A lo largo del desarrollo del componente se realizaron pruebas unitarias para ir comprobando el funcionamiento del software. Su ejecución, por parte del desarrollador, permite aprovechar las ventajas de compilación, paso a paso, que brinda el entorno de desarrollo.

Las pruebas se evaluarán en base al resultado que arrojen, siguiendo la escala que a continuación se presenta:

- **Prueba satisfactoria**: cuando el resultado de la prueba es exactamente el esperado por el equipo de desarrollo.

- **Prueba parcialmente satisfactoria**: cuando el resultado no es completamente el esperado por el equipo de desarrollo pues muestra resultados erróneos o fuera de contexto.

- **Prueba insatisfactoria:** cuando el resultado de la prueba realizada genera un error de codificación en la aplicación. El proceso se realizó en tres iteraciones (ver Figura 10) obteniendo resultados no satisfactorios en las dos primeras. Para la tercera y última iteración, los resultados fueron satisfactorios en un 100% por lo que se considera cumplido el proceso de validación para las pruebas unitarias.

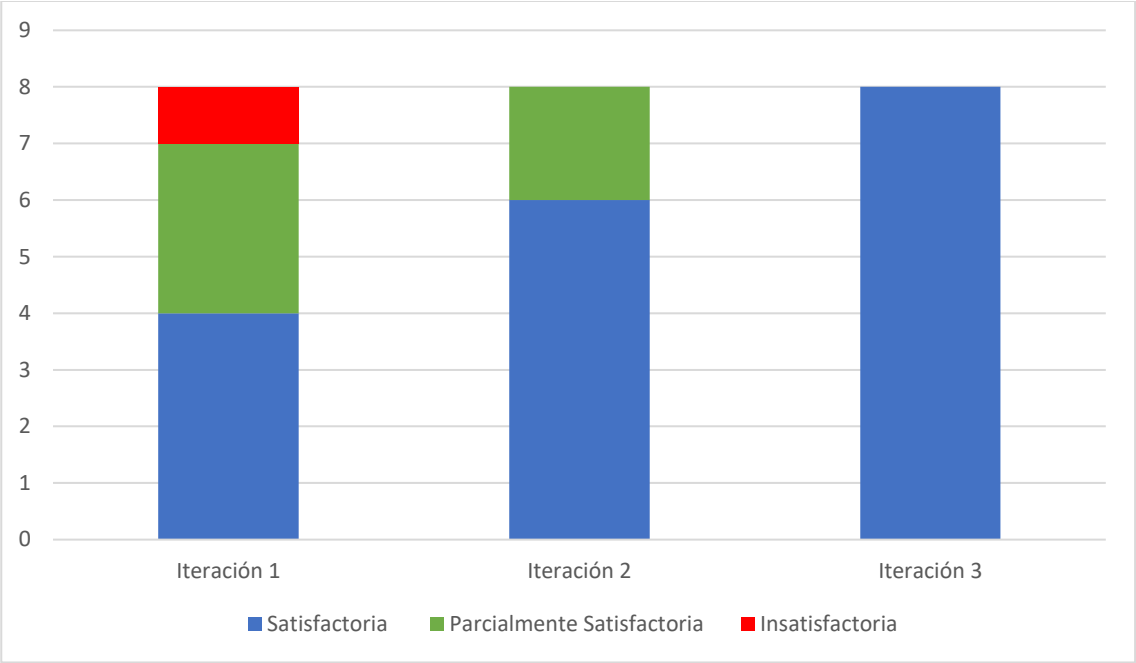


Figura 10. Resultado de las pruebas de aceptación. (Elaboración propia)

3.5.2 Pruebas de seguridad.

"Permite asegurar que el sistema modificado o nuevo, incluya controles de acceso apropiados y no contenga ningún agujero de seguridad que pudiera comprometer otros sistemas" (García, y otros, 2014) . Estas pruebas miden la confidencialidad, integridad y disponibilidad de la información, con el objetivo de identificar amenazas, riesgos y la probabilidad de enfrentarse a ataques informáticos que descubran y exploten las vulnerabilidades de la aplicación (QUALITY, 2016). Están enfocadas a garantizar que el acceso a la información por parte de los usuarios sea según el rol que desempeñan y a las funcionalidades específicas para ese rol. Dependiendo del rol que esté asignado al usuario que se autentique será capaz de ejecutar determinadas funciones según su nivel de acceso. Este tipo de pruebas permite comprobar los niveles de seguridad lógica del sistema. Las pruebas de seguridad aseguran que los usuarios tengan acceso solo a la información que están autorizados a acceder, garantizan que no existan brechas de seguridad en el sistema pues solo el personal autorizado puede acceder a la aplicación.

En el grupo de Seguridad del Departamento de CIDI se han establecido niveles para este tipo de pruebas. La evaluación de la seguridad de las aplicaciones en un primer nivel (nivel 1) se definió 15 indicadores agrupados en 4 tipos de pruebas.

Resultados Pruebas de Seguridad nivel 1.

1. **Pruebas de Autorización:** Ningún usuario estándar pudo modificar sus privilegios ni los de otro usuario en la aplicación.
2. **Pruebas de Gestión de Sesiones:** No se pudo acceder a la aplicación copiando la URL después de estar autenticado, cerrar el navegador y volver a abrirlo. Al cerrar la sesión de un usuario y dar clic en el botón del navegador "Atrás" la aplicación no entró a la sesión autenticada.
3. **Validación de Datos:** Se enmascararon los datos confiables cuando se visualizan en la aplicación. Solamente se permiten contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres, como mínimo, de longitud. El sistema no mostró mensajes indebidos al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples (´´), signos de *ampersand* (&), o signos*: + - /.*
4. **Comprobación del Sistema de Autenticación:** Los mensajes de error para distintas combinaciones de autenticación muestrearon la misma información. Los tiempos de respuestas usuario correcto - contraseña incorrecta y usuario incorrecto – contraseña incorrecta fueron los mismos. El sistema protegió el envío de los datos mediante protocolo.
5. **Comprobación del Sistema de Autenticación:** Los mensajes de error para distintas combinaciones de autenticación mostraron la misma información. Los tiempos de respuestas usuario correcto - contraseña incorrecta y usuario incorrecto – contraseña incorrecta fueron los mismos. El sistema protegió el envío de los datos mediante protocolo seguro (*HTTPS*). Se bloqueó la sesión del usuario después de 1 hora de inactividad, pero no se bloqueó la cuenta del usuario después de un número de intentos de autenticación fallidos. El campo usuario de la autenticación al sistema tiene el auto completamiento desactivado (no guarda los usuarios que se autentican). El sistema usa un certificado.

Resultados de las pruebas de seguridad nivel 2.

Para valorar la seguridad en el nivel número dos se utilizó la herramienta *Acunetix Web Vulnerability Scanner 8*, que se emplea para detectar vulnerabilidades de seguridad en el sistema. *Acunetix Web Vulnerability Scanner* es una herramienta capaz de escanear sitios *web* en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página una vez publicada en Internet. Esta aplicación ejecuta una serie de pruebas, configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor, detecta técnicas de *hacking* como pueden ser *SQL injection*, *Cross Site Scripting*, *Passwords* débiles, etc. Una vez concluido este proceso, la versión genera una serie de informes, especificando los fallos detectados en el análisis de la página *web*. Estos informes se mostrarán en la pantalla en forma de gráfica para una mejor perfección, o bien podrán visualizarse los resultados más exhaustivamente con la descripción de la vulnerabilidad.

Durante la primera iteración, la herramienta realiza veinte mil (20 000) ataques informáticos y se generan sesenta y cinco (65) alertas de vulnerabilidad, divididas en cincuenta (50) de nivel medio y quince (15) de carácter informacional. Las alertas identificadas fueron corregidas en la primera iteración, para una segunda iteración, se igualaron la cantidad de ataques a la primera iteración y no se generan nuevas alertas.

Las alertas de nivel medio estuvieron relacionadas con cuarenta y cuatro (44) mensajes de error provocados por el modo de ejecución *DEV para* el desarrollo de aplicaciones en el marco de trabajo Symfony, de los que se puede obtener información sensible de los registros *DEBUG* del módulo. Se identificó un (1) formulario *HTML* vulnerable a la falsificación de petición en sitios cruzados (CSFR, por sus siglas en inglés). El formulario fue comprobado de forma manual como recomienda la herramienta *Acunetix* y se identificó que no pueden ser utilizados por el atacante para alterar el sistema ni obtener información, se denomina estas vulnerabilidades como falsos positivos.

3.5.3 Pruebas de rendimiento.

Las pruebas de rendimiento están enfocadas en evaluar el desempeño que tiene determinado sistema o componente. Estas pruebas generalmente se realizan mediante herramientas automatizadas que generan una cantidad de usuarios, carga y volumen de información, monitorizando también el rendimiento del *hardware*.

3.5.4.Pruebas de carga.

Dentro de las pruebas de rendimiento, es la más sencilla. Consiste en observar el comportamiento del sistema bajo una cantidad de peticiones. La carga puede ser el número de usuarios de manera concurrente usando la aplicación y que durante el tiempo que dura la carga realizan un número de peticiones. Si se monitorizan durante esta prueba el servidor de aplicaciones y la base de datos entonces puede mostrar el cuello de botella de la aplicación.

3.5.5 Pruebas de estrés.

Estas pruebas se utilizan generalmente para colapsar la aplicación. Se va duplicando el número de usuarios que se agregan a la aplicación y se ejecutan pruebas de carga hasta que el sistema falla. Estas pruebas se realizan para determinar la solidez en los momentos de carga extrema del sistema y constituye una ayuda para los administradores en función de conocer si la aplicación rendirá en el caso de que la carga real supere a la carga esperada.

3.6 Herramienta utilizada para la realización de las pruebas de carga y estrés.

La realización de estas pruebas se hizo de manera automatizada mediante el *software JMeter* en su versión 2.8.4 la cual es desarrollada en *Java* y permite la realización de pruebas de rendimiento en aplicaciones web. JMeter es un *software* que se encarga de realizar pruebas de carga y estrés de código abierto. En sus inicios fue desarrollado para probar aplicaciones *web*, y se ha expandido a otras funciones de prueba. Se utiliza para probar tanto recursos estáticos como dinámicos y se puede utilizar para simular una carga pesada en un servidor de red, probar su resistencia o analizar el rendimiento ante diferentes tipos de cargas.

Entorno de las pruebas de carga y estrés.

***Hardware* de prueba (PC cliente)**:

- Tipo de procesador: *Intel (R)* Core (TM) i3-2100 CPU @ 3.10Hz
- Memoria RAM: 4,00 GB Tipo de Red: Ethernet 10/100Mbps

***Software* instalado en ambas PC:**

Tipo de servidor web: *Apache*

Memoria máxima: 2048 MB Máximo de hilos concurrentes: 150

Plataforma: SO Ubuntu 18.04

Servidor de BD: PostgreSQL v9.1

Resultados de las pruebas de carga y estrés.

Las pruebas de rendimiento fueron ejecutadas con las mínimas prestaciones que se recomiendan, arrojaron resultados que ofrecen un contacto directo con el comportamiento del sistema en situaciones complicadas de funcionamiento. Para llevar a cabo esta ejecución se simuló un total de 150 usuarios conectados concurrentemente, con un período entre una petición y otra de 1 segundo. Bajo estas condiciones el sistema dio los siguientes resultados:

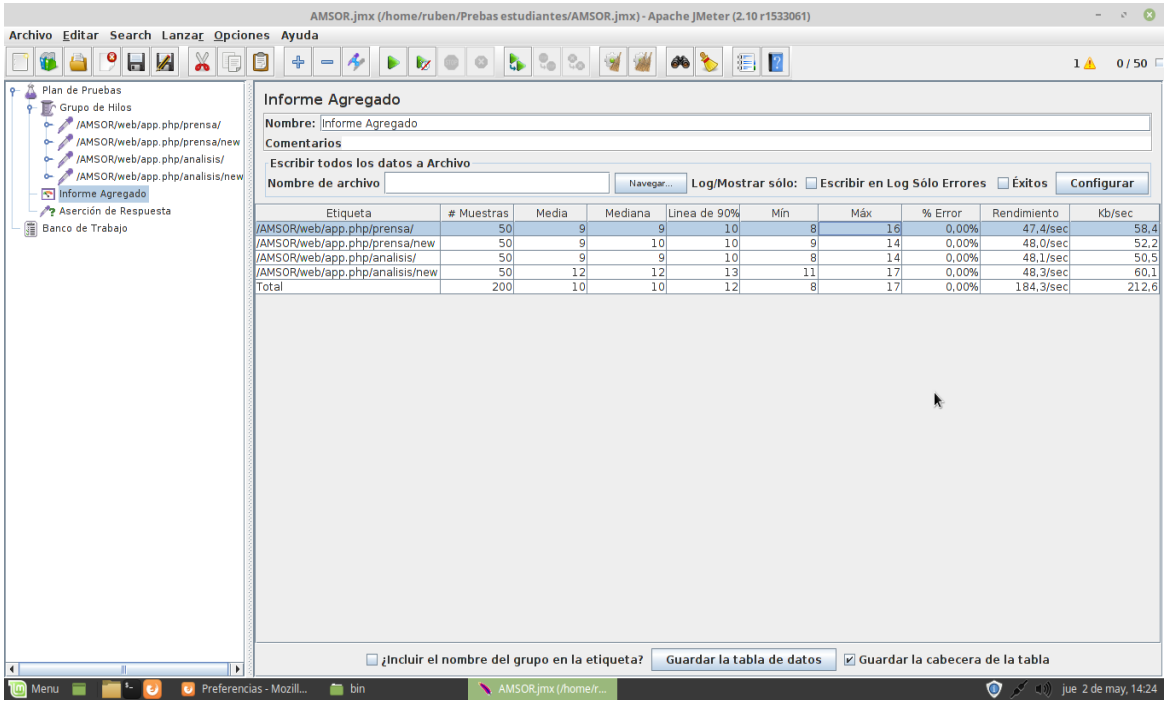


Figura 11. Resultados de prueba carga y estrés. (Elaboración propia)

Interpretación de resultados de las pruebas de rendimiento

- **Etiqueta:** El nombre de la muestra (conjunto de muestras)
- **# Muestras:** El número de muestras para cada *URL*.
- **Media:** El tiempo medio transcurrido para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido para las muestras de la *URL* dada.

- **Max:** El máximo tiempo transcurrido para las muestras de la *URL* dada. **%**
- **Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/sec:** Rendimiento medido en Kilobytes por segundo.
- **Media de Bytes:** Tamaño medio de la respuesta de la muestra medido en bytes.

Teniendo en cuenta que se realizaron las pruebas con un total de 200 usuarios concurrentes, con un período de peticiones enviadas al servidor, por los usuarios, de 1segundo, equivalente a 1 petición por cada 0.0699 segundos aproximadamente. Se puede observar que las pruebas se han realizado con un margen de error igual a 0.00% sobre un total de 16500 peticiones realizadas para un rendimiento aproximado de 184.3 peticiones por segundo. Atendiendo a la cantidad de peticiones por segundo que se enviaron, la velocidad de respuesta de las peticiones enviadas y las prestaciones del *hardware* donde se realizaron las pruebas, los resultados alcanzados son considerablemente buenos ya que la pagina por sí sola no tiene mucha carga porque la mayor parte del trabajo se realiza por parte de la aplicación en java.

3.7 Conclusiones parciales.

La representación y modelado de cada uno de los elementos del diagrama de componentes permitió una mejor compresión y gestión del mismo. Además, la utilización de un estilo de código permitió trabajar con una mayor limpieza y claridad en la implementación, también ayudará a entender con facilidad el código a otros desarrolladores si en un futuro se desea realizar otras versiones del sistema.

La realización de las pruebas garantizó que el sistema cumpla con todos los requerimientos funcionales previamente acordados, además garantizan un completo funcionamiento del mismo. Todas las no conformidades detectadas en la fase de pruebas quedaron resueltas.

Conclusiones.

La investigación realizada permitió arribar a las siguientes conclusiones:

- El impacto de los movimientos sociales en Internet, observado desde el sitio de redes sociales Twitter, puede ser estimado mediante la métrica de centralidad de grado para las relaciones de retuit y favoritos, para un conjunto determinado de etiquetas.
- El procesamiento asíncrono de los datos, mediante la llamada a procedimiento remoto, permite el análisis de grandes colecciones de tuits, haciendo uso de la información recolectada para la vecindad inmediata de los nodos estudiados.
- El sistema implementado cuenta con flexibilidad, extensibilidad, claridad y limpieza, al basarse en componentes con una alta cohesión, bajo acoplamiento y emplear el estilo de código definido para el lenguaje de programación Java.
- Se realizaron las pruebas al sistema desarrollado para el análisis de redes sociales, obteniéndose un producto calificado para su explotación, que contribuye al incremento de la capacidad de análisis de los movimientos sociales en Internet por parte del departamento DOWAI.
- La entrevista a los expertos seleccionados arrojó un alto nivel de concordancia sobre la utilidad de la aplicación en el análisis de movimientos sociales en la red social Twitter.

Recomendaciones

Para el desarrollo de futuras investigaciones sobre esta temática se recomienda al centro CIDI:

- Extender el estudio del impacto de los movimientos sociales en Internet, mediante el análisis de redes sociales centrado en el actor. Tomando como base el perfil del usuario y la clasificación de sus publicaciones.
- Se recomienda integrar las dimensiones de “recursos de comunicación y coordinación” y la dimensión de “recursos para captar adeptos y solidaridad” en la aplicación para un análisis más profundo del impacto que puede tener un tuit.

Bibliografía

1. **Achour, M., y otros. 2006.** *PHP Manual.* 2006.
2. **ActiveMQ. 2019.** ActiveMQ. [En línea] 2019. [Citado el: 10 de mayo de 2019.] https://activemq.apache.org/.
3. **Alvarez, Miguel, A. 2003.** Desarrolloweb6. [En línea] 19 de noviembre de 2003. [Citado el: 9 de marzo de 2019.] https://desarrolloweb.com/articulos/1325.php.
4. **APACHE. 2017.** What is the Apache HTTP Server Project? [En línea] 2017. [Citado el: 26 de mayo de 2019.] https://httpd.apache.org/ABOUT_APACHE.html..
5. **Applications., Social Network Analysis Theory and. 2011.** 2011.
6. **Baeldung. 2018.** Baeldung. [En línea] 2 de noviembre de 2018. https://www.baeldung.com/twitter4j.
7. **Barnes, Arundel,John. 1954.** Class and committees in a norwegian island parish. s.l. : Human relations, 1954. Vol. 7, págs. 39-58.
8. **Barrera, Martínez,Waldo. 2017.** Tesis para obtener el título de Máster en Historia Contemporánea y Relaciones Internacionales. *Redes sociales de Internet: una visión histórica de su impacto en los movimientos sociales contemporáneos (1995-2016). .* La Habana : s.n., 2017.
9. **Berge, Claude. 1985.** *Graphs.* 1985.
10. **Boyd, Danah, M. y Ellison, Nicole, B. 2007.** Social network sites: Definition, history, and scholarship. s.l. : Journal of Computer-Mediated Communication., 2007. Vol. 13, págs. 210-230.
11. **Brunn, Jonathan F., Hoy, Jeffrey R. y Silva, Asima. 2018.** *Evaluating an impact of a user's content utilized in a social network. . 9,881,345,* Estados Unidos, 30 de Enero de 2018.
12. **Cambronero, A. 2016.** El pequeño gran manual de Twitter. s.l. : Creative Commons., 2016.
13. **Carcar, Benito,J.E. 2015.** Las redes y los movimientos sociales ¿una acción colectiva o marketing viral?. 2015, Vol. 13, págs. 125-250.
14. **Chang. 2011.** “A New Perspective on Twitter Hashtag Use: Diffusion of Innovation Theory Paper presented at the 2010 Annual Meeting, Association for Information Science and Technology”. [En línea] 2011. [Citado el: 29 de Marzo de 2016.] https://www.asis.org/asist2010/proceedings/proceedings/ASIST_AM10/submissions/295_Final_Submission.pdf.
15. **Cronin, Bruce y al, et. 2016.** Social network analysis. . 2016.

16. **Daida, Midori. 2005.** Javalobby. [En línea] 4 de agosto de 2005. [Citado el: 21 de enero de 2019.] https://www.javalobby.org/java/forums/t20978.html.
17. **Escribano, Francisco. 2016.** beeva. [En línea] 30 de marzo de 2016. https://www.beeva.com/beeva-view/tecnologia/descubriendo-rabbitmq-una-solucion-para-colas-de-mensajeria/.
18. **Espinosa, Ramirez,Gabriel J. 2018.** Tésis para obtener el título de Máster en Informática Avanzada. *Sistema para la detección de roles sobre colecciones de tuits.* La Habana, Cuba : s.n., 2018.
19. **Figueiras, Carlos Julio. 2014.** Subsistema de detección de roles de usuario dentro de comunidades dentro de Twitter. La Habana : s.n., Diciembre de 2014.
20. **Fischer, E y Reuber, R. 2011.** "Social interaction via new social media: (How) can interactions on Twitter affect effectual thinking and behavior?". 2011. 26, págs. 1-18.
21. **Flores, González,Isabel, de la Cruz, Cruzata,.M,Cesar y Miguel, Medina,Vladimir. 2017.** Una contribución a la gestión de la información de ciencia, tecnología e innovación. s.l. : VivatAcademia. RevistadeComunicación, 15 de Septiembre de 2017. 140.
22. **Flores, Isabel González y Medina, Vladimir Miguel. 2017.** Una contribución a la gestión de la información de ciencia, tecnología e innovación. *Vivat Academia.* s.l. : Vivat Academia, 2017. 140, págs. 55-63.
23. **Gamma, Erich y otros. 2002.** *Patrones de Diseño.Elementos de software orientado a objetos reutilizables.* España : ADDISON-WESLEY, 2002.
24. **García, Luciano ,Richard y Lima, Isequilla,Carlos R. 2014.** Sistema basado en servicios web para el análisis de redes sociales. Habana : s.n., Junio de 2014.
25. **González, Fuertes G. 1988.** Psicología Comunitaria. España : s.n., 1988.
26. **Grundberg, Dahlberg, M. y Lindgren, S. 2014.** "Translocal Frame Extensions in a Networked Protest: situating the #IdleNoMore hashtag". 2014. Vol. 11, págs. 49-77.
27. **Hansen, Derek, Shneiderman, Ben y Smith, Mark A. 2011.** *Analyzing Social Media Networks with NodeXL.* 2011.
28. **Hashtagify.me. 2011.** Hashtagify.me. [En línea] CyBranding Ltd. , 2011. [Citado el: 23 de mayo de 2019.] http://hashtagify.me/.
29. **Hashtags.org. 2006.** Hashtags.org. [En línea] LOGIKA Corporation., 2006. [Citado el: 6 de abril de 2019.] : https://www.hashtags.org/.

30. **Hashtracking. 2010.** Hashtracking.com. [En línea] 2010. [Citado el: 3 de mayo de 2019.] https://www.hashtracking.com/.
31. **Hautz, Wolf,.E, y otros. 2016.** Six degrees of separation:the small world of medical education. 12 s.l. : Medical education, 2016. Vol. 50, págs. 1274–1279.
32. **Hommel, Scott y Otros. 1999.** Convenciones de Código para el lenguaje de programación Java. 20 de Abril de 1999.
33. **Hootsuite, We are social &. 2019.** Essential insights into how people around the world use the internet, mobile devisis, social media, and e-commerce. [En línea] Digital 2019, 2019. https://wearesocial.com/ .
34. **Huang, Shaobin, y otros. 2014.** Identifying node role in social network based on multiple indicators. 8 s.l. : PLOS ONE, agosto de 2014. Vol. 9, págs. 1-16.
35. **Ibarra, Carlos. 2000.** 2, Los estudios sobre los movimientos sociales: estado de la cuestión. : s.n., 2000, Revista Española de Ciencia Política, Vol. 1, págs. 271-290.
36. **ICTA. 2019.** [En línea] 2019. [Citado el: 15 de febrero de 2019.] http://www.ictea.com/cs/index.php?rp=/knowledgebase/8858/iQue-es-el-lenguaje-de-programacion-Cplusplus.html.
37. **Jacobson, Ivar. 2004.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004.
38. **JetBrains. 2019.** JetBrains. [En línea] 2019. [Citado el: 23 de marzo de 2019.] https://www.jetbrains.com/phpstorm/features/.
39. **Kafka, Apache. 2019.** Apache Kafka. [En línea] 2019. [Citado el: 10 de mayo de 2019.] https://kafka.apache.org/intro.
40. **Kleinberg, .M, Jon. 1999.** Authoritative sources in a hyperlinked environment. 1999. Vol. 5, 46, págs. 604-632.
41. **Kruchten, Philippe y Kroll, Per. 2003.** *The Rational Unified Process Made Easy: A Practitioners Guide to the RUP. Amsterdam : Addison-Wesley.* 2003.
42. **Kuz, Antonieta, Falco, Mariana y Giandini, Roxana. 2015.** El analisis de redes sociales. *Análisis de redes sociales :un caso práctico.* Buenos Aires, Argentina : s.n., Diciembre de 2015.

43. **Langrado, Daniela, Rita, Paulo y DE FÁTIMA SALGUEIRO, María. 2018.** Do social networking sites contribute for building brands? Evaluating the impact of users' participation on brand awareness and brand attitude. 2018. Vol. 24, 2, págs. 146-168.
44. **Larman, Craig. 2004.** *UML y Patrones. Una introducción al análisis y diseño orientado a objetos.* La Habana : Félix Varela, 2004. Vols. 842-053-438-2.
45. **López, María y Sierra, Patricia. 2017.** *Herramienta CASE Visual Paradigm.* Univ. Cantabria – Fac. de Ciencias : s.n., 2017.
46. **McCarty, Christopher y Molina, Jose ,L. 2015.** *Social network analysis.Handbook of methods in cultural anthropology. .* Lanham : Rowman and Littlefield, 2015. págs. 631–657. Vol. 2.
47. **MDN. 2019.** MDN web docs. [En línea] 18 de marzo de 2019. [Citado el: 14 de mayo de 2019.] https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n.
48. **Mederos, Díaz,.D,Dayni. 2017.** Formación formación por competencias del docente de categoría superior de la Universidad de Cienfuegos para la gestión de proyectos de internacionalización. Cienfuegos, Cuba : s.n., 2017.
49. **Metricool. 2019.** Metricool. [En línea] 2019. https://app.metricool.com/.
50. **Moffitt, Jim. 2016.** Updating Twitter Hosebird Client for PowerTrack 2.0. [En línea] 2016. [Citado el: 2 de junio de 2019.] http://support.gnip.com/articles/updating-hbc-for-ptv2.html.
51. **Muñoz, Caro, C., Niño, A. y Vizcaíno, Barceló, A. 2002.** Introducción a la programación con orientación a objetos. s.l. : Prentice-Hall, 2002.
52. **Naranjo, Roberto,Esteban. 2018.** Módulo para la gestion de servidores proxy squid del sistema xilema smart keeper. La Hbana : s.n., 2018.
53. **NetBeans. 2019.** NetBeans IDE. [En línea] 2019. [Citado el: 1 de Mayo de 2019.] https://netbeans.org/features/index.html..
54. **Nginx. 2019.** Nginx. [En línea] 2019. [Citado el: 2 de junio de 2019.] https://www.nginx.com/.
55. *Node centrality in weighted networks: Generalizing degree and shortest paths.* **Tore Opsahl, Filip Agneessens, y John Skvoretz. 2010.** 2010, Social networks, págs. 32(3):245– –251.
56. **Opsahl, Tore, Agneessens, Filip y Skvoretz, John. 2010.** *Node centrality in weighted networks: Generalizing degree and shortest paths.* s.l. : Social networks, 2010. págs. 245-251. Vol. 32.
57. **ORACLE. 2012.** ¿Qué es la tecnología Java y por qué lo necesito? 2012.

58. **Oracle. 2019.** Oracle. [En línea] 2019. [Citado el: 8 de mayo de 2019.] https://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html.
59. **Ortiz, Galindo, R. 2016.** Los cibermovimientos sociales: una revisión del concepto y marco teórico. s.l. : Communication & Society, 2016. Vol. 29, 4, págs. 165-183.
60. **Pacheco, Nacho. 2013.** Bases de datos y Doctrine. [En línea] 2013. [Citado el: 2019 de enero de 15.] http://gitnacho.github.io/symfony-docs-es/book/doctrine.html.
61. **Passmore, .L,David. 2011.** Social network analysis: Theory and applications. s.l. : Institute for Research in Training & Development–IRTD, 2011.
62. **Perera, Ramos, R. 2015.** Movimientos 2.0. Memorias VIII Encuentro de Investigadores y Estudiosos de la Información y la Comunicación y IX Congreso de la Unión Latina de la Economía Política de la Información, la Comunicación y la Cultura (ICOM-ULEPICC 2015). La Habana : s.n., 2015.
63. **Pérez, Pérez,Arturo J.** La investigación REDES SOCIALES VIRTUALES Y LA BIOÉTICA. Medellín : s.n.
64. **PHP-Fig. 2018.** PHP Standards Recomentation. [En línea] 2018. [Citado el: 20 de mayo de 2019.] https://www.php-fig.org/psr/.
65. **PostgreSQL. 2010.** *PostgreSQL.* 2010.
66. **Pressman, Roger. 2009..** *Ingenieria de Software: Un enfoque práctico. .* Nueva York : McGraw-Hill, 2009.
67. **Protzel, J. 2016.** Comunicación por correo electrónico. Lima, Perú : s.n., 2 de 8 de 2016.
68. **Proyecto, Lider de. 2014.** Lider de Proyecto. [En línea] 1 de mayo de 2014. [Citado el: 10 de enero de 2019.] http://www.liderdeproyecto.com/glosario/#m.
69. **QUALITY. 2016.** Pruebas de Seguridad. [En línea] 2016. [Citado el: 5 de mayo de 2019.] http://vyvquality.com/pruebas-seguridad/.
70. **Ramos, Alonso ,Juan. 2014.** adictos al trabajo. [En línea] 7 de julio de 2014. https://www.adictosaltrabajo.com/tutoriales/neo4j-first-steps.
71. **Redis. 2019.** Redis. [En línea] 2019. [Citado el: 10 de mayo de 2019.] https://redis.io/topics/introduction.
72. **Rees, Dayle. 2012.** *Code Happy.* 2012.
73. **Reichmann, J. y Fernández, Buey, F. 1994.** Redes que dan libertad. Barcelona : Paidós, 1994.

74. **Rincón, Carrera, J. M. 2011.** Estudio de la Evolución Web y Lenguajes Dinámicos. Madrid, España : Universidad Carlos III de Madrid, 2011.
75. **Roldán, Santana,Carlos.** ¿Qué es el Análisis de Redes Sociales (Social Network Analysis)? Code Jobs Aprende a programar. .
76. **Ros, Martín, M. 2009.** Evolución de los servicios de Redes Sociales en Internet. s.l. : El Profesional de la Información, 2009. Vol. 18, 5, págs. 552-558.
77. **Rosa, Amaro La. 2016.** Movimientos sociales, redes sociales y recursos simbólicos. 2016.
78. **Sanchez, Rodriguez,.T. 2014.** Metodologia de desarrollo para la actividad productiva en la UCI. 2014.
79. **Sariol, Martínez ,Elsa, y otros. 2018.** Identificación de las competencias específicas de los profesionales de enfermería en la atención al neonato en estado grave. *III Dra. C. Omayda Urbina Laza IV y Lic. Irasbel Martínez Ramírez.* Santiago de Cuba : s.n., 2018.
80. **Serrat, Olivier. 2017.** *Social network analysis. In Knowledge solutions.* s.l. : Springer, 2017. págs. 39–43.
81. **2019.** Significados. [En línea] 13 de febrero de 2019. https://www.significados.com/tweet/.
82. **Slutsker, Stas. 2017.** Quora. [En línea] 30 de Julio de 2017. [Citado el: 2 de Mayo de 2019.] https://www.quora.com/How-does-RPC-work.
83. **Sommerville. 2011.** *Software Engeniering 9.* 2011.
84. **Sommerville, Ian. 2005.** *Ingeniería de Software.* La Habana : Félix Varela, 2005.
85. **Pressman. 2002.** *Ingeniería de Software. 6ta. Edición.* s.l. : Prentice-Hall, 2002.
86. **Steven, Liu y Kevin, Oliver. 2013.** GitHub. [En línea] 2013. [Citado el: 2 de junio de 2019.] https://github.com/twitter/hbc.
87. **Symfony. 2019.** Symfony. [En línea] 2019. [Citado el: 18 de febrero de 2019.] http://symfony.es/que.
88. **Tanenbaum, Aarom, .M y Langsam, Yedidyah. 1993.** *Estructura de datos en C 1era edición.* s.l. : Prentice Hall Hispanoamerica, S.A., 1993. Vol. 1.
89. **Twitter. 2018.** Twitter. [En línea] 10 de noviembre de 2018. https://help.twitter.com/es/rules-and-policies/twitter-api.
90. **Wasserman, Stanley y Faust, Katherine. 1994.** *Social network analysis: Methods and applications.* s.l. : Cambridge university press, 1994. Vol. volumen 8.